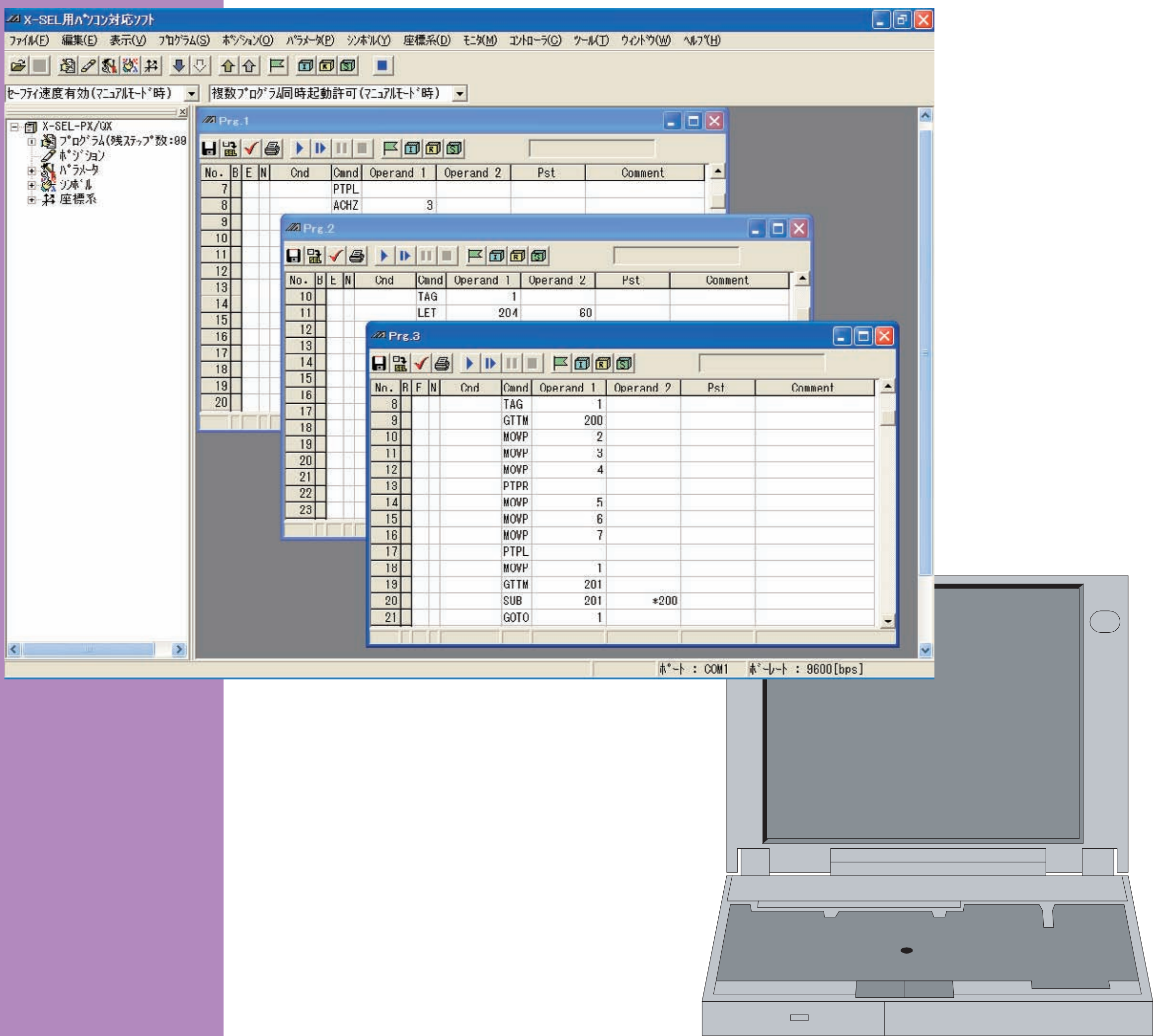
INTELLIGENT ACTUATOR

# SEL Language Programming Manual

Tenth Edition




IAI Corporation

INTELLIGENT ACTUATOR

# Please Read Before Use

Thank you for purchasing our product.

This Instruction Manual describes all necessary information items to operate this product safely such as the operation procedure, structure and maintenance procedure.

Before the operation, read this manual carefully and fully understand it to operate this product safely. The enclosed CD/DVD in this product package includes the Instruction Manual for this product.
For the operation of this product, print out the necessary sections in the Instruction Manual or display them using the personal computer.

After reading through this manual, keep this Instruction Manual at hand so that the operator of this product can read it whenever necessary.

## [Important]

- This Instruction Manual is original.
- The product cannot be operated in any way unless expressly specified in this Instruction Manual. IAI shall assume no responsibility for the outcome of any operation not specified herein.
- Information contained in this Instruction Manual is subject to change without notice for the purpose of product improvement.
- If you have any question or comment regarding the content of this manual, please contact the IAI sales office near you.
- 
- 

# Contents

# Table of Contents of Commands in Alphabetical Order

Some commands cannot be used depending on the actuator. For details, refer to individual commands.

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CLLV | Collision Detection Level Setting | | | | | | | | ○ (V1.10 or later) | | | | | 349 |
| CLOS | Close channel | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 466 |
| CLR | Clear variable | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 266 |
| CNTP | PTP Continuous Operation Mode Setting | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 410 |
| COL | Collision Detection Feature Valid / Invalid Setting | | | | | | | | ○ (V1.10 or later) | | | | | 350 |
| COMP | Compliance Mode Setting | | | | | | | | ○ (V1.10 or later) | | | | | 636 |
| COS | Cosine | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 274 |
| CP□□ | Compare number of variable comparisons based on free comparison | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 287 |
| **D** | | | | | | | | | | | | | | |
| DCL | Set deceleration | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 337 |
| DCLS | Set deceleration ratio for PTP operation | | | ○ | ○ | | | ○ | | | | ○ (PCX/PGX only) | | 338 |
| DEG | Set division angle | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 344 |
| DFIF | Define coordinates of simple interference check zone | | | ○ | ○ | | | ○ | | | | ○ (PCX/PGX only) | | 386 |
| DFTL | Define tool coordinate system | | | ○ | ○ | | | ○ | | | | ○ | ○ (TTA only) | 365 |
| DFWK | Define load coordinate system | | | ○ | ○ | | | ○ | | | | ○ | ○ (TTA only) | 372 |
| DIS | Set spline division distance | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 356 |
| DIV | Divide | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 270 |
| DTOR | Angle conversion (degrees to radians) | | | | | | ○ | | ○ | | | ○ | | 279 |
| DW□□ | Loop [EQ, NE, GT, GE, LT, LE] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 448 |
| **E** | | | | | | | | | | | | | | |
| ECMD1 | Get motor current value | | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ (V1.18 or later) | ○ (TTA only V1.22 or later) | 519 |
| ECMD2 | Get home sensor status | | ○ | | | ○ | ○ | ○ | ○ | | | | | 520 |
| ECMD3 | Get overrun sensor status | | ○ | | | ○ | ○ | ○ | ○ | | | | | 521 |
| ECMD4 | Get creep sensor status | | ○ | | | ○ | ○ | ○ | ○ | | | | | 522 |
| ECMD5 | Get axis operation status | | ○ | | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | 523 |
| ECMD6 | Current position acquirement on each axis system | | | | ○ | | | ○ (V1.20 or later) | ○ | | | ○ | ○ (TTA only) | 524 |
| ECMD7 | Get total movement count | | | | | | | | | | | ○ | ○ (TTA only) | 525 |
| ECMD8 | Get total mileage | | | | | | | | | | | ○ | ○ (TTA only) | 526 |
| ECMD9 | Get position deviation | | | | | | | | | | | ○ | ○ (TTA only) | 527 |
| ECMD10 | Acquirement of Overload Level | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 528 |
| ECMD11 | Acquirement of Encoder Overheated Level | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 529 |
| ECMD20 | Get parameter | | ○ | | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | 530 |
| ECMD250 | Set torque limit/detection time for torque limit over error | | ○ | | ○ | ○ | ○ | ○ | ○ | | ○ (V0.52 or later) | Δ (With a condition) | | 532 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ECMD280 | Conversion from each axis coordinates to work coordinates in wrist unit equipped robot | | | | | | ○ | | | | | | | 535 |
| ECMD281 | Conversion from work coordinates to each axis coordinates in wrist unit equipped robot | | | | | | ○ | | | | | | | 536 |
| ECMD282 | Conversion from tool coordinates to work coordinates in wrist unit equipped robot | | | | | | ○ | | | | | | | 538 |
| ECMD290 | Conversion from each axis coordinates to work coordinates in wrist unit equipped robot | | | | | | ○ | | | | | | | 535 |
| ECMD291 | Conversion from work coordinates to each axis coordinates in wrist unit equipped robot | | | | | | ○ | | | | | | | 536 |
| ECMD292 | Conversion from tool coordinates to work coordinates in wrist unit equipped robot | | | | | | ○ | | | | | | | 538 |
| ECMD300 | User system error output | | | | | | ○ | | ○ | | | | | 540 |
| EDDO | Declare end of DO | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 451 |
| EDIF | Declare end | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 447 |
| EDPA | Declare end of palletizing setting | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 492 |
| EDSL | Declare end of SLCT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 456 |
| EDSR | End subroutine | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 308 |
| ELSE | Declare execution destination when IF command condition is not satisfied | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 446 |
| EOR | Logical exclusive OR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 283 |
| EXIT | End program | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 309 |
| EXPG | Start program | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 310 |
| EXSR | Execute subroutine | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 306 |
| **F** | | | | | | | | | | | | | | |
| FMIO | Set IN (B)/OUT (B) command format | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 299 |
| **G** | | | | | | | | | | | | | | |
| GACC | Get acceleration data | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 329 |
| GARM | Get current arm system | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PG X only) | | 460 |
| GCLX | Acquiring Max. Collision Level | | | | | | | | ○ (V1.10 or later) | | | | | 351 |
| GDCL | Get deceleration data | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 330 |
| GOTO | Jump | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 304 |
| GRP | Set group axes | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 346 |
| GTIF | Get definition coordinates of simple interference check zone | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PG X only) | | 390 |
| GTTL | Get tool coordinate system definition data | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 370 |
| GTAM | Acquirement of target arm system data | | | | | | | ○ | ○ | | | ○ (PCX/PG X only) | | 327 |
| GTTM | Get time | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 290 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GTVD | Image capture command | | ○ | | | ○ | ○ | | | | | ○ (PC/PG only) | | 633 |
| GTWK | Get load coordinate system definition data | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 377 |
| GVEL | Get speed data | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 328 |
| **H** | | | | | | | | | | | | | | |
| HOLD | Declare port to pause | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 347 |
| HOME | Return to home | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 398 |
| **I** | | | | | | | | | | | | | | |
| IF☐☐ | Compare [EQ, NE, GT, GE, LT, LE] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 444 |
| INB | Input BCD (8 digits max.) | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 296 |
| IN | Input binary (32 bits max.) | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 295 |
| IPCN | Connected Destination IP address / Port Number Setting | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 475 |
| IS☐☐ | Compare strings | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 445 |
| ITER | Repeat DO | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 450 |
| **J** | | | | | | | | | | | | | | |
| J☐W☐ | Jog [FN, FF, BN, BF] | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 414 |
| **L** | | | | | | | | | | | | | | |
| LEAV | Pull out from DO | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 449 |
| LEFT | Change current arm system to left arm | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 381 |
| LET | Assign | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 264 |
| LSFT | Logic Shifted to Left | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 285 |
| **M** | | | | | | | | | | | | | | |
| MOD | Calculate remainder | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 271 |
| MOVD | Move by direct value specification | | | | | | | | | ○ | ○ | ○ (PCX/PGX only) | | 407 |
| MOVL | Move to specified position via interpolation | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 401 |
| MOVP | Move to specified position | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 399 |
| MULT | Multiply | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 269 |
| MVDI | Move incrementally by direct value specification | | | | | | | | | ○ | ○ | ○ (PC/PG only) | | 408 |
| MVLI | Move to relative position via interpolation | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 405 |
| MVPI | Move to relative position | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 403 |
| **N** | | | | | | | | | | | | | | |
| NBND | Set close distance | ○ | | | | | | | | | | | | 395 |
| NOT | Deny | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 284 |
| NTCH | Anti-Vibration Control Parameter Set Select | | ○ | | | ○ | ○ | ○ (Linear drive axis) | ○ (Linear drive axis) | | | | | 635 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **O** | | | | | | | | | | | | | | |
| OFAZ | Set arch-motion Z-axis offset | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 490 |
| OFPZ | Set palletizing Z-axis offset | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 505 |
| OFST | Set offset | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 343 |
| OPEN | Open channel | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 465 |
| OR | Logical OR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 282 |
| OTHE | Declare branching destination when condition is not satisfied | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 455 |
| OTPS | Output current position | | | | | | | | | | ○ | | | 303 |
| OUT | Output binary (32 bits max.) | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 297 |
| OUTB | Output BCD (8 digits max.) | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 298 |
| OUTR | Output relay for ladder | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 175 |
| OVRD | Set speed coefficient | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 333 |
| **P** | | | | | | | | | | | | | | |
| PACC | Assign position acceleration | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 323 |
| PACH | Palletizing point arch motion | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 514 |
| PAPG | Get palletizing calculation data | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 511 |
| PAPI | Set palletizing counts | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 493 |
| PAPN | Set palletizing pattern | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 494 |
| PAPR | Set PUSH command distance, speed | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 358 |
| PAPS | Set palletizing points for 3-point or 4-point teaching | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 498 |
| PAPT | Set palletizing pitches | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 496 |
| PARG | Get palletizing angle | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 510 |
| PASE | Set palletizing axes | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 495 |
| PAST | Set palletizing reference point | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 497 |
| PATH | Move along path | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 409 |
| PAXS | Read axis pattern | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 325 |
| PBND | Set positioning band | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 436 |
| PCHZ | Set palletizing Z-axis | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 502 |
| PCLR | Clear position data | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 316 |
| PCPY | Copy position data | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 317 |
| PDCL | Assign position deceleration | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 324 |
| PDEC | Decrement palletizing position number by 1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 508 |
| PEND | Wait for end of operation of axis using current program | ○ | | | | | | | | | | | | 443 |
| PEXT | Set palletizing composition | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 504 |
| PGET | Assign position to variable 199 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 314 |
| PGST | Get program status | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 458 |
| PINC | Increment palletizing position number by 1 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 507 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PMVL | Move to palletizing points via interpolation | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ (PC/PG only) | ○ | 513 |
| PMVP | Move to palletizing points via PTP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 512 |
| POTP | Set PATH output type | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 357 |
| PPUT | Assign value of variable 199 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 315 |
| PRDQ | Read current axis position (1 axis direct) | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 319 |
| PRED | Read current axis position | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 318 |
| PSET | Set palletizing position number directly | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 509 |
| PSIZ | Confirm position size | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 326 |
| PSLI | Set zigzag | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 501 |
| PSPL | Move along spline | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 418 |
| PTAM | Substitution of target arm system data | | | | | | | ○ | ○ | | | ○ (PCX/PGX only) | | 320 |
| PTNG | Get palletizing position number | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 506 |
| PTPD | Specify current arm as PTP target arm system | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 384 |
| PTPE | Specify current arm as PTP target arm system | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 385 |
| PTPL | Specify left arm as PTP target arm system | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 383 |
| PTPR | Specify right arm as PTP target arm system | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 382 |
| PTRG | Set palletizing arch triggers | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 503 |
| PTRQ | Change push torque limit parameter | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 421 |
| PTST | Confirm position data | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 321 |
| PUSH | Move by push motion | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 419 |
| PVEL | Assign position speed | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 322 |
| **Q** | | | | | | | | | | | | | | |
| QRTN | Set quick return mode | ○ | ○ | | | ○ | ○ | | ○ (V1.10 or later) | ○ | ○ | ○ (PC/PG only) | ○ | 359 |
| **R** | | | | | | | | | | | | | | |
| RAXS | Set RC axis pattern | | ○ | | ○ | ○ | | ○ | | | | | | 555 |
| RCST | Get RC axis status | | ○ | | ○ | ○ | | ○ | | | | | | 565 |
| READ | Read from channel | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 467 |
| RGAD | Assign RC axis position acceleration/ deceleration to variable 199 | | ○ | | ○ | ○ | | ○ | | | | | | 552 |
| RGIP | Assign RC axis position positioning width to variable 199 | | ○ | | ○ | ○ | | ○ | | | | | | 553 |
| RGTQ | Assign RC axis position current-limiting value to variable 199 | | ○ | | ○ | ○ | | ○ | | | | | | 554 |
| RGVL | Assign RC axis position speed to variable 199 | | ○ | | ○ | ○ | | ○ | | | | | | 551 |
| RHOM | Return RC axis to its home | | ○ | | ○ | ○ | | ○ | | | | | | 558 |
| RIGH | Change right arm of current arm system | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 380 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RMDI | Incremental move by RC axis direct specification | | ○ | | ○ | ○ | | ○ | | | | | | 562 |
| RMPI | Incremental move by RC axis position specification | | ○ | | ○ | ○ | | ○ | | | | | | 560 |
| RMVD | Move by RC axis direct specification | | ○ | | ○ | ○ | | ○ | | | | | | 561 |
| RMVP | Move by RC axis position specification | | ○ | | ○ | ○ | | ○ | | | | | | 559 |
| RPAD | Assign variable 199 to RC axis position acceleration/ deceleration | | ○ | | ○ | ○ | | ○ | | | | | | 548 |
| RPCP | Copy RC axis position data | | ○ | | ○ | ○ | | ○ | | | | | | 544 |
| RPCR | Clear RC axis position data | | ○ | | ○ | ○ | | ○ | | | | | | 543 |
| RPGT | Assign RC axis position to variable 199 | | ○ | | ○ | ○ | | ○ | | | | | | 541 |
| RPIP | Assign variable 199 to RC axis position positioning band | | ○ | | ○ | ○ | | ○ | | | | | | 549 |
| RPPT | Assign variable 199 to RC axis position | | ○ | | ○ | ○ | | ○ | | | | | | 542 |
| RPRD | Read current RC axis position | | ○ | | ○ | ○ | | ○ | | | | | | 545 |
| RPRQ | Read current RC axis position (1 axis, direct) | | ○ | | ○ | ○ | | ○ | | | | | | 546 |
| RPTQ | Assign variable 199 to RC axis position current-limiting value | | ○ | | ○ | ○ | | ○ | | | | | | 550 |
| RPUS | Move by RC axis push-motion operation | | ○ | | ○ | ○ | | ○ | | | | | | 563 |
| RPVL | Assign variable 199 to RC axis position speed | | ○ | | ○ | ○ | | ○ | | | | | | 547 |
| RSFT | Logic Shifted to Right | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 286 |
| RSOF | Turn RC axis servo OFF | | ○ | | ○ | ○ | | ○ | | | | | | 557 |
| RSON | Turn RC axis servo ON | | ○ | | ○ | ○ | | ○ | | | | | | 556 |
| RSPG | Resume program | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 313 |
| RSTP | Decelerate RC axis to stop | | ○ | | ○ | ○ | | ○ | | | | | | 564 |
| RTOD | Angle conversion (radians to degrees) | | | | | | ○ | | ○ | | | ○ | | 279 |
| **S** | | | | | | | | | | | | | | |
| SCHA | Set end character | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 474 |
| SCLO 0 | Compliance Mode Option Feature Setting (Searching Operation Setting) | | | | | | | | ○ (V1.10 or later) | | | | | 638 |
| SCLO 1 | Compliance Mode Option Feature Setting (J1 & J2-Axes Torque Limit Mode Setting) | | | | | | | | ○ (V1.10 or later) | | | | | 640 |
| SCLG | Compliance Gain Setting | | | | | | | | ○ (V1.10 or later) | | | | | 641 |
| SCMP | Compare character strings | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 477 |
| SCPY | Copy character string | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 476 |
| SCRV | Set sigmoid motion ratio | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 339 |
| SEIF | Specify error type for simple contact check area | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 389 |
| SGET | Get character | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 478 |
| SGN | Get signs | | | | | | ○ | | ○ | | | ○ | | 280 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SIN | Sine | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 272 |
| SLCT | Declare start of multi-branching | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 452 |
| SLEN | Set length | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 484 |
| SLTL | Select tool coordinate system | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 368 |
| SLVS | Declare use of Vision System | | ○ | | | ○ | ○ | | | | | ○ (PC/PG only) | | 631 |
| SLWK | Select load coordinate system | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 375 |
| SOIF | Specify output for simple interference check zone | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 388 |
| SPUT | Set character | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 479 |
| SQR | Root | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 278 |
| SSPG | Pause program | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 312 |
| STOP | Decelerate and stop axis | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 417 |
| STR | Convert character string; decimal | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 480 |
| STRH | Convert character string; hexadecimal | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 481 |
| SUB | Subtract | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 268 |
| SV□□ | Servo [ON, OF] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 397 |
| SYST | Get system status | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 459 |
| **T** | | | | | | | | | | | | | | |
| TAG | Jump destination | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 305 |
| TAN | Tangent | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 276 |
| TIMC | Cancel waiting | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 289 |
| TIMR | Timer relay for ladder | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 175 |
| TIMW | Wait | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 288 |
| TMLI | Move relatively between positions on tool coordinate system via interpolation | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 438 |
| TMPI | Move relatively between positions on tool coordinate system | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 437 |
| TMRD | Set read timeout value | ○ | | ○ | | | | | | | | | ○ (TT only) | 469 |
| TMRW | Set read timeout value | | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ (TTA only) | 471 |
| TPCD | Specify processing to be performed when input condition is not specified | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 517 |
| TRAC | Image capturing and tracking command | | ○ | | ○ | ○ | ○ | ○ (Linear drive axis) | ○ (Linear drive axis) | | | | | 628 |
| TRAN | Copy | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 265 |
| TSLP | Task sleep | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 518 |
| TRMD | Declare use of Conveyor Tracking | | ○ | | ○ | ○ | ○ | ○ | ○ | | | | | 627 |
| **V** | | | | | | | | | | | | | | |
| VAL | Convert character string data; decimal | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 482 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VALH | Convert character string data; hexadecimal | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 483 |
| VEL | Set speed | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 331 |
| VELS | Set speed ratio for PTP operation | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 332 |
| VLMX | Specify VLMX speed | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 355 |
| **W** | | | | | | | | | | | | | | |
| WGHT | Set tip load mass/inertial moment | | | | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 391 |
| WGT2 | Tip load condition setting 2 | | | | | | | ○ | ○ | | | | | 393 |
| WH□□ | Branch value [EQ, NE, GT, GE, LT, LE] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 453 |
| WRIT | Output to channel | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 473 |
| WS□□ | Branch character string [EQ, NE] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 454 |
| WT□□ | Wait for I/O, flag [ON, OF] | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 294 |
| WZFA | Wait for zone OFF, with AND | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 463 |
| WZFO | Wait for zone OFF, with OR | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 464 |
| WZNA | Wait for zone ON, with AND | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 461 |
| WZNO | Wait for zone ON, with OR | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 462 |
| **X** | | | | | | | | | | | | | | |
| XA16 | Extension motion control axis patterns setting (16 to 31 axes) | | | | | | ○ | | ○ | | | | | 584 |
| XACH | Extension motion control axis arch motion | | | | | | ○ | | ○ | | | | | 622 |
| XACZ | Extension motion control axis arch motion Z-axis declaration | | | | | | ○ | | ○ | | | | | 624 |
| XAEX | Extension motion control axis arch motion composition setting | | | | | | ○ | | ○ | | | | | 624 |
| XAST | Acquire extension motion control axis status | | ○ (0 to 15 axes) | | | ○ | ○ | | ○ | | | | | 619 |
| XATG | Extension motion control axis arch trigger setting | | | | | | ○ | | ○ | | | | | 625 |
| XAXS | Set extension motion control axis patterns | | ○ | | | ○ | ○ | | ○ | | | | | 583 |
| XCAS | Start synchronizing extension motion control axis electronic cam (indicating main axis) | | ○ | | | ○ | ○ | | | | | | | 601 |
| XCRP | Clear input counter record for extension motion control axis | | ○ | | | ○ | ○ | | | | | | | 567 |
| XCTM | Move extension motion control axis individual electronic cam (indicating time) | | ○ | | | ○ | ○ | | | | | | | 612 |
| XGAC | Read extension motion control axis acceleration data | | ○ | | | ○ | ○ | | ○ | | | | | 580 |
| XGDC | Read extension motion control axis deceleration data | | ○ | | | ○ | ○ | | ○ | | | | | 581 |
| XGIP | Read extension motion control axis poaitioning width data | | ○ | | | ○ | ○ | | ○ | | | | | 582 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XGTP | Acquire current record of extension motion control axis input counter | | ○ | | | ○ | | | | | | | | 568 |
| XGVL | Read extension motion control axis speed data | | ○ | | | ○ | ○ | | ○ | | | | | 579 |
| XHOM | Return extension motion control axis to home position | | ○ | | | ○ | ○ | | ○ | | | | | 587 |
| XJ□□ | Perform extension motion control axis jog operation | | ○ | | | ○ | ○ | | ○ | | | | | 596 |
| XMDI | Move extension motion control axis to directly indicated relative position | | ○ | | | ○ | ○ | | ○ | | | | | 593 |
| XMLI | Move extension motion control axis for position relative interpolation | | ○ | | | ○ | ○ | | ○ | | | | | 591 |
| XMPI | Perform extension motion control axis position relative movement | | ○ | | | ○ | ○ | | ○ | | | | | 589 |
| XMVD | Move extension motion control axis to directly indicated absolute position | | ○ | | | ○ | ○ | | ○ | | | | | 592 |
| XMVL | Move extension motion control axis for position indicated interpolation | | ○ | | | ○ | ○ | | ○ | | | | | 590 |
| XMVP | Move extension motion control axis to indicated position | | ○ | | | ○ | ○ | | ○ | | | | | 588 |
| XOAZ | Extension motion control axis arch motion Z-axis offset setting | | | | | | ○ | | ○ | | | | | 626 |
| XPAC | Write extension motion control board axis acceleration data | | ○ | | | ○ | ○ | | ○ | | | | | 576 |
| XPCP | Copy extension motion control board axis position data | | ○ | | | ○ | ○ | | ○ | | | | | 572 |
| XPCR | Erase extension motion control axis position data | | ○ | | | ○ | ○ | | ○ | | | | | 571 |
| XPDC | Write extension motion control axis deceleration data | | ○ | | | ○ | ○ | | ○ | | | | | 577 |
| XPED | Waiting for extension motion control axis to finish positioning operation of axis used by self-program | | ○ | | | ○ | ○ | | ○ | | | | | 597 |
| XPGT | Read extension motion control axis position data | | ○ | | | ○ | ○ | | ○ | | | | | 569 |
| XPIP | Write extension motion control axis positioning complete width data | | ○ | | | ○ | ○ | | ○ | | | | | 578 |
| XPPT | Write extension motion control axis position data | | ○ | | | ○ | ○ | | ○ | | | | | 570 |
| XPRD | Read extension motion control axis current command position | | ○ | | | ○ | ○ | | ○ | | | | | 573 |
| XPRQ | Read extension motion control axis current command position (single-axis direct) | | ○ | | | ○ | ○ | | ○ | | | | | 574 |
| XPTH | Extension motion control axis path operation | | | | | | ○ | | ○ | | | | | 594 |
| XPVL | Write extension motion control axis speed data | | ○ | | | ○ | ○ | | ○ | | | | | 575 |

| Command | Function | XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| XSFS | Start synchronizing of extension motion control axis electronic shaft | | ○ | | | ○ | ○ | | | | | | | 614 |
| XSOF | Extension motion control axis servo OFF | | ○ | | | ○ | ○ | | ○ | | | | | 586 |
| XSON | Extension motion control axis servo ON | | ○ | | | ○ | ○ | | ○ | | | | | 585 |
| XSTP | Cancel operation of extension motion control axis | | ○ | | | ○ | ○ | | ○ | | | | | 598 |
| XSYE | Synchronizing of extension motion control axis completed | | ○ | | | ○ | ○ | | | | | | | 617 |
| XWIP | Waiting for extension motion control axis positioning complete signal to be turned ON | | ○ | | | ○ | ○ | | ○ | | | | | 599 |

## Table of Contents of Commands by Function

Some commands cannot be used depending on the actuator. For details, refer to individual commands.

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I/O, flag operation | Output OFF pulse | BTPF | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 293 |
| | Output ON pulse | BTPN | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 292 |
| | Output, flag [ON, OF, NT] | BT□□ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 291 |
| | Set IN (B)/OUT (B) command format | FMIO | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 299 |
| | Input binary (32 bits max.) | IN | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 295 |
| | Input BCD (8 digits max.) | INB | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 296 |
| | Output current position | OTPS | | | | | | | | | | ○ | | | 303 |
| | Output binary (32 bits max.) | OUT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 297 |
| | Output BCD (8 digits max.) | OUTB | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 298 |
| | Wait for I/O, flag [ON, OF] | WT□□ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 294 |
| Program control | Start subroutine | BGSR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 307 |
| | End subroutine | EDSR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 308 |
| | Execute subroutine | EXSR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 306 |
| | Jump | GOTO | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 304 |
| | Declare jump destination | TAG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 305 |
| Task management | Stop other program | ABPG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 311 |
| | End program | EXIT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 309 |
| | Start program | EXPG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 310 |
| | Resume program | RSPG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 313 |
| | Pause program | SSPG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 312 |
| Position operation | Get acceleration data | GACC | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 329 |
| | Get deceleration data | GDCL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 330 |
| | Acquirement of target arm system data | GTAM | | | | | | | ○ | ○ | | | ○ (PCX/PGX only) | | 327 |
| | Get speed data | GVEL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 328 |
| | Assign position acceleration | PACC | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 323 |
| | Read axis pattern | PAXS | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 325 |
| | Clear position data | PCLR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 316 |
| | Copy position data | PCPY | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 317 |
| | Assign position deceleration | PDCL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 324 |
| | Assign position to variable 199 | PGET | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 314 |
| | Assign value of variable 199 | PPUT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 315 |
| | Read current axis position (1 axis direct) | PRDQ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 319 |
| | Read current axis position | PRED | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 318 |
| | Confirm position size | PSIZ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 326 |
| | Substitution of target arm system data | PTAM | | | | | | | ○ | ○ | | | ○ (PCX/PGX only) | | 320 |
| | Confirm position data | PTST | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 321 |
| | Assign position speed | PVEL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 322 |

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Actuator control declaration | Set acceleration | ACC | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 335 |
| | Set acceleration ratio for PTP operation | ACCS | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 336 |
| | Indicate ACMX acceleration | ACMX | | ○ | | | ○ | ○ | ○ | ○ | | ○ | | | 352 |
| | Set reference axis | BASE | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 345 |
| | Declare port to abort | CANC | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 348 |
| | Collision Detection Level Setting | CLLV | | | | | | | | ○ (V1.10 or later) | | | | | 349 |
| | Collision Detection Feature Valid / Invalid Setting | COL | | | | | | | | ○ (V1.10 or later) | | | | | 350 |
| | Set deceleration | DCL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 337 |
| | Set deceleration ratio for PTP operation | DCLS | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 338 |
| | Set division angle | DEG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 344 |
| | Define coordinate for simple contact check area | DFIF | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 386 |
| | Define tool coordinate system | DFTL | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 365 |
| | Define load coordinate system | DFWK | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 372 |
| | Set spline division distance | DIS | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 356 |
| | Acquiring Max. Collision Level | GCLX | | | | | | | | ○ (V1.10 or later) | | | | | 351 |
| | Set group axes | GRP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 346 |
| | Get defined coordinate for simple contact check area | GTIF | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 390 |
| | Get defined data of tool coordinate system | GTTL | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 370 |
| | Get defined data of load coordinate system | GTWK | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 377 |
| | Declare port to pause | HOLD | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 347 |
| | Change left arm of current arm system | LEFT | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 381 |
| | Set close distance | NBND | ○ | | | | | | | ○ | | | | | 395 |
| | Set offset | OFST | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 343 |
| | Set speed coefficient | OVRD | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 333 |
| | Set PUSH command distance, speed | PAPR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 358 |
| | Set output type | POTP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 357 |
| | Specify current arm of PTP target arm system | PTPD | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 384 |
| | Specify current arm of PTP target arm system | PTPE | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 385 |
| | Specify left arm of PTP target arm system | PTPL | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 383 |
| | Specify right arm of PTP target arm system | PTPR | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 382 |
| | Set quick return mode | QRTN | ○ | ○ | | | ○ | ○ | | ○ (V1.10 or later) | ○ | ○ | ○ (PC/PG only) | ○ | 359 |
| | Change right arm of current arm system | RIGH | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 380 |
| | Set sigmoid motion ratio | SCRV | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 339 |
| | Specify error type for simple contact check area | SEIF | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 389 |

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Actuator control declaration | Select tool coordinate system | SLTL | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 368 |
| | Select load coordinate system | SLWK | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 375 |
| | Specify output for simple interference check zone | SOIF | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 388 |
| | Set speed | VEL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 331 |
| | Set speed ratio for PTP operation | VELS | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 332 |
| | Specify VLMX speed | VLMX | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 355 |
| | Set tip load mass/inertial moment | WGHT | | | | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 391 |
| | Tip load condition setting 2 | WGT2 | | | | | | | ○ | ○ | | | | | 393 |
| | PTP Continuous Operation Mode Setting | CNTP | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 410 |
| Actuator control command | Move along arc (ARC2 is recommended) | ARC | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 441 |
| | Move along arc 2 (arc interpolation) | ARC2 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 424 |
| | Move along arc via specification of center position and center angle | ARCC | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 434 |
| | Move along arc via specification of end position and center angle | ARCD | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 432 |
| | Move three-dimensionally along arc | ARCS | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | ○ | 428 |
| | Change speed | CHVL | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 430 |
| | Move along circle (CIR2 is recommended) | CIR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 439 |
| | Move along circle 2 (arc interpolation) | CIR2 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 422 |
| | Move three-dimensionally along circle | CIRS | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 426 |
| | Return to home | HOME | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 398 |
| | Jog [FN, FF, BN, BF] | J□W□ | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 414 |
| | Move by direct value specification | MOVD | | | | | | | | | ○ | ○ | ○ (PC/PG only) | | 407 |
| | Move to specified position via interpolation | MOVL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 401 |
| | Move to specified position | MOVP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 399 |
| | Move incrementally by direct value specification | MVDI | | | | | | | | | ○ | ○ | ○ (PC/PG only) | | 408 |
| | Move to relative position via interpolation | MVLI | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 405 |
| | Move to relative position | MVPI | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 403 |
| | Move along path | PATH | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 409 |
| | Set positioning width | PBND | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 436 |
| | Wait for end of operation of axis using current program | PEND | ○ | | | | | | | | | | | | 443 |
| | Move along spline | PSPL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 418 |
| | Change push torque limit parameter | PTRQ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 421 |
| | Move by push motion | PUSH | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 419 |
| | Decelerate and stop axis | STOP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 417 |
| | Servo [ON, OF] | SV□□ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 397 |

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Actuator control command | Move relatively between positions on tool coordinate system via interpolation | TMLI | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 438 |
| | Move relatively between positions on tool coordinate system | TMPI | | | ○ | ○ | | | ○ | ○ | | | ○ | ○ (TTA only) | 437 |
| Structural IF | Declare end of IF | EDIF | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 447 |
| | Declare execution destination when IF command condition is not satisfied | ELSE | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 446 |
| | Compare [EQ, NE, GT, GE, LT, LE] | IF□□ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 444 |
| | Compare strings | IS□□ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 445 |
| Structural DO | Loop [EQ, NE, GT, GE, LT, LE] | DW□□ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 448 |
| | Declare end of DO | EDDO | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 451 |
| | Repeat DO | ITER | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 450 |
| | Pull out from DO | LEAV | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 449 |
| Multi-branching | Declare end | EDSL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 456 |
| | Declare branching destination when condition is not satisfied | OTHE | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 455 |
| | Declare start of multi-branching | SLCT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 452 |
| | Branch value [EQ, NE, GT, GE, LT, LE] | WH□□ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 453 |
| | Branch character string [EQ, NE] | WS□□ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 454 |
| System information acquisition | Get axis status | AXST | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 457 |
| | Get current arm system | GARM | | | ○ | ○ | | | ○ | ○ | | | ○ (PCX/PGX only) | | 460 |
| | Get program status | PGST | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 458 |
| | Get system status | SYST | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 459 |
| Zone | Wait for zone OFF, with AND | WZFA | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 463 |
| | Wait for zone OFF, with OR | WZFO | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 464 |
| | Wait for zone ON, with AND | WZNA | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 461 |
| | Wait for zone ON, with OR | WZNO | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 462 |
| Communication | Close channel | CLOS | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 466 |
| | Open channel | OPEN | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 465 |
| | Read from channel | READ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 467 |
| | Set end character | SCHA | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 474 |
| | Set read timeout value | TMRD | ○ | | ○ | | | | | ○ | | | | ○ (TT only) | 469 |
| | Set timeout value | TMRW | | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ (TTA only) | 471 |
| | Output to channel | WRIT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 473 |
| | Connected Destination IP address / Port Number Setting | IPCN | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 475 |
| String operation | Compare character strings | SCMP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 477 |
| | Copy character string | SCPY | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 476 |
| | Get character | SGET | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 478 |
| | Set length | SLEN | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 484 |
| | Set character | SPUT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 479 |

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| String operation | Convert character string; decimal | STR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 480 |
| | Convert character string; hexadecimal | STRH | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 481 |
| | Convert character string data; decimal | VAL | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 482 |
| | Convert character string data; hexadecimal | VALH | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 483 |
| Arch motion | Arch motion | ARCH | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 485 |
| | Declare arch motion Z-axis | ACHZ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 487 |
| | Set arch motion composition | AEXT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 489 |
| | Set arch triggers | ATRG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 488 |
| | Set arch motion Z-axis offset | OFAZ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 490 |
| Palletizing definition | Declare start of palletizing setting | BGPA | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 491 |
| | Declare end of palletizing setting | EDPA | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 492 |
| | Set palletizing Z-axis offset | OFPZ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 505 |
| | Set palletizing counts | PAPI | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 493 |
| | Set palletizing pattern | PAPN | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 494 |
| | Set palletizing points for 3-point or 4-point teaching | PAPS | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 498 |
| | Set palletizing pitches | PAPT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 496 |
| | Set palletizing axes | PASE | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 495 |
| | Set palletizing reference point | PAST | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 497 |
| | Set palletizing Z-axis | PCHZ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 502 |
| | Set palletizing composition | PEXT | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 504 |
| | Set zigzag | PSLI | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 501 |
| | Set palletizing arch triggers | PTRG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 503 |
| Palletizing operation | Get palletizing calculation data | PAPG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 511 |
| | Get palletizing angle | PARG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 510 |
| | Decrement palletizing position number by 1 | PDEC | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 508 |
| | Increment palletizing position number by 1 | PINC | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 507 |
| | Set palletizing position number directly | PSET | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 509 |
| | Get palletizing position number | PTNG | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 506 |
| Palletizing movement | Palletizing-point arch motion | PACH | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | 514 |
| | Move to palletizing points via interpolation | PMVL | ○ | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ (PC/PG only) | ○ | 513 |
| | Move to palletizing points via PTP | PMVP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 512 |
| Building of pseudo-ladder task | Change task level | CHPR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 516 |
| | Output relay for ladder | OUTR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 175 |
| | Timer relay for ladder | TIMR | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 175 |
| | Specify processing to be performed when input condition is not specified | TPCD | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 517 |
| | Task sleep | TSLP | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | 518 |

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Extended commands | Get motor current value | ECMD1 | | ○ | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ (V1.18 or later) | ○ (TTA only V1.22 or later) | 519 |
| | Get home sensor status | ECMD2 | | ○ | | | ○ | ○ | ○ | ○ | | | | | 520 |
| | Get overrun sensor status | ECMD3 | | ○ | | | ○ | ○ | ○ | ○ | | | | | 521 |
| | Get creep sensor status | ECMD4 | | ○ | | | ○ | ○ | ○ | ○ | | | | | 522 |
| | Get axis operation status | ECMD5 | | ○ | | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | 523 |
| | Current position acquirement on each axis system | ECMD6 | | | | ○ | | | ○ (V1.20 or later) | ○ | | | ○ | ○ (TTA only) | 524 |
| | Get total movement count | ECMD7 | | | | | | | | | | | ○ | ○ (TTA only) | 525 |
| | Get total mileage | ECMD8 | | | | | | | | | | | ○ | ○ (TTA only) | 526 |
| | Get position deviation | ECMD9 | | | | | | | | | | | ○ | ○ (TTA only) | 527 |
| | Acquirement of Overload Level | ECMD 10 | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 528 |
| | Acquirement of Encoder Overheated Level | ECMD 11 | | | | | | ○ (V1.10 or later) | | ○ (V1.10 or later) | | | | | 529 |
| | Get parameter | ECMD 20 | | ○ | | | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | 530 |
| | Set torque limit/ detection time for torque limit over error | ECMD 250 | | ○ | | ○ | ○ | ○ | ○ | ○ | | ○ (V0.52 or later) | Δ (With a condition) | | 532 |
| | Conversion from each axis coordinates to work coordinates in wrist unit equipped robot | ECMD 280 | | | | | | ○ | | | | | | | 535 |
| | Conversion from work coordinates to each axis coordinates in wrist unit equipped robot | ECMD 281 | | | | | | ○ | | | | | | | 536 |
| | Conversion from tool coordinates to work coordinates in wrist unit equipped robot | ECMD 282 | | | | | | ○ | | | | | | | 538 |
| | Conversion from each axis coordinates to work coordinates in wrist unit equipped robot | ECMD 290 | | | | | | ○ | | | | | | | 535 |
| | Conversion from work coordinates to each axis coordinates in wrist unit equipped robot | ECMD 291 | | | | | | ○ | | | | | | | 536 |
| | Conversion from tool coordinates to work coordinates in wrist unit equipped robot | ECMD 292 | | | | | | ○ | | | | | | | 538 |
| | User system error output | ECMD 300 | | | | | | ○ | | ○ | | | | | 540 |
| Vision System I/F Related | Declare use of Vision System | SLVS | | ○ | | | ○ | ○ | | | | | ○ (PC/PG only) | | 631 |
| | Image Caputure command | GTVD | | ○ | | | ○ | ○ | | | | | ○ (PC/PG only) | | 633 |
| Conveyor Tracking Related | Declare use of Conveyor Tracking | TRMD | | ○ | | ○ | ○ | ○ | ○ | ○ | | | | | 627 |
| | Image capturing and tracking command | TRAC | | ○ | | | ○ | ○ | ○ (Linear drive axis) | ○ (Linear drive axis) | | | | | 628 |
| Anti-Vibration Control Related | Anti-Vibration Control Parameter Set Select | NTCH | | ○ | | | ○ | ○ | ○ (Linear drive axis)○ | ○ (Linear drive axis) | | | | | 635 |

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Compliance Control | Compliance Mode Setting | COMP | | | | | | | | ○ (V1.10 or later) | | | | | 636 |
| | Compliance Mode Option Feature Setting (Searching Operation Setting) | SCLO 0 | | | | | | | | ○ (V1.10 or later) | | | | | 638 |
| | Compliance Mode Option Feature Setting (J1 & J2-Axes Torque Limit Mode Setting) | SCLO 1 | | | | | | | | ○ (V1.10 or later) | | | | | 640 |
| | Compliance Gain Setting | SCLG | | | | | | | | ○ (V1.10 or later) | | | | | 641 |

## RC Gateway Function Commands (Controllers with Gateway Function Only)

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RC axis position operation | Assign RC axis position to variable 199 | RPGT | | ○ | | ○ | ○ | | ○ | | | | | | 541 |
| | Assign variable 199 to RC axis position | RPPT | | ○ | | ○ | ○ | | ○ | | | | | | 542 |
| | Clear RC axis position data | RPCR | | ○ | | ○ | ○ | | ○ | | | | | | 543 |
| | Copy RC axis position data | RPCP | | ○ | | ○ | ○ | | ○ | | | | | | 544 |
| | Read current RC axis position | RPRD | | ○ | | ○ | ○ | | ○ | | | | | | 545 |
| | Read current RC axis position (1 axis, direct) | RPRQ | | ○ | | ○ | ○ | | ○ | | | | | | 546 |
| | Assign variable 199 to RC axis position speed | RPVL | | ○ | | ○ | ○ | | ○ | | | | | | 547 |
| | Assign variable 199 to RC axis position acceleration/ deceleration | RPAD | | ○ | | ○ | ○ | | ○ | | | | | | 548 |
| | Assign variable 199 to RC axis position positioning width | RPIP | | ○ | | ○ | ○ | | ○ | | | | | | 549 |
| | Assign variable 199 to RC axis position current-limiting value | RPTQ | | ○ | | ○ | ○ | | ○ | | | | | | 550 |
| | Assign RC axis position speed to variable 199 | RGVL | | ○ | | ○ | ○ | | ○ | | | | | | 551 |
| | Assign RC axis position acceleration/ deceleration to variable 199 | RGAD | | ○ | | ○ | ○ | | ○ | | | | | | 552 |
| | Assign RC axis position positioning width to variable 199 | RGIP | | ○ | | ○ | ○ | | ○ | | | | | | 553 |
| | Assign RC axis position current-limiting value to variable 199 | RGTQ | | ○ | | ○ | ○ | | ○ | | | | | | 554 |
| RC actuator control command | Set RC axis pattern | RAXS | | ○ | | ○ | ○ | | ○ | | | | | | 555 |
| | Turn RC axis servo ON | RSON | | ○ | | ○ | ○ | | ○ | | | | | | 556 |
| | Turn RC axis servo OFF | RSOF | | ○ | | ○ | ○ | | ○ | | | | | | 557 |
| | Return RC axis to its home | RHOM | | ○ | | ○ | ○ | | ○ | | | | | | 558 |
| | Move by RC axis position specification | RMVP | | ○ | | ○ | ○ | | ○ | | | | | | 559 |
| | Incremental move by RC axis position specification | RMPI | | ○ | | ○ | ○ | | ○ | | | | | | 560 |
| | Move by RC axis direct specification | RMVD | | ○ | | ○ | ○ | | ○ | | | | | | 561 |
| | Incremental move by RC axis direct specification | RMDI | | ○ | | ○ | ○ | | ○ | | | | | | 562 |
| | Move by RC axis push-motion operation | RPUS | | ○ | | ○ | ○ | | ○ | | | | | | 563 |
| | Decelerate RC axis to stop | RSTP | | ○ | | ○ | ○ | | ○ | | | | | | 564 |
| RC axis information acquisition | Get RC axis status | RCST | | ○ | | ○ | ○ | | ○ | | | | | | 565 |

## Extension Motion Control Function Related Commands

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Extension motion control board input operations | Clear input counter record for extension motion control | XCRP | | ○ | | | ○ | | | | | | | | 567 |
| | Acquire current record of extension motion control input counter | XGTP | | ○ | | | ○ | | | | | | | | 568 |
| Extension motion control board axis position operations | Read extension motion control axis position data | XPGT | | ○ | | | ○ | ○ | | ○ | | | | | 569 |
| | Write extension motion control axis position data | XPPT | | ○ | | | ○ | ○ | | ○ | | | | | 570 |
| | Erase extension motion control axis position data | XPCR | | ○ | | | ○ | ○ | | ○ | | | | | 571 |
| | Copy extension motion control axis position data | XPCP | | ○ | | | ○ | ○ | | ○ | | | | | 572 |
| | Read extension motion control axis current command position | XPRD | | ○ | | | ○ | ○ | | ○ | | | | | 573 |
| | Read extension motion control axis current command position (single-axis direct) | XPRQ | | ○ | | | ○ | ○ | | ○ | | | | | 574 |
| | Write extension motion control axis speed data | XPVL | | ○ | | | ○ | ○ | | ○ | | | | | 575 |
| | Write extension motion control axis acceleration data | XPAC | | ○ | | | ○ | ○ | | ○ | | | | | 576 |
| | Write extension motion control axis deceleration data | XPDC | | ○ | | | ○ | ○ | | ○ | | | | | 577 |
| | Write extension motion control axis positioning complete width data | XPIP | | ○ | | | ○ | ○ | | ○ | | | | | 578 |
| | Read extension motion control axis speed data | XGVL | | ○ | | | ○ | ○ | | ○ | | | | | 579 |
| | Read extension motion control axis acceleration data | XGAC | | ○ | | | ○ | ○ | | ○ | | | | | 580 |
| | Read extension motion control axis deceleration data | XGDC | | ○ | | | ○ | ○ | | ○ | | | | | 581 |
| | Read extension motion control axis positioning width data | XGIP | | ○ | | | ○ | ○ | | ○ | | | | | 582 |
| Extension motion control board axis actuator control declarations | Set extension motion control axis patterns (0 to 15 axes) | XAXS | | ○ | | | ○ | ○ | | ○ | | | | | 583 |
| | Set extension motion control axis patterns (16 to 31 axes) | XA16 | | | | | | ○ | | ○ | | | | | 584 |
| Extension motion control board axis actuator control commands | Extension motion control axis servo ON | XSON | | ○ | | | ○ | ○ | | ○ | | | | | 585 |
| | Extension motion control axis servo OFF | XSOF | | ○ | | | ○ | ○ | | ○ | | | | | 586 |
| | Return extension motion control axis to home position | XHOM | | ○ | | | ○ | ○ | | ○ | | | | | 587 |
| | Move extension motion control axis to indicated position | XMVP | | ○ | | | ○ | ○ | | ○ | | | | | 588 |
| | Perform extension motion control axis position relative movement | XMPI | | ○ | | | ○ | ○ | | ○ | | | | | 589 |
| | Move extension motion control axis for position indicated interpolation | XMVL | | ○ | | | ○ | ○ | | ○ | | | | | 590 |

| Category | Function | Command | XSEL -J/K/ KE/KT/KET | XSEL -P/Q/ PCT/QCT | XSEL -JX/KX/ KETX | XSEL -PX/QX | XSEL -R/S | XSEL -RA/SA | XSEL -RX/SX/ RXD/SXD | XSEL -RAX/SAX/ RAXD/SAXD | ASEL PSEL | SSEL | MSEL | TT/TTA | Page |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Extension motion control board axis actuator control commands | Move extension motion control axis for position relative interpolation | XMLI | | ○ | | | ○ | ○ | | ○ | | | | | 591 |
| | Move extension motion control axis to directly indicated absolute position | XMVD | | ○ | | | ○ | ○ | | ○ | | | | | 592 |
| | Move extension motion control axis to directly indicated relative position | XMDI | | ○ | | | ○ | ○ | | ○ | | | | | 593 |
| | Extension motion control axis path operation | XPTH | | | | | | ○ | | ○ | | | | | 594 |
| | Perform extension motion control axis jog operation | XJ□□ | | ○ | | | ○ | ○ | | ○ | | | | | 596 |
| | Waiting for extension motion control axis to finish positioning operation of axis used by self-program | XPED | | ○ | | | ○ | ○ | | ○ | | | | | 597 |
| | Cancel operation of extension motion control axis | XSTP | | ○ | | | ○ | ○ | | ○ | | | | | 598 |
| | Waiting for extension motion control axis positioning complete signal to be turned ON | XWIP | | ○ | | | ○ | ○ | | ○ | | | | | 599 |
| | Start synchronizing extension motion control axis electronic cam (indicating main axis) | XCAS | | ○ | | | ○ | | | | | | | | 601 |
| Extension motion control board axis actuator control commands | Move extension motion control axis individual electronic cam (indicating time) | XCTM | | ○ | | | ○ | ○ | | | | | | | 612 |
| | Start synchronizing of extension motion control axis electronic shaft | XSFS | | ○ | | | ○ | ○ | | | | | | | 614 |
| | Cancel operation of extension motion control axis | XSYE | | ○ | | | ○ | ○ | | | | | | | 617 |
| Extension motion control board axis status acquirement | Acquire extension motion control axis status | XAST | | ○ | | | ○ | ○ | | ○ | | | | | 619 |
| Arch Motion related | Extension motion control axis arch motion | XACH | | | | | | ○ | | ○ | | | | | 622 |
| | Extension motion control axis arch motion Z-axis declaration | XACZ | | | | | | ○ | | ○ | | | | | 624 |
| | Extension motion control axis arch motion composition setting | XAEX | | | | | | ○ | | ○ | | | | | 624 |
| | Extension motion control axis arch trigger setting | XATG | | | | | | ○ | | ○ | | | | | 625 |
| | Extension motion control axis arch motion Z-axis offset setting | XOAZ | | | | | | ○ | | ○ | | | | | 626 |

# Safety Guide

"Safety Guide" has been written to use the machine safely and so prevent personal injury or property damage beforehand. Make sure to read it before the operation of this product.

## Safety Precautions for Our Products

The common safety precautions for the use of any of our robots in each operation.

| No. | Operation Description | Description |
|---|---|---|
| 1 | Model Selection | ● This product has not been planned and designed for the application where high level of safety is required, so the guarantee of the protection of human life is impossible. Accordingly, do not use it in any of the following applications.<br>1) Medical equipment used to maintain, control or otherwise affect human life or physical health.<br>2) Mechanisms and machinery designed for the purpose of moving or transporting people (For vehicle, railway facility or air navigation facility)<br>3) Important safety parts of machinery (Safety device, etc.)<br>● Do not use the product outside the specifications. Failure to do so may considerably shorten the life of the product.<br>● Do not use it in any of the following environments.<br>1) Location where there is any inflammable gas, inflammable object or explosive<br>2) Place with potential exposure to radiation<br>3) Location with the ambient temperature or relative humidity exceeding the specification range<br>4) Location where radiant heat is added from direct sunlight or other large heat source<br>5) Location where condensation occurs due to abrupt temperature changes<br>6) Location where there is any corrosive gas (sulfuric acid or hydrochloric acid)<br>7) Location exposed to significant amount of dust, salt or iron powder<br>8) Location subject to direct vibration or impact<br>● For an actuator used in vertical orientation, select a model which is equipped with a brake. If selecting a model with no brake, the moving part may drop when the power is turned OFF and may cause an accident such as an injury or damage on the work piece. |

| No. | Operation Description | Description |
|---|---|---|
| 2 | Transportation | ● When carrying a heavy object, do the work with two or more persons or utilize equipment such as crane.<br>● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers.<br>● When in transportation, consider well about the positions to hold, weight and weight balance and pay special attention to the carried object so it would not get hit or dropped.<br>● Transport it using an appropriate transportation measure.<br>The actuators available for transportation with a crane have eyebolts attached or there are tapped holes to attach bolts. Follow the instructions in the instruction manual for each model.<br>● Do not step or sit on the package.<br>● Do not put any heavy thing that can deform the package, on it.<br>● When using a crane capable of 1t or more of weight, have an operator who has qualifications for crane operation and sling work.<br>● When using a crane or equivalent equipments, make sure not to hang a load that weighs more than the equipment's capability limit.<br>● Use a hook that is suitable for the load. Consider the safety factor of the hook in such factors as shear strength.<br>● Do not get on the load that is hung on a crane.<br>● Do not leave a load hung up with a crane.<br>● Do not stand under the load that is hung up with a crane. |
| 3 | Storage and Preservation | ● The storage and preservation environment conforms to the installation environment. However, especially give consideration to the prevention of condensation.<br>● Store the products with a consideration not to fall them over or drop due to an act of God such as earthquake. |
| 4 | Installation and Start | (1) Installation of Robot Main Body and Controller, etc.<br>● Make sure to securely hold and fix the product (including the work part). A fall, drop or abnormal motion of the product may cause a damage or injury.<br>Also, be equipped for a fall-over or drop due to an act of God such as earthquake.<br>● Do not get on or put anything on the product. Failure to do so may cause an accidental fall, injury or damage to the product due to a drop of anything, malfunction of the product, performance degradation, or shortening of its life.<br>● When using the product in any of the places specified below, provide a sufficient shield.<br>1) Location where electric noise is generated<br>2) Location where high electrical or magnetic field is present<br>3) Location with the mains or power lines passing nearby<br>4) Location where the product may come in contact with water, oil or chemical droplets |

| No. | Operation Description | Description |
|---|---|---|
| 4 | Installation and Start | (2) Cable Wiring<br>● Use our company's genuine cables for connecting between the actuator and controller, and for the teaching tool.<br>● Do not scratch on the cable. Do not bend it forcibly. Do not pull it. Do not coil it around. Do not insert it. Do not put any heavy thing on it. Failure to do so may cause a fire, electric shock or malfunction due to leakage or continuity error.<br>● Perform the wiring for the product, after turning OFF the power to the unit, so that there is no wiring error.<br>● When the direct current power (+24V) is connected, take the great care of the directions of positive and negative poles. If the connection direction is not correct, it might cause a fire, product breakdown or malfunction.<br>● Connect the cable connector securely so that there is no disconnection or looseness. Failure to do so may cause a fire, electric shock or malfunction of the product.<br>● Never cut and/or reconnect the cables supplied with the product for the purpose of extending or shortening the cable length. Failure to do so may cause the product to malfunction or cause fire. |
| | | (3) Grounding<br>● The grounding operation should be performed to prevent an electric shock or electrostatic charge, enhance the noise-resistance ability and control the unnecessary electromagnetic radiation.<br>● For the ground terminal on the AC power cable of the controller and the grounding plate in the control panel, make sure to use a twisted pair cable with wire thickness $0.5mm^2$ (AWG20 or equivalent) or more for grounding work. For security grounding, it is necessary to select an appropriate wire thickness suitable for the load. Perform wiring that satisfies the specifications (electrical equipment technical standards).<br>● Perform Class D Grounding (former Class 3 Grounding with ground resistance 100Ω or below). |

| No. | Operation Description | Description |
|---|---|---|
| 4 | Installation and Start | (4) Safety Measures<br>● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers.<br>● When the product is under operation or in the ready mode, take the safety measures (such as the installation of safety and protection fence) so that nobody can enter the area within the robot's movable range. When the robot under operation is touched, it may result in death or serious injury.<br>● Make sure to install the emergency stop circuit so that the unit can be stopped immediately in an emergency during the unit operation.<br>● Take the safety measure not to start up the unit only with the power turning ON. Failure to do so may start up the machine suddenly and cause an injury or damage to the product.<br>● Take the safety measure not to start up the machine only with the emergency stop cancellation or recovery after the power failure. Failure to do so may result in an electric shock or injury due to unexpected power input.<br>● When the installation or adjustment operation is to be performed, give clear warnings such as "Under Operation; Do not turn ON the power!" etc. Sudden power input may cause an electric shock or injury.<br>● Take the measure so that the work part is not dropped in power failure or emergency stop.<br>● Wear protection gloves, goggle or safety shoes, as necessary, to secure safety.<br>● Do not insert a finger or object in the openings in the product. Failure to do so may cause an injury, electric shock, damage to the product or fire.<br>● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity. |
| 5 | Teaching | ● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers.<br>● Perform the teaching operation from outside the safety protection fence, if possible. In the case that the operation is to be performed unavoidably inside the safety protection fence, prepare the "Stipulations for the Operation" and make sure that all the workers acknowledge and understand them well.<br>● When the operation is to be performed inside the safety protection fence, the worker should have an emergency stop switch at hand with him so that the unit can be stopped any time in an emergency.<br>● When the operation is to be performed inside the safety protection fence, in addition to the workers, arrange a watchman so that the machine can be stopped any time in an emergency. Also, keep watch on the operation so that any third person can not operate the switches carelessly.<br>● Place a sign "Under Operation" at the position easy to see.<br>● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity.<br>* Safety protection Fence : In the case that there is no safety protection fence, the movable range should be indicated. |

| No. | Operation Description | Description |
|---|---|---|
| 6 | Trial Operation | ● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers.<br>● After the teaching or programming operation, perform the check operation one step by one step and then shift to the automatic operation.<br>● When the check operation is to be performed inside the safety protection fence, perform the check operation using the previously specified work procedure like the teaching operation.<br>● Make sure to perform the programmed operation check at the safety speed. Failure to do so may result in an accident due to unexpected motion caused by a program error, etc.<br>● Do not touch the terminal block or any of the various setting switches in the power ON mode. Failure to do so may result in an electric shock or malfunction. |
| 7 | Automatic Operation | ● Check before starting the automatic operation or rebooting after operation stop that there is nobody in the safety protection fence.<br>● Before starting automatic operation, make sure that all peripheral equipment is in an automatic-operation-ready state and there is no alarm indication.<br>● Make sure to operate automatic operation start from outside of the safety protection fence.<br>● In the case that there is any abnormal heating, smoke, offensive smell, or abnormal noise in the product, immediately stop the machine and turn OFF the power switch. Failure to do so may result in a fire or damage to the product.<br>● When a power failure occurs, turn OFF the power switch. Failure to do so may cause an injury or damage to the product, due to a sudden motion of the product in the recovery operation from the power failure. |

| No. | Operation Description | Description |
|---|---|---|
| 8 | Maintenance and Inspection | ● When the work is carried out with 2 or more persons, make it clear who is to be the leader and who to be the follower(s) and communicate well with each other to ensure the safety of the workers.<br>● Perform the work out of the safety protection fence, if possible. In the case that the operation is to be performed unavoidably inside the safety protection fence, prepare the “Stipulations for the Operation” and make sure that all the workers acknowledge and understand them well.<br>● When the work is to be performed inside the safety protection fence, basically turn OFF the power switch.<br>● When the operation is to be performed inside the safety protection fence, the worker should have an emergency stop switch at hand with him so that the unit can be stopped any time in an emergency.<br>● When the operation is to be performed inside the safety protection fence, in addition to the workers, arrange a watchman so that the machine can be stopped any time in an emergency. Also, keep watch on the operation so that any third person can not operate the switches carelessly.<br>● Place a sign “Under Operation” at the position easy to see.<br>● For the grease for the guide or ball screw, use appropriate grease according to the Instruction Manual for each model.<br>● Do not perform the dielectric strength test. Failure to do so may result in a damage to the product.<br>● When releasing the brake on a vertically oriented actuator, exercise precaution not to pinch your hand or damage the work parts with the actuator dropped by gravity.<br>● The slider or rod may get misaligned OFF the stop position if the servo is turned OFF. Be careful not to get injured or damaged due to an unnecessary operation.<br>● Pay attention not to lose the cover or untightened screws, and make sure to put the product back to the original condition after maintenance and inspection works.<br>Use in incomplete condition may cause damage to the product or an injury.<br>* Safety protection Fence : In the case that there is no safety protection fence, the movable range should be indicated. |
| 9 | Modification and Dismantle | ● Do not modify, disassemble, assemble or use of maintenance parts not specified based at your own discretion. |
| 10 | Disposal | ● When the product becomes no longer usable or necessary, dispose of it properly as an industrial waste.<br>● When removing the actuator for disposal, pay attention to drop of components when detaching screws.<br>● Do not put the product in a fire when disposing of it.<br>The product may burst or generate toxic gases. |
| 11 | Other | ● Do not come close to the product or the harnesses if you are a person who requires a support of medical devices such as a pacemaker. Doing so may affect the performance of your medical device.<br>● See Overseas Specifications Compliance Manual to check whether complies if necessary.<br>● For the handling of actuators and controllers, follow the dedicated instruction manual of each unit to ensure the safety. |

# Alert Indication

The safety precautions are divided into “Danger”, “Warning”, “Caution” and “Notice” according to the warning level, as follows, and described in the Instruction Manual for each model.

| Level | Degree of Danger and Damage | Symbol |
|---|---|---|
| Danger | This indicates an imminently hazardous situation which, if the product is not handled correctly, will result in death or serious injury. | Danger |
| Warning | This indicates a potentially hazardous situation which, if the product is not handled correctly, could result in death or serious injury. | Warning |
| Caution | This indicates a potentially hazardous situation which, if the product is not handled correctly, may result in minor injury or property damage. | Caution |
| Notice | This indicates lower possibility for the injury, but should be kept to use this product properly. | Notice |

# 1. Preparation in Advance

SEL language is the simplest type of language in many existing robot languages.
Even though SEL language is an interpreter program, it enables to perform high level controls in simple expression ways.
In this manual, describes how to use SEL language, explanations of command language, examples of how to create programs for each actuator, etc.

In this section, explains what are needed to be prepared beforehand to start programming, or the things that you need to know for programming.

## 1.1 Related Manuals

Please make sure to refer also to the instruction manuals for the controller and accessories that you intend to use. Listed below are the related instruction manuals.

| No. | Name | Manual No. |
|---|---|---|
| 1 | XSEL-J/K Controller Instruction Manual | ME0116 |
| 2 | XSEL-KT Controller Instruction Manual | ME0134 |
| 3 | XSEL-JX/KX Controller Instruction Manual | ME0119 |
| 4 | XSEL-P/Q/PCT/QCT Controller Instruction Manual | ME0148 |
| 5 | XSEL-PX/QX Controller Instruction Manual | ME0152 |
| 6 | PSEL Controller Instruction Manual | ME0172 |
| 7 | ASEL Controller Instruction Manual | ME0165 |
| 8 | SSEL Controller Instruction Manual | ME0154 |
| 9 | TT Controller | ME0149 |
| 10 | PC Software IA-101-X-MW/IA-101-X-USBMW | ME0154 |
| 11 | Teaching Pendant SEL-T/TD/TG | ME0183 |
| 12 | Teaching Pendant IA-T-X/XD | ME0160 |
| 13 | Touch Panel Teaching TB-01/01D/01DR Applicable for Program Controller Instruction Manual | ME0325 |
| 14 | Touch Panel Teaching TB-02/02D Applicable for Program Controller Instruction Manual | ME0356 |
| 15 | DeviceNet Instruction Manual | ME0124 |
| 16 | CC-Link Instruction Manual | ME0123 |
| 17 | PROFIBUS Instruction Manual | ME0153 |
| 18 | XSEL Ethernet Instruction Manual | ME0140 |
| 19 | XSEL Controller RC Gateway Function Instruction Manual | ME0188 |
| 20 | XSEL-P/Q/PCT/QCT Controller Electronic Cam Function Instruction Manual | ME0246 |
| 21 | OMRON Vision Sensor Tracking Instruction Manual | ME0237 |
| 22 | Keyence Vision Sensor Tracking Instruction Manual | ME0238 |
| 23 | Cognex Vision Sensor Tracking Instruction Manual | ME0239 |
| 24 | XSEL-P/Q/PCT/QCT Controller Vision System I/F Function Instruction Manual | ME0264 |
| 25 | XSEL-R/S/RX/SX/RXD/SXD Controller Instruction Manual | ME0308 |
| 26 | XSEL-RA/SA/RAX/SAX/RAXD/SAXD Controller Instruction Manual | ME0359 |
| 27 | Tabletop Robot TTA Instruction Manual | ME0320 |
| 28 | MSEL Instruction Manual | ME0336 |

## 1.2 Programming Tool

To create a program with SEL language, it is necessary to prepare a dedicated teaching pendant or PC software provided by IAI.
Please confirm in the table below that the controller you intend to use complies with the programming tool that you have.

| No. | Item | Controller Model / Model Code of Programming Tool | XSEL-J/K/KE/KT/KET | XSEL-P/PCT/R | XSEL-Q/QCT/S | XSEL-JX/KX/KETX | XSEL-PX/RX/RXD | XSEL-QX/SX/SXD | XSEL-RA/RAX/RAXD | XSEL-SA/SAX/SAXD | ASEL | PSEL | SSEL | TT/TTA | MSEL |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | PC software (with RS232C cable + emergency stop box) | IA-101-X-MW | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | ○ | |
| 2 | PC software (with USB conversion adapter + RS232C cable + emergency stop box) | IA-101-X-USBMW | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | | | | ○ | |
| 3 | PC software (with RS232C cable + emergency stop box + connector conversion cable) | IA-101-X-MW-J | | | | | | | | | ○ | ○ | ○ | | ○ |
| 4 | PC software (with USB cable + dummy plug) | IA-101-X-USB | | | | | | | | | ○ | ○ | ○ | | ○ |
| 5 | PC software (with safety category 4 cable + emergency stop box) | IA-101-XA-MW | | | ○ | | | ○ | | ○ | | | | | |
| 6 | Teaching pendant | SEL-T | ○ (J is excluded) | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | ○ | ○ (TTA has some restrictions) | |
| 7 | Teaching pendant (with deadman switch) | SEL-TD | ○ (J is excluded) | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | ○ | ○ (TTA has some restrictions) | |
| 8 | Teaching pendant (with TP adapter for Safety Categories) | SEL-TG | ○ | ○ | ○ | ○ | ○ | ○ | | | ○ | ○ | ○ | ○ (TTA has some restrictions) | |
| 9 | Teaching pendant | IA-T-X | ○ | ○ (Q is excluded) | ○ | ○ | ○ | ○ (QX is excluded) | | | ○ | ○ | ○ | ○ (TTA is excluded) | |
| 10 | Teaching pendant (with deadman switch) | IA-T-XD | ○ | ○ (Q is excluded) | ○ | ○ | ○ | ○ (QX is excluded) | | | ○ | ○ | ○ | ○ (TTA is excluded) | |
| 11 | Touch panel teaching | TB-01 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 12 | Touch panel teaching (with deadman switch) | TB-01D | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 13 | Touch panel teaching | TB-02 | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |
| 14 | Touch panel teaching (with deadman switch) | TB-02D | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

○: Applicable, Blank: Not applicable

## 1.3 PC Operational Environment

If you use the PC software, make sure your PC meets the following specifications before installing the software. [Refer to PC Software Instruction Manual for how to install it.]
Also, confirm in the next section that it is applicable for the controller that you intend to use.

1) Operating System (OS)
   Windows XP SP2 or later, Windows Vista, Windows 7, Windows 8/8.1
2) Main Memory
   It should possess memory capacity necessary to operate Windows®.
3) Display Monitor
   XGA or more.
4) Hard Disk
   Hard disk with free space of 30MB or more
   (This software is to be used with being installed in the hard disk.)
5) Serial Port
   There should be 1 unit of RS232C port that is capable for the communication speed setting of 9600bps or more.
   (Note) This is for the case the model code of PC software is IA-101-*-MW.
6) USB Port
   There should be 1 unit of USB port with its version 1.1 or more.
   (Note) This is for the case the model code of PC software is IA-101-*-USBMW.
7) Keyboard
   It should comply with the PC main unit. (PC/AT compatible keyboard)
8) Pointing Device
   It should be operated in Windows® OS.
9) Drive Device
   The PC should possess a CD-ROM drive device that complies with the PC or a compatible drive device that can read CD-ROM.

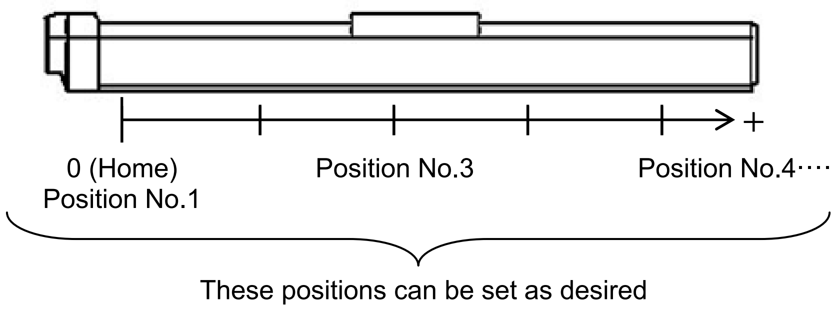
* SEL language is available on the following controllers.
   1) XSEL (all types)
   2) ASEL
   3) PSEL
   4) SSEL
   5) TT/TTA
   6) MSEL

## 1.4 Axes on Each Actuator and Precautions

In this section, explains the construction of axis number of each actuator on the 3-dimensional coordinate system, X, Y and Z.

### 1.4.1 Single-Direction Axis

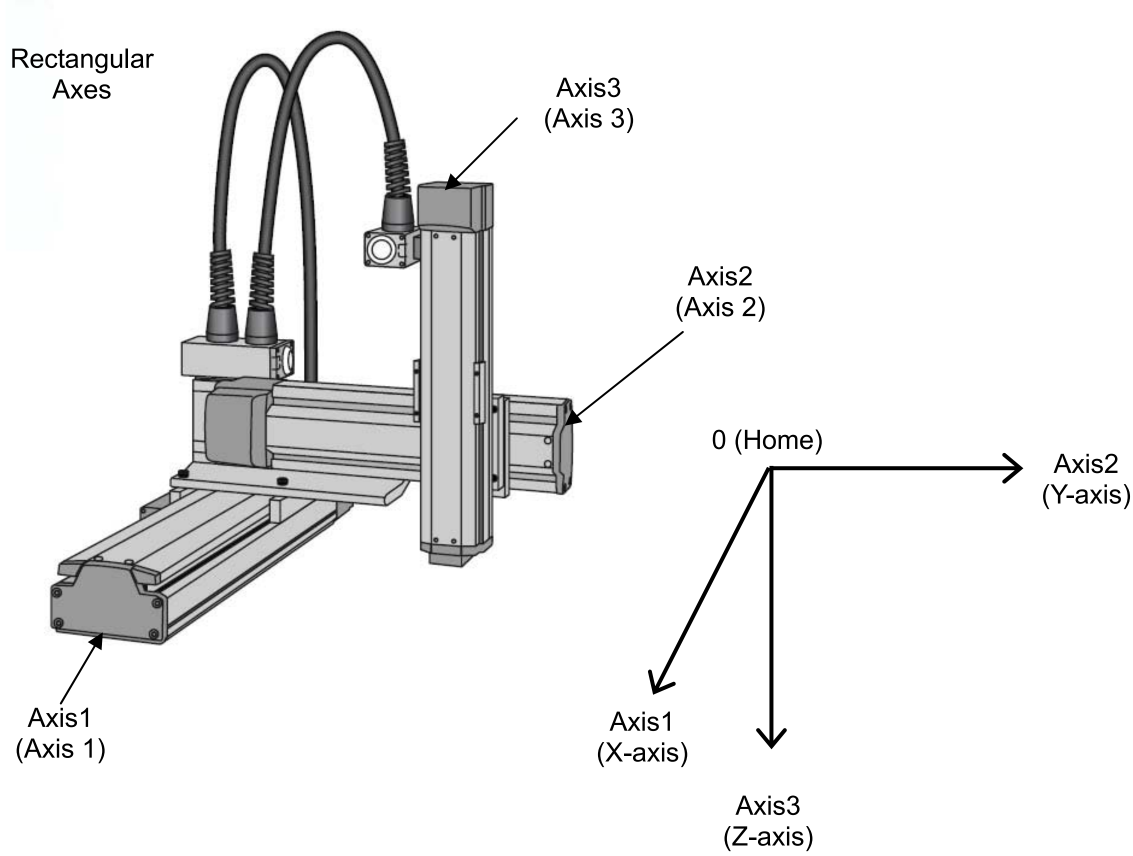
The coordinate value from the home corresponds to 0mm in position data.
Positions from the home represent position data.
The direction is reversed if the actuator is of reversed-home specification.



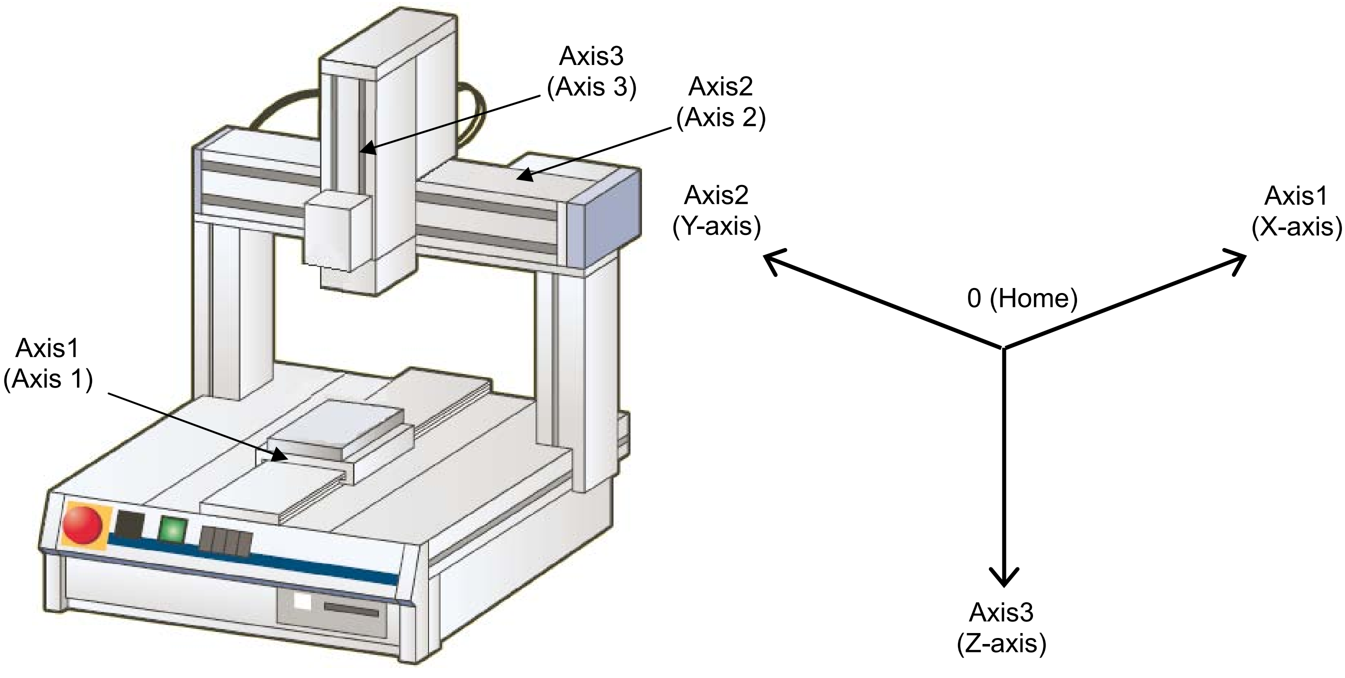

Axis Number of the position table will be that of the connector that is connected physically to the actuator motor and encoder cable. [Refer to the instruction manual of the controller for the details.]

### 1.4.2 Rectangular Axes, TT/TTA

The coordinate value from the home of each axis corresponds to 0mm in position data.
With each axis, positions from the home represent position data.

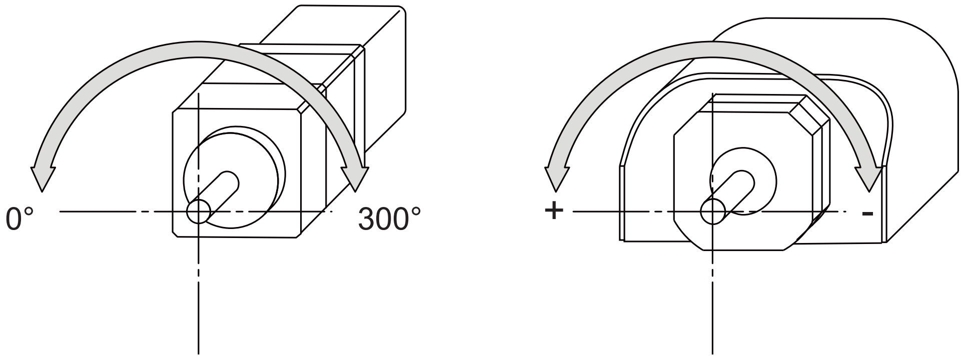



Axis Number is that the actuator motor and encoder cable is connected physically.
The coordinate system shown in the figure below is when each actuator motor and encoder cable is connected to the Axis Number in bracket.
Refer to the instruction manual of each controller for the details.



### 1.4.3 Rotational Axis

The coordinate value from the home corresponds to 0° in position data.
Rotational angles from the home represent position data.

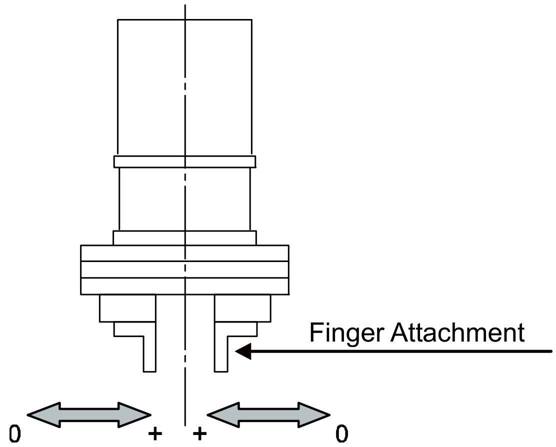



To the rotation axes, there is the finite stroke type that operates within the established angular range and the infinite stroke type that can rotate for a number of times in the indicated direction.
Refer to the instruction manual of each actuator for the details.

### 1.4.4 Gripper

The coordinate value (opening side) from the home corresponds to 0mm in position data.
1/2 stroke[*1] from the home represent position data.

*1 Stroke : Distance between both grippers




### 1.4.5 TTA, MSEL-PC/PG

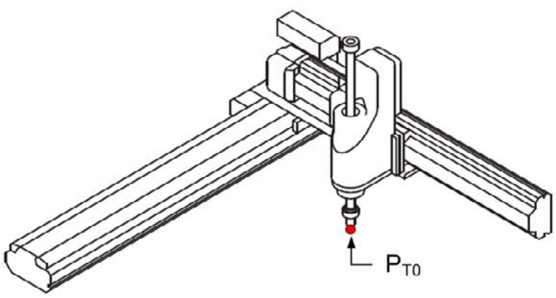
TTA : V 2.00 or later
MSEL-P/C/PG : V 2.00 or later

[1] Overview

In the groups of axes can be structured by indicating linear axes in parameters, and the work coordinate system and tool coordinate system can be determined to these axis groups.
The structured axis groups are called “coordinate system definition unit”, and each axis constructing the unit is called “coordinate system definition unit axis”.

The coordinate system definition unit axis usually makes positioning of the tool attachment datum point [(*1)] or tool tip on the selected work coordinate system. However, it makes positioning of SEL language commands TMPI, TMLI and XY (tool) coordinate system jog on the tool coordinate system.

*1 Tool Attachment Datum Point ($P_{T0}$) : It is a point of datum when a tool is attached. If there is R-axis in the coordinate system constructing axes, it will be one point on the rotation axis of R-axis. If there is no R-axis in the coordinate system constructing axes, it will be one point on the axis that a tool is attached.



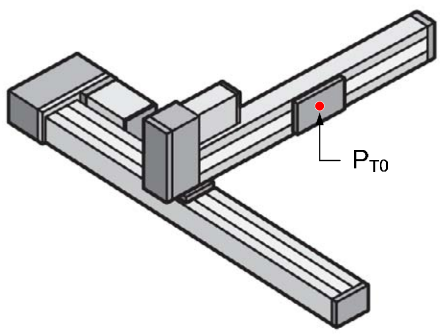




Figure : Example for Tool Attachment Datum Point
(Left : Combination for Four-Axis, Right : Combination for Two-Axis)

[2] Setting of parameters

Shown below, describes how to set the parameters necessary for using the work and tool coordinate system features on the linear axis.

- By setting All Axes Parameter No. 55 “Coordinate System Definition 1 Control” to “1h”, the coordinate system definition unit becomes effective.

| No. | Parameter name | Initial value (Reference) | Input range | Unit | Access right | Remarks |
|---|---|---|---|---|---|---|
| 55 | Coordinate System Definition 1 Control | 0H | 0 ~ FFFFFFFFH | | F | Bits 0-3 : Unit Valid Indication (0: Invalid, 1: Valid)<br>Bits 4-31 : Reservation |

- Indicate the physical axis applicable for the coordinate axes (X-axis, Y-axis, Z-axis and R-axis) in the coordinate system in All Axes Parameter No. 56 “Coordinate System Definition 1 Constructing Axis Setting”.

| No. | Parameter name | Initial value (Reference) | Input range | Unit | Access right | Remarks |
|---|---|---|---|---|---|---|
| 56 | Coordinate System Definition 1 Constructing Axis Setting | 0H | 0 ~ FFFFFFFFH | | F | Bits 0 to 3: X Indicated Axis Number<br>Bits 4 to 7: Y Indicated Axis Number<br>Bits 8 to 11: Z Indicated Axis Number<br>Bits 12 to 15: R Indicated Axis Number<br>Bits 16 to 31: Reservation<br>(No applicable definition axis when Axis No. = 0) |

Stated below are the caution items regarding the setting of All Axes Parameter No. 56. “5C0 Coordinate System Setting Error” will occur when there is an error in setting related to these caution items described below.

- The patterns of constructing axes that are available to indicate should be either of “XY”, “XYZ”, “XYZR” and “XYR”.
- Duplicated indication of one physical axis is not allowed.
- It is necessary that the axes indicated as the X, Y and Z-axes are linear driving axes (Each Axis Parameter No. 1 = “0”), and that the axis indicated as R-axis should be rotary driving axis (Each Axis Parameter No. 1 = “1”).
- The following axes cannot be indicated.
  [Synchronized Slave Axes, Infinite Stroke Axes, Shortcut Control Valid Axes]

> ⚠ Caution : · There may be a case that an error such as “D41 Coordinate System Data Management Domain ID Error” occurs after conducting parameter writing → reboot when the coordinate system definition unit gets activated by setting All Axes Parameter No. 55 for the first time.
> In such a case, initialize the memory for the “coordinate system definition data” using the PC software or a teaching pendant and then reboot the controller.
> · There will be limitation in some part of operation if indicated to the coordinate system definition unit axis. (Refer to [4] for details.)

• In case there is an axis indicated as the R-axis in All Axes Parameter No. 56, establish the setting in All Axes Parameter No. 57 "Coordinate System Definition 1 R-Axis Coordinates Direction Setting".

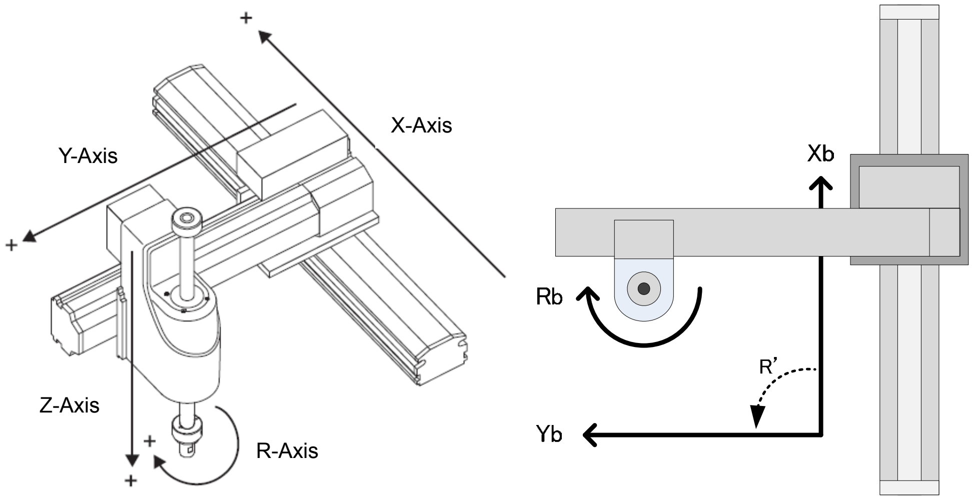
| No. | Parameter name | Initial value (Reference) | Input range | Unit | Access right | Remarks |
|---|---|---|---|---|---|---|
| 57 | Coordinate System Definition 1 R-Axis Coordinates Direction Setting | 0 | 0 ~ 1 | | F | Relation between direction of rotation from X-axis to Y-axis and R-axis direction in base coordinate system<br>0: Same direction<br>1: Opposite direction |

(All Axes Parameter No. 57 is the parameter to define the relations of X-axis, Y-axis and R-axis in the base coordinate system.
In case this parameter is not established correctly, the axes will not operate as expected. Pay attention when establish the setting for this parameter as the controller cannot judge right or wrong for the contents of the setting.)

Establish the setting as described below for the value in All Axes Parameter No. 57.

・When "Rb Positive Direction" is the same as "Rotating Direction from Xb Positive Direction to Yb Positive Direction" → "0"
・When "Rb Positive Direction" is opposite "Rotating Direction from Xb Positive Direction to Yb Positive Direction" → "1"

(Example) In the construction below, Rb positive direction (clockwise) is opposite the rotating direction from Xb positive direction to Yb positive direction R' (counterclockwise), thus All Axes Parameter No. 56 should be set to "1".




Example for unit (Left : Each axis system / Right : Base coordinate system)

(Example) Example for Standard Setting in TTA

| All Axes Parameter No.55 Setting Value |
|---|
| 1h |

| TTA Type | All Axes Parameter No.56 Setting Value |
|---|---|
| TTA-A2 / TTA-C2 (2-axis Type) | 21h |
| TTA-A3 / TTA-C3 (3-axis Type) | 321h |
| TTA-A4 / TTA-C4 (4-axis Type) | 4321h |

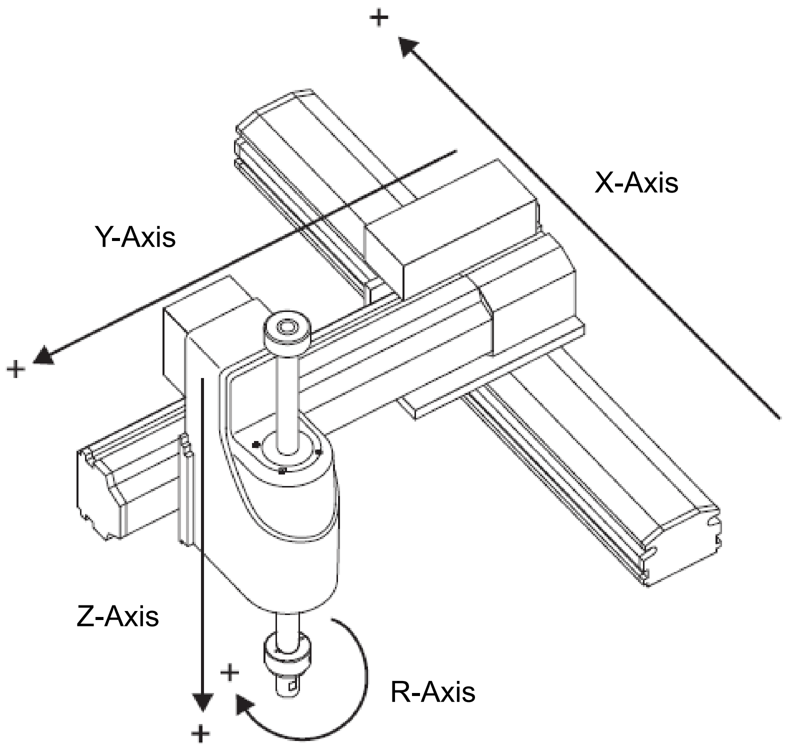
| X-Axis Home Type | Y-Axis Home Type | All Axes Parameter No.57 Setting Value (for four-axis type) |
|---|---|---|
| Standard | Standard | 0 |
| Reversed Specification (NM) | Standard | 1 |
| Standard | Reversed Specification (NM) | 1 |
| Reversed Specification (NM) | Reversed Specification (NM) | 0 |

* Shown above are just examples for standard settings. There would be no problem if change is made in the setting for All Axes Parameter No. 56 and 57 as long as there is no contradiction to the composition of the axes.
In such a case, make sure to check the caution items for each parameter so the settings can be established with no failure.

(Example) In TTA-C4 (XY Home Standard Type)
- Indicate Axis1 as Y-axis and Axis2 as X-axis
  → All Axes Parameter No. 56 = "4312h", All Axes Parameter No. 57 = "1"
- Construct the coordinate system only with Axis1 and Axis2
  → All Axes Parameter No. 56 = "21h"

[3] Coordinates for Coordinate System Definition Unit
The coordinate system for the coordinate system definition unit consists of four coordinate axes at the maximum (X-axis, Y-axis, Z-axis and R-axis).
At this time, it should be premised that the physical axes applicable for each coordinate axis satisfy the following conditions.

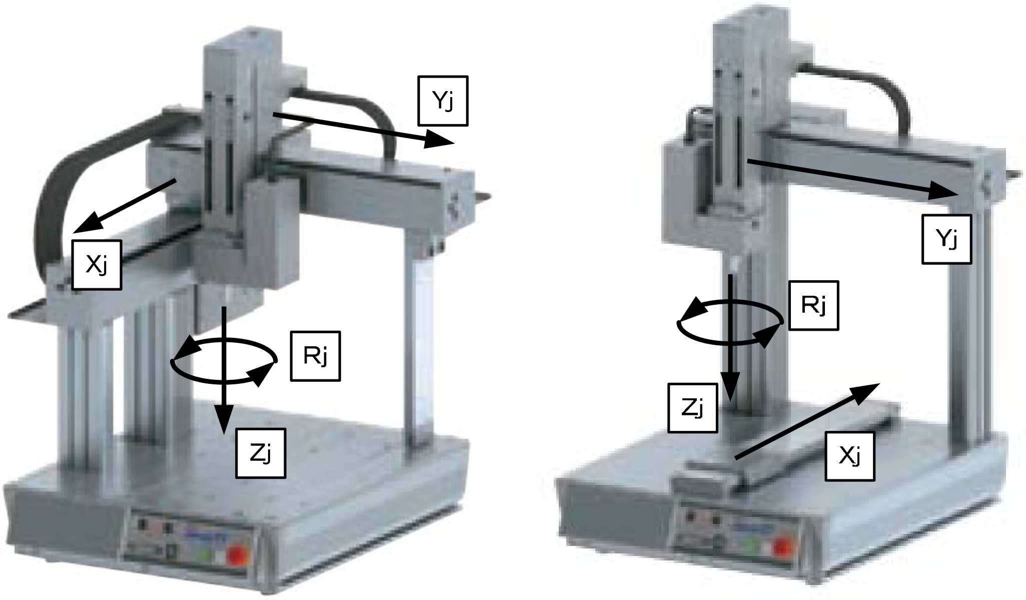
| |
|---|
| · X-axis, Y-axis and Z-axis are laid orthogonally to each other.<br>· The center axis of rotary for the R-axis is laid orthogonally to XY plane. (It is parallel to Z-axis if there is Z-axis.)<br>· A tool is attached on the R-axis if there is R-axis. (It should be premised that the tool attachment datum point is on the R rotation axis.) |




Example for orthogonal unit satisfying conditions above

[Each Axis System]

Each axis system is the coordinate system specific for each linear axis.
In below, shows each axis system in four-axis type TTA for example.

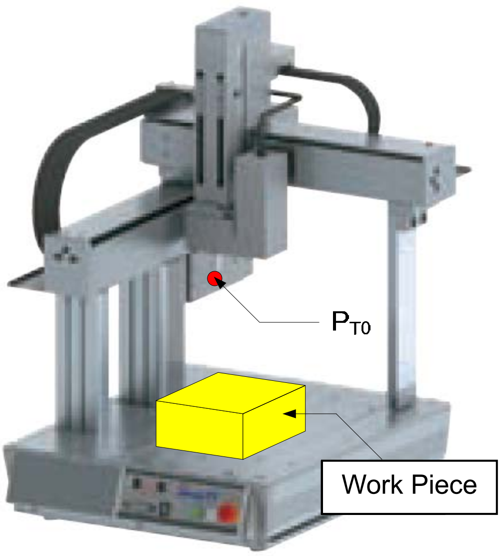



Example for Each Axis System in Four-Axis Type TTA (XYZ Home Standard Specification)
(Left:TTA-C4 / Right:TTA-A4)

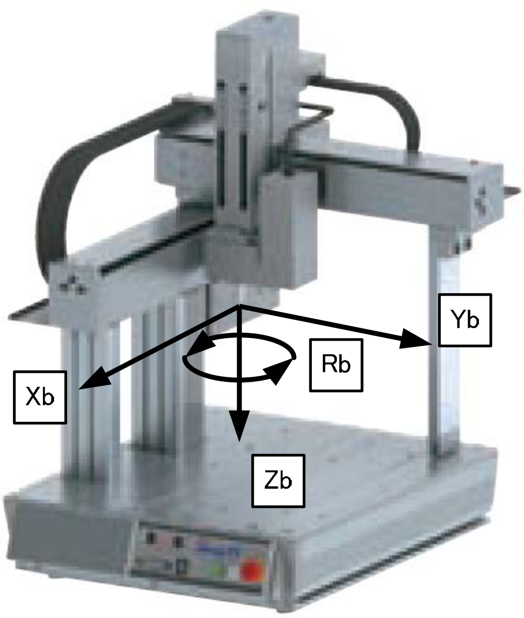
[Base Coordinate System]

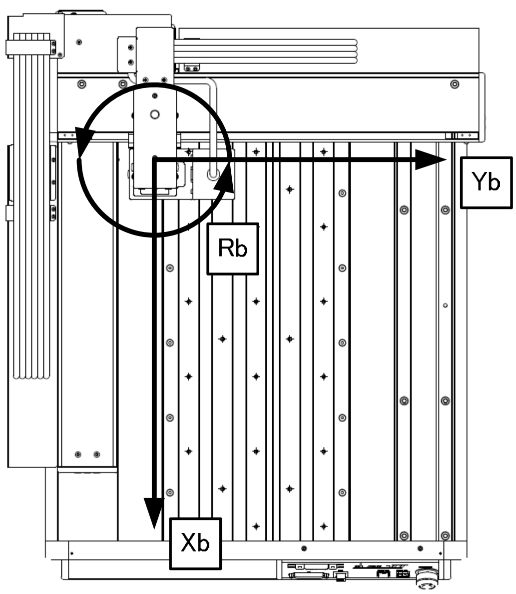
It is the coordinate system to indicate the position of the datum point for tool installation against the work piece mount face. Work Coordinate System No. 0 (work coordinate system offset 0) = Base Coordinate System. X axis of Base Coordinate System is described as Xb, Y axis as Yb, Z axis as Zb and R axis as Rb.

(Example) TTA-C4 (XYZ Home Standard Specification)

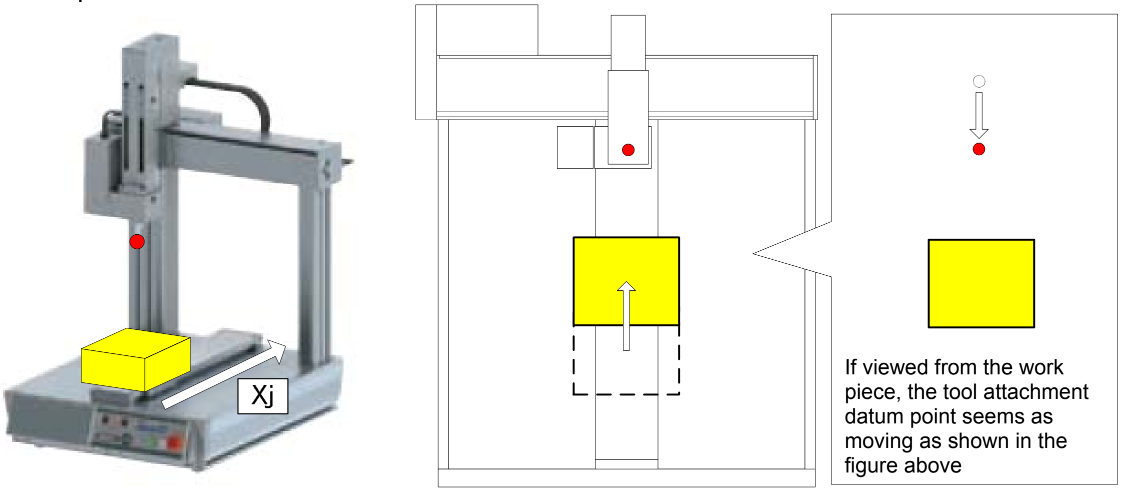



The base coordinate system can be defined as shown below.

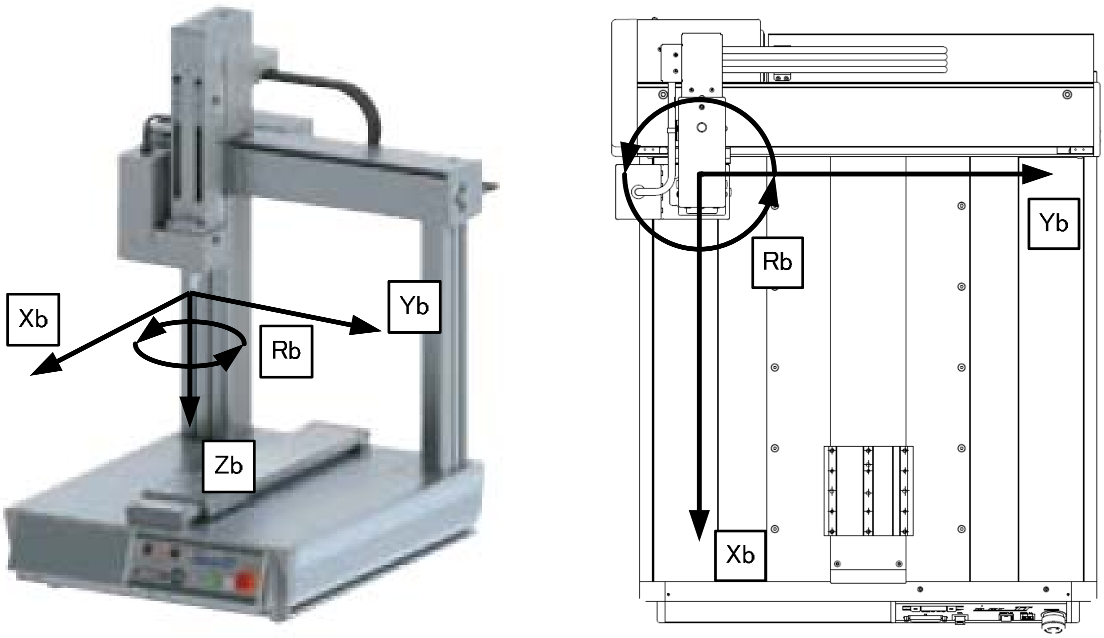


Also, when a work piece is mounted on an axis of either X, Y or Z-axis, the positive side of the base coordinate system is the opposite direction of the operation direction of the physical axis. For instance, in below explains for when a work piece is mounted on the X-axis in TTA-A4 (XYZ Home Standard Specification).
In this case, if moving the X-axis in the positive direction of each axis system, the tool attachment datum point seems to move in opposite direction to the motion if viewed from the work piece.




Therefore, the base coordinate system is shown as the figure below.




In below, shows the definition related to the home of the base coordinate system and the direction of coordinates.

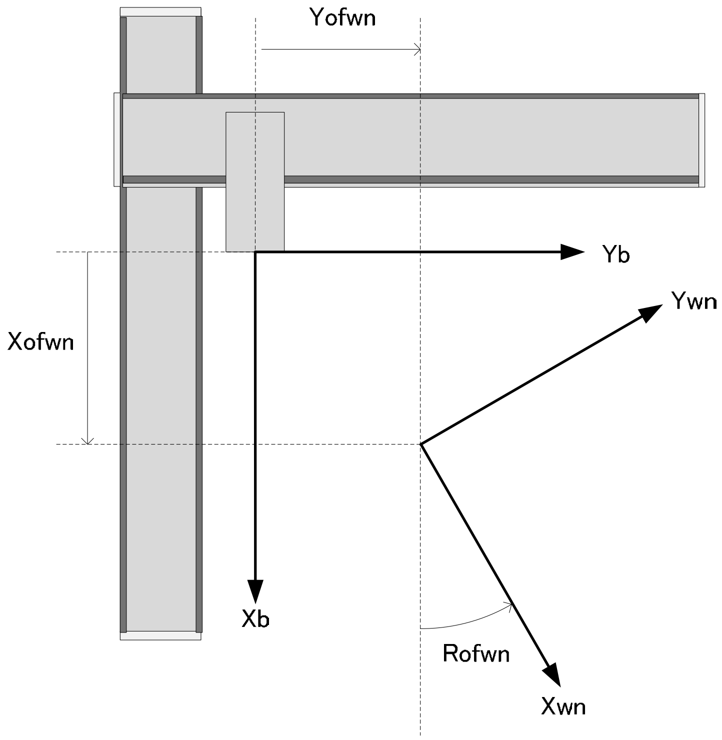
| | | |
|---|---|---|
| Home | : | The position of the tool attachment datum point under condition that each axis system coordinates for all the unit constructing axes = 0 |
| Positive direction of coordinate | : | The direction that the tool attachment datum point moves to the work piece mount face when each axis in the each axis system is moved in the positive direction. |

## [Work Coordinate System]

It is the 32 kinds of coordinate systems defined by the offset of each axis against the base coordinate system.
Work Coordinate System No. 0 is reserved as Base Coordinate System (= Work Coordinate System Offset = 0) by the system.

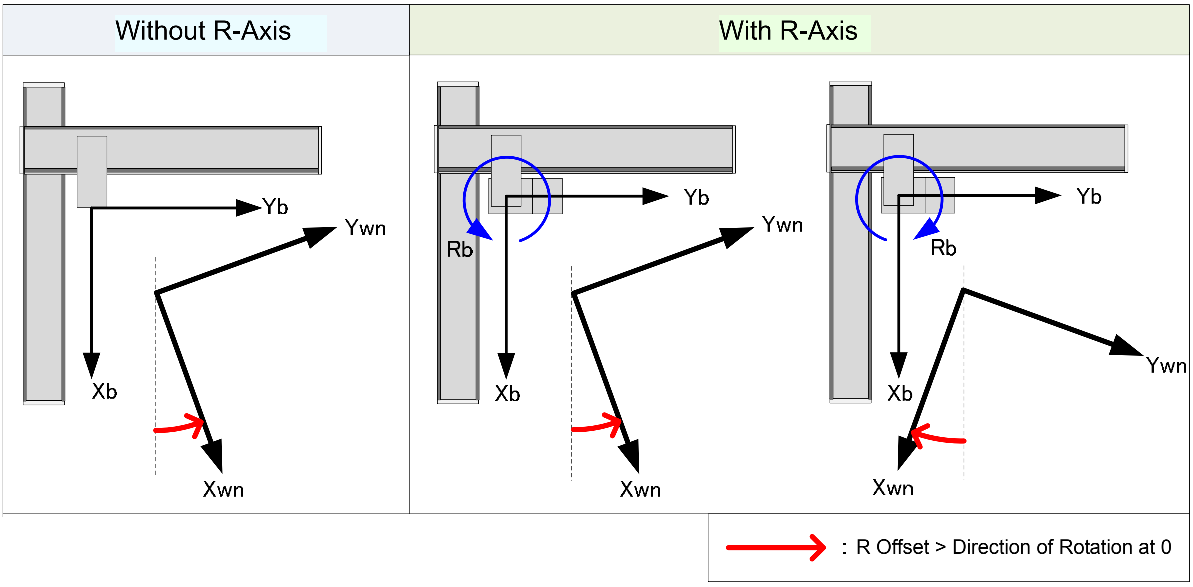
Set the offset of each axis as described below.

- X, Y, Z Offset
  Distance from the base coordinate system home to the work coordinate system home along Xb, Yb and Zb directions
- R Offset
  Amount of work coordinate system rotation when base coordinate system taken as the datum



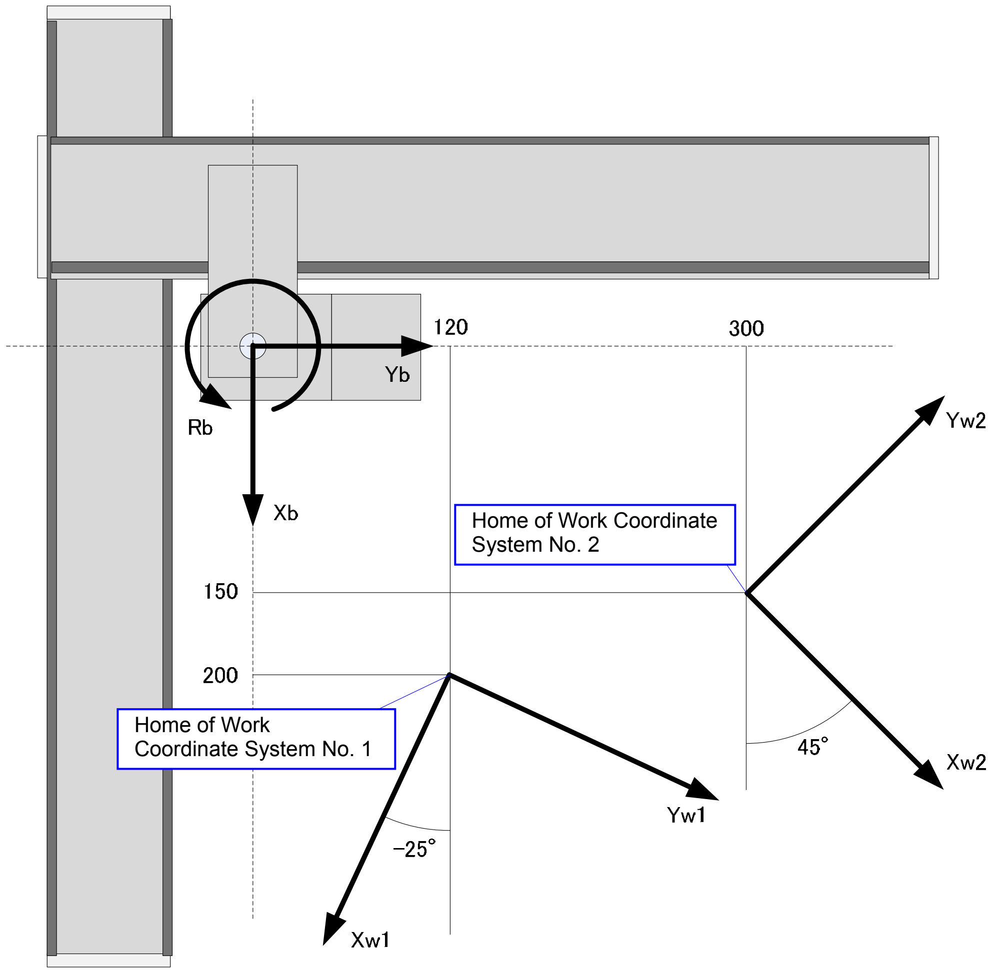
Also, the direction of the coordinate system rotation (positive direction of R offset) is defined as described below by the axis structure of the coordinate system definition unit.

| |
|---|
| · When no R-axis in constructing axes · · · · · · · Rotating Direction from Xb positive direction to Yb positive direction<br>· When there is R-axis in constructing axes · · · Rb Positive direction |



(1) Setting of Work Coordinate System

When required to define Work Coordinate System No. 1 and No. 2 as shown in the figure below;




The offset of Work Coordinate System No. 1 and No. 2 are to be set as shown in the table below.

| Work Coordinate System No. | Offset | | | |
|---|---|---|---|---|
| | X [mm] | Y [mm] | Z [mm] | R [deg] |
| 1 | 200.000 | 120.000 | 0.000 | -25.000 |
| 2 | 150.000 | 300.000 | 0.000 | 45.000 |

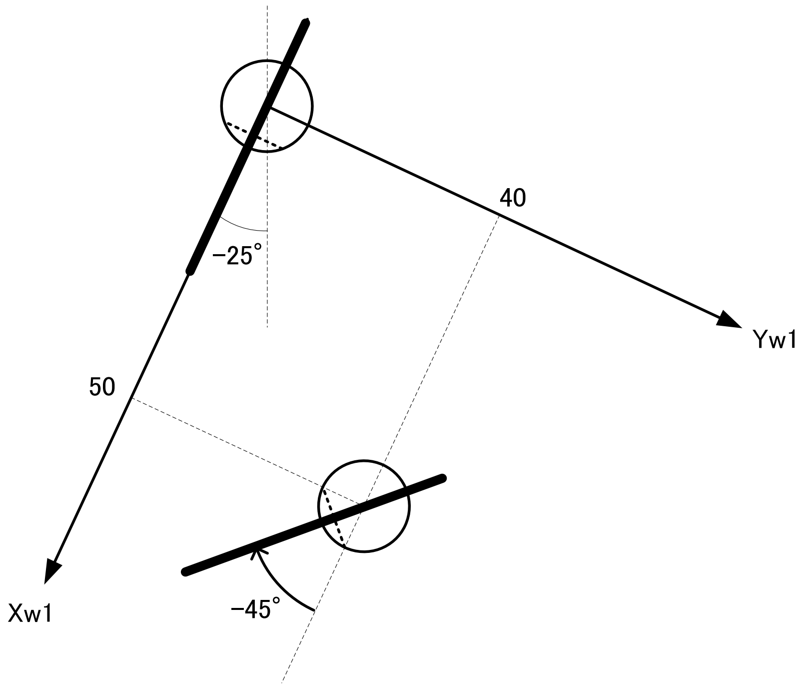
(2) Positioning on Work Coordinate System

Select the work coordinate system to be used and perform positioning.
When selecting the work coordinate system number in SEL program, use SLWK Command.
Also, the selected work coordinate system number is valid after program complete and after rebooting.

1) When having PTP positioning to Position No. 5 and No. 6 on Work Coordinate System No. 1.

| Work Coordinate System No. | Offset | | | |
|---|---|---|---|---|
| | X [mm] | Y [mm] | Z [mm] | R [deg] |
| 1 | 200.000 | 120.000 | 0.000 | -25.000 |

| Position No. | Coordinate Data | | | |
|---|---|---|---|---|
| | Axis1(X) | Axis2(Y) | Axis3(Z) | Axis4(R) |
| 5 | 0.000 | 0.000 | 0.000 | 0.000 |
| 6 | 50.000 | 40.000 | 0.000 | -45.000 |

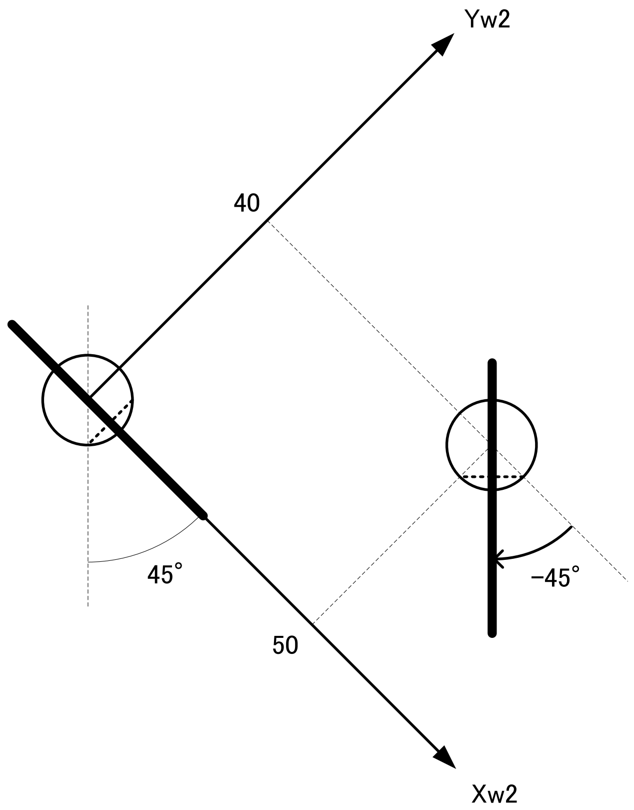



* Example for when the datum surface on the R-axis orients Xb positive direction when each axis system coordinate on all the unit axes = 0

2) When having PTP positioning to Position No. 5 and No. 6 on Work Coordinate System No. 2.

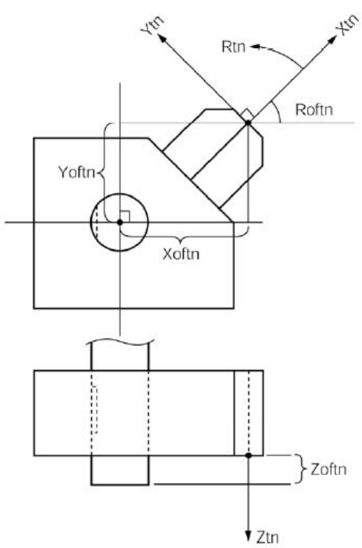
| Work Coordinate System No. | Offset | | | |
|---|---|---|---|---|
| | X [mm] | Y [mm] | Z [mm] | R [deg] |
| 2 | 150.000 | 300.000 | 0.000 | 45.000 |

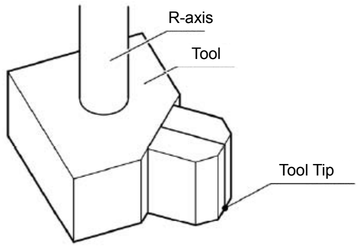
| Position No. | Coordinate Data | | | |
|---|---|---|---|---|
| | Axis1(X) | Axis2(Y) | Axis3(Z) | Axis4(R) |
| 5 | 0.000 | 0.000 | 0.000 | 0.000 |
| 6 | 50.000 | 40.000 | 0.000 | -45.000 |



## [Tool Coordinate System]

It is the 128 kinds of coordinate systems defined by the too (such as hand) dimensions (offset) of that attached on the tool attached position. Work Coordinate System No. 0 is reserved as offset = 0 of Tool Coordinates by the system.
Select the defined tool coordinate system number, and it is used as the destination point at positioning of the tool tip as well as the tool attached position.



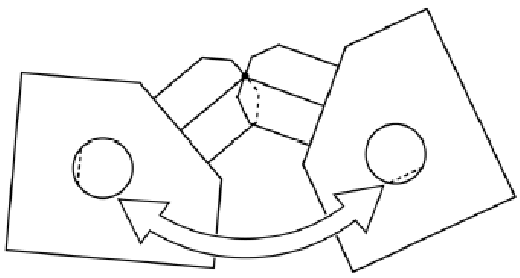




Xoftn : X Tool Coordinate offset
Yoftn : X Tool Coordinate offset
Zoftn : X Tool Coordinate offset
Roftn : X Tool Coordinate offset

Xtn : Tool Coordinate X-axis
Ytn : Tool Coordinate X-axis
Ztn : Tool Coordinate X-axis
Rtn : Tool Coordinate X-axis

(n is coordinate system number)

Select the defined Tool Coordinate System and operate the R axis with JOG operation, and such movement as shown in the figure below can be performed.

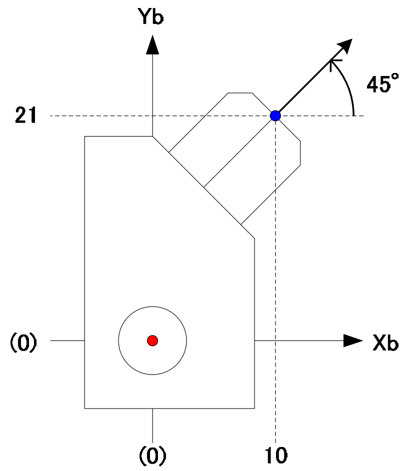


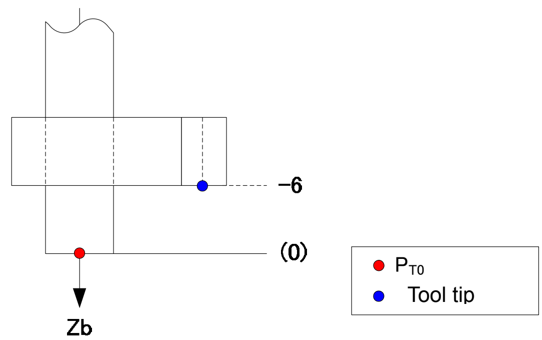
● Regarding directions of X and Y axes (Xtn and Ytn) in tool coordinate system
With R-Axis : Varies relying on coordinates of R-axis
Without R-Axis : Always fixed, the directions depends on R offset of tool coordinate system

(1) Setting the tool coordinate system
Set the offset amount from the tool attached position to the tool tip.
Set the tool offset as explained below under condition that each axis system coordinates for all the unit constructing axes is 0 is taken as the datum.

· X, Y, Z offset
  Distance from the tool attached position to the tool tip along Xb, Yb and Zb directions of the base coordinate system
· R offset
  Angle of the working direction with Xb positive direction as the datum
  (Definition for direction of angle is the same as work coordinate system R offset)

· Example for Tool Coordinate System Setting
When required to set Tool Coordinate System No. 1 as shown in the figure below;
(The figure below is assumed to show that each axis system coordinates for all the unit constructing axes is 0)




The offset of Tool Coordinate System No. 1 are to be set as shown in the table below.

| Tool Coordinate System No. | Offset | | | |
|---|---|---|---|---|
| | X [mm] | Y [mm] | Z [mm] | R [deg] |
| 1 | 10.000 | 21.000 | -6.000 | 45.000 |

(2) Positioning using Tool Coordinate System Offset

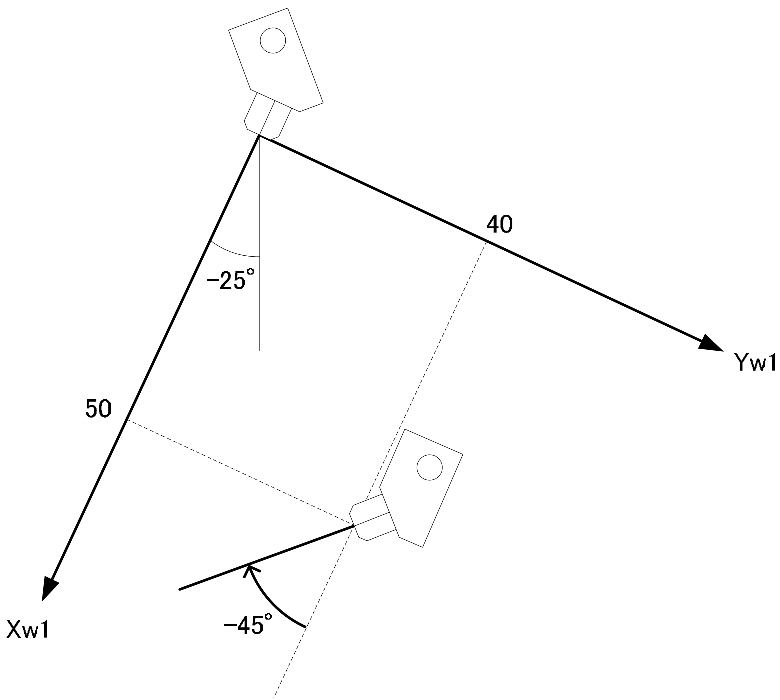
Select the work coordinate system to be used and perform positioning.
When selecting the work coordinate system number in SEL program, use SLTL Command.
Also, the selected work coordinate system number is valid after program complete and after rebooting.

1) When having the tool tip on Tool Coordinate System No. 1 to perform PTP positioning from Position No. 5 to No. 6 in Work Coordinate System No. 1;

| Work Coordinate System No. | Offset | | | |
|---|---|---|---|---|
| | X [mm] | Y [mm] | Z [mm] | R [deg] |
| 1 | 200.000 | 120.000 | 0.000 | -25.000 |

| Tool Coordinate System No. | Offset | | | |
|---|---|---|---|---|
| | X [mm] | Y [mm] | Z [mm] | R [deg] |
| 1 | 10.000 | 21.000 | -6.000 | 45.000 |

| Position No. | Coordinate Data | | | |
|---|---|---|---|---|
| | Axis1(X) | Axis2(Y) | Axis3(Z) | Axis4(R) |
| 5 | 0.000 | 0.000 | 0.000 | 0.000 |
| 6 | 50.000 | 40.000 | 0.000 | -45.000 |

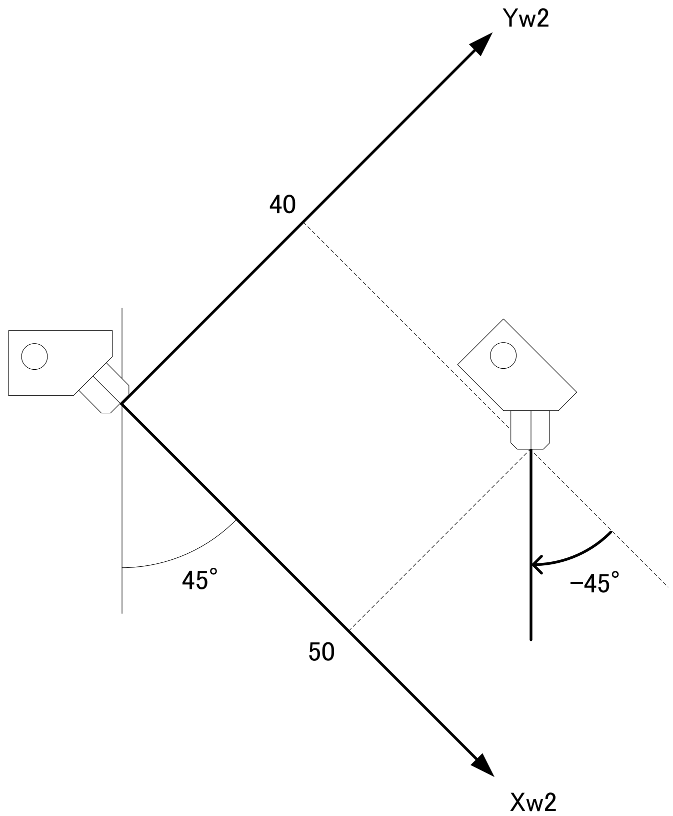


2) When having the tool tip on Tool Coordinate System No. 1 to perform PTP positioning from Position No. 5 to No. 6 in Work Coordinate System No. 2;

| Work Coordinate System No. | Offset | | | |
|---|---|---|---|---|
| | X [mm] | Y [mm] | Z [mm] | R [deg] |
| 2 | 150.000 | 300.000 | 0.000 | 45.000 |

| Tool Coordinate System No. | Offset | | | |
|---|---|---|---|---|
| | X [mm] | Y [mm] | Z [mm] | R [deg] |
| 1 | 10.000 | 21.000 | -6.000 | 45.000 |

| Position No. | Coordinate Data | | | |
|---|---|---|---|---|
| | Axis1(X) | Axis2(Y) | Axis3(Z) | Axis4(R) |
| 5 | 0.000 | 0.000 | 0.000 | 0.000 |
| 6 | 50.000 | 40.000 | 0.000 | -45.000 |



[4] Caution Note

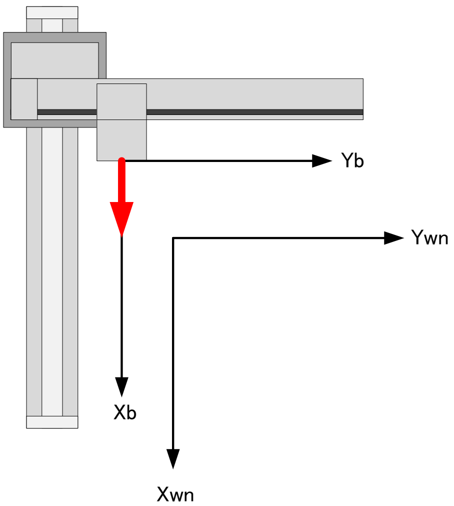
(1) Limitation in Coordinate System Constructing Axes
Shown below are the cases when limitation is applied to indication for operation of X, Y and R-axes in the coordinate system definition unit.

| Operations Subject for Limitation | | Indicated Axis | | | |
|---|---|---|---|---|---|
| | | X | Y | Z | R |
| Pressing Movement | (PUSH) | Δ*1 | Δ*1 | ○ | Δ*1 |
| Arch motion Z-axis | (ACHZ/PCHZ) | Δ*1 | Δ*1 | ○ | × |
| palletize XY-axis | (PASE/PAPS) | ○ | ○ | ○ | × |
| Circular / Arc Movement<br>3-Dimensional Circular / Arc Movement | (CIRS/ARCS)<br>(CIR/ARC) | ○ | ○ | ○ | × |
| Circular / Arc Movement 2<br>Arc Movement with Center Indication | (CIR2/ARC2)<br>(ARCC/ARCD) | ○ | ○ | Δ*2 | × |

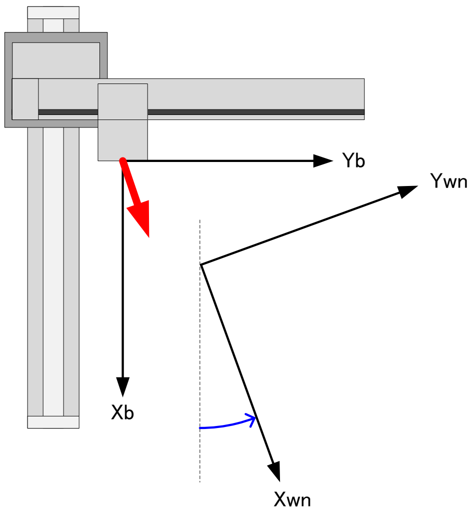
○ : Available for Indication Δ : Available for Indication with Conditions × : Unavailable for Indication

(*1) Permitted only when operated physical axis is one axis to the indicated axis
(*2) Permitted only when operated axes are two axes in total

Example : To indicate X-axis in PUSH Command
1) Work coordinate system R offset is 0 (in such case as Work Coordinate System No. 0)
→ Available to indicate (Only X-axis operates)
2) Work coordinate system R offset is not 0
→ Unavailable to indicate (Operation necessary on X and Y-axes)




○ : Available for Indication




× : Unavailable for Indication

(2) Note for Other Caution

- While in operation of either X-axis or Y-axis (or R-axis) in a SEL program, or with operation of PC software or TP, X-axis and Y-axis (and R-axis) cannot be operated in another SEL program.

- When operating either X-axis or Y-axis (or R-axis), it is basically necessary that servo is turned on and home return operation is completed on all of X-axis and Y-axis (and R-axis). It could cause an error if operation is attempted with those conditions above unsatisfied.

- In the position edit window for the PC software and teaching tools applicable for the feature, when operation is made to turn on/off the servo to the coordinate system definition unit axes, servo turns on/off on all the axes in the applicable units.

- When performing CP operation on the R-axis by setting the tool coordinate system offset, the operation speed and acceleration / deceleration on the X-axis and Y-axis will be limited so they do not exceed the indicated velocity and acceleration / deceleration. Therefore, even if the indicated velocity and acceleration / deceleration are the same, the actual operation speed may tend to be slower as the tool length (distance from the tool attachment datum point to the tool tip) gets longer.

- When attempted to perform CP operation or JOG operation on the work coordinate system or tool coordinate system while the coordinate system definition unit axis is out of the soft limit, “Error No. C73 Target Track Soft Limit Exceeding Error” could occur. In such a case, use the PC software or a teaching pendant and have JOG operation on each axis system to have the axis moved inside the soft limit from outside.

- When a velocity change command is issued to CP operation including the R-axis, the specification could be limited depending on the condition. Refer in the section for CHVL Command for details. [Refer to the explanation of section of CHVL command for details.]

- When a synchronized master axis is indicated as the coordinate system definition unit axis, the applicable synchronized slave axes are to be treated as the axes outside the unit (* Operation will be synchronized to the synchronized master axis as it does in ordinary use). Therefore, the work coordinate system and the tool coordinate system are not considered in the current position display of the slave axes, and will be in the each axis system value.

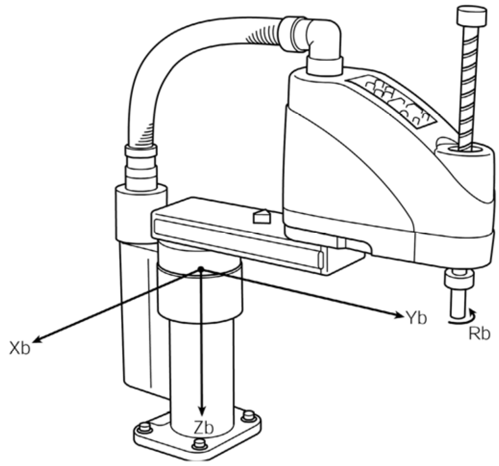
### 1.4.6 SCARA Robot

[1] Coordinates

A horizontal articulated (SCARA) robot has three coordinate systems including the base coordinate system, work coordinate system and tool coordinate system.

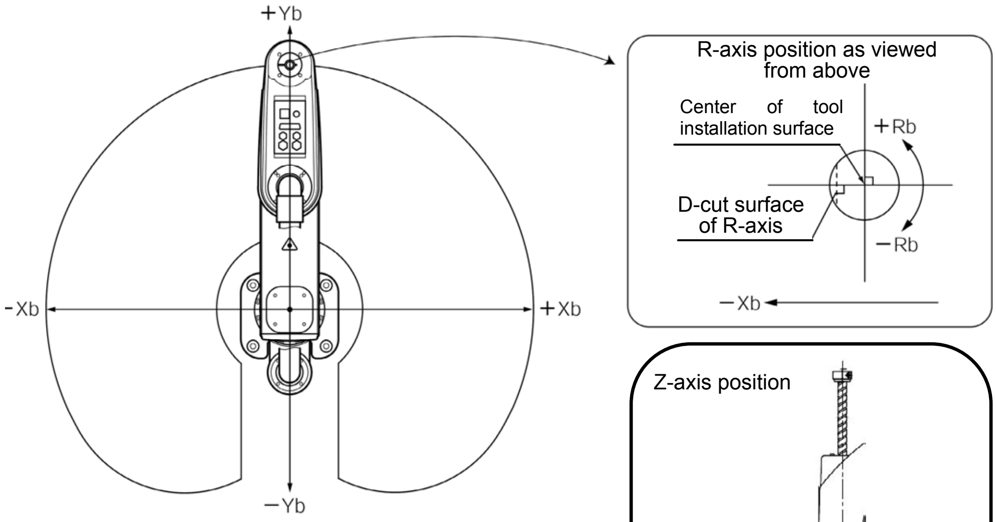
Coordinates
- Base coordinate system
- Work coordinate system
- Tool coordinate system

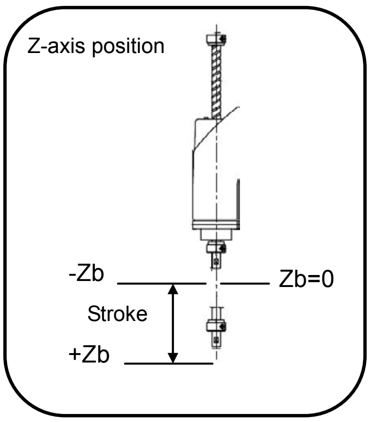
(1) Base coordinate system (= Work Coordinate System No. 0)
This is a combination of three-dimensional rectangular coordinates and rotational axis coordinates defined in the robot prior to shipment.
Work Coordinate System No. 0 (= 0 work coordinate system offsets) = Base coordinate system.



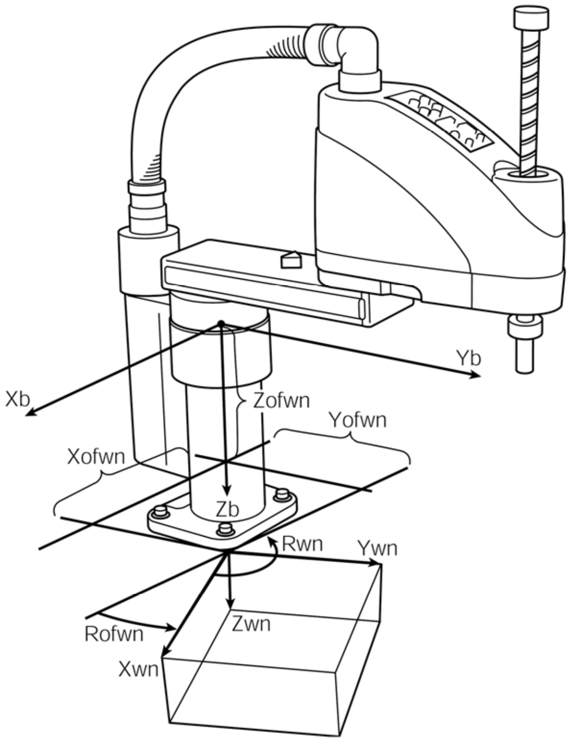

* There is no rotary axis on Three-Axis Type SCARA ROBOT (IXP-3N****). (Xb, Yb and Zb) are available to indicate as the target position.

The XY-axis home is the center of the base (center of rotation of arm 1).
The Z-axis home is the top edge of the effective Z-axis stroke.
The R-axis home is where the D-cut surface faces the -Xb direction.
The X-axis, Y-axis, Z-axis and R-axis on the base coordinate system are indicated as Xb, Yb, Zb and Rb, respectively.



(2) Work coordinate system
This is a combination of 32 types of three-dimensional rectangular coordinates and rotational axis coordinates defined by the offset of each axis relative to the base coordinate system. Note that Work Coordinate System No. 0 is reserved as the base coordinate (= 0 work coordinate system offset) by the system.




Xofwn: X work coordinate offset
Yofwn: Y work coordinate offset
Zofwn: Z work coordinate offset
Rofwn: R work coordinate offset

Xwn: Work coordinate system, X-axis
Ywn: Work coordinate system, Y-axis
Zwn: Work coordinate system, Z-axis
Rwn: Work coordinate system, R-axis

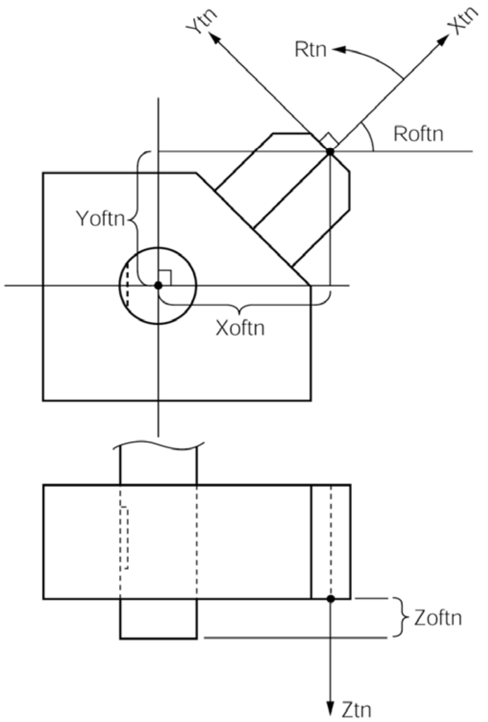
(n indicates the work coordinate system number.)

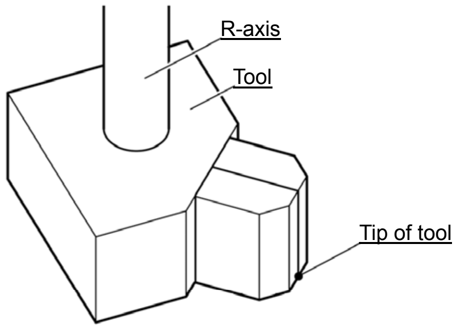
* R work coordinate offset is valid also on Three-Axis Type SCARA ROBOT (IXP-3N****).

(3) Tool coordinate system

This is a combination of 128 types of three-dimensional rectangular coordinates and rotational axis coordinates defined by the dimension (offset) of the tool (hand, etc.) installed on the tool installation surface. Note that Tool Coordinate System No. 0 is reserved as one with 0 tool coordinate system offset by the system.

When a defined tool coordinate system number is selected, the robot uses the tip of the tool, not the center of the tool installation surface, as the point to reach by positioning.



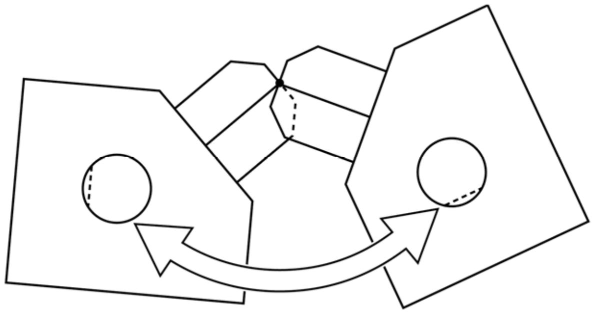




Xoftn: X tool coordinate offset
Yoftn: Y tool coordinate offset
Zoftn: Z tool coordinate offset
Roftn: R tool coordinate offset

Xtn: Tool coordinate system, X-axis
Ytn: Tool coordinate system, Y-axis
Ztn: Tool coordinate system, Z-axis
Rtn: Tool coordinate system, R-axis

(n indicates the tool coordinate system number.)

When a defined tool coordinate system is elected and the R-axis is jogged, the axis operates as shown below.



<u>The orientation of the tool coordinate system axis is always the same as the orientation of the base coordinate system axis</u> on Three-Axis Type SCARA ROBOT (IXP-3N****). As there is no R axis, the control of the orientation (posture) of the tool cannot be conducted. Therefore, there may be a case the tool end may not be the reachable point at positioning. Also, the setting of the tool coordinate system offset on the R axis will be ignored.

[2] CP Operation and PTP Operation

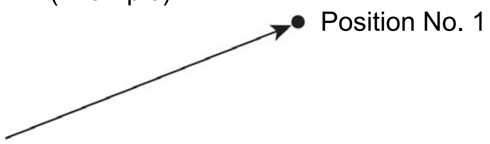
How CP operation and PTP operation differ as they pertain to SCARA robots is explained.

(1) CP operation

1) Path

The axes move to the target position while interpolating with one another. The path of the tip of movement can be specified by a command (linear, circle, arc, path movement, etc.).

(Example)




MOVL 1

Move from the current position to position No. 1 along a straight line.

The arm system does not change during CP operation.

CP operation commands: MOVL MVLI TMLI PATH PSPL PUSH CIR2 ARC2 ARCD ARCC CIRS ARCS CIR ARC

For details on these commands, refer to “Explanation of Each Command”

2) Setting of speed and acceleration/deceleration for CP operation

In CP operation, the speed and acceleration/deceleration are set beforehand in the program using a control declaration command.

Speed setting command “VEL” unit [mm/sec]

Acceleration setting command “ACC” unit [G]

Deceleration setting command “DCL” unit [G]

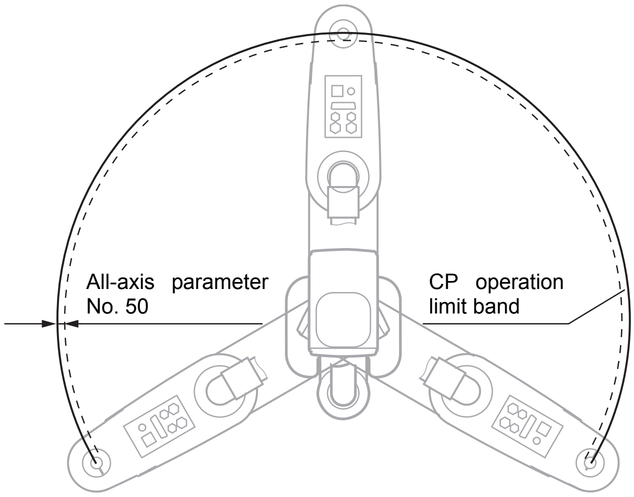
(Example)

| | | |
|---|---|---|
| ACC | 0.5 | Set the acceleration for CP operation to 0.5G. |
| DCL | 0.5 | Set the deceleration for CP operation to 0.5G. |
| VEL | 500 | Set the speed for CP operation to 500mm/sec. |
| | | |
| MOVL | 2 | Move to position No. 2 along a straight line. |

The speed and acceleration/deceleration for CP operation can also be set in the VEL, ACC and DCL fields of the position data table.

To set these items as part of position data, do so for each position number. If the VEL, ACC and DCL fields of the position data table contain settings for a given position number, they are given priority over the “VEL”, “ACC” and “DCL” commands in the program when moving to the applicable position number.

3) Notes on CP operation
The singular point is where both arms 1 and 2 extend straight.
If the actuator moves near the singular point via CP operation, poor path precision, vibration (abnormal noise) or error may occur. The following errors may generate: “D09: Driver overspeed error”, “B91: Main overspeed error”, “C64: Abnormal servo acceleration/deceleration error”, “B74: CP operation limit band entry error”, “CB6: Deviation overflow error”, etc.
These errors may be avoided by lowering the speed and/or acceleration/deceleration.




With the area inside for the amount of either All Axes Parameter No. 50 or No. 227 from the singularity defined as the CP operation limit band, the CP operation is limited within this area. (In the figure on the left, the area between the solid line and dotted line is the CP operation limit band.)

The controller generates an error upon detecting an entry of the target path or actual movement path into the CP operation limit band as a result of path calculation. If the target movement path enters the CP operation band as a result of path calculation, “B7C: Target path in CP operation limit band error (PTP/jotting of axis permitted)” occurs.

If the actual movement path enters the CP operation limit band, on the other hand, “B74: CP operation limit band entry error (PTP/jogging of axis permitted)” or “C74: Actual position soft limit over error” occurs.
The width of the CP operation limit band (distance between the solid line and dotted line) varies depending on the arm length of the robot.
(If the arm length is 500 or 600, this width becomes approx. 0.5mm (All Axes Parameter No. 50 or No. 227, “Width of CP operation limit band around point directly above Arm 1/2”).)

Avoid writing programs that cause the actuator to pass the CP operation limit band via CP operation.
The actuator cannot pull out from the CP operation limit band by means of CP operation. In this case, move the actuator in PTP operation. Exercise caution in situations where the condition of each arm is not recognized, such as when the program is started, etc.
With CP operation, conduct test operation at low speed at first and confirm absence of problems, and then gradually raise the speed to an appropriate level.

(2) PTP operation

1) Movement path

Each axis moves to the target position at the specified speed. The path of the tip of movement cannot be specified by a command.

(Example)

Position No. 1

MOVP 1

Move from the current position to position No. 1 via PTP operation.

The arm system may change during movement depending on the operation area and arm system control command.

PTP operation commands: MOVP MVPI TMPI PACH PMVP ARCH

For details on these commands, refer to “Explanation of Each Command”

2) Setting of speed and acceleration/deceleration for PTP operation

In PTP operation, the speed and acceleration/deceleration are set beforehand in the program using a control declaration command.

Speed setting command “VELS”, unit [% (percentage relative to the maximum PTP speed (SCARA axis) set in axis-parameter No. 28)]

Acceleration setting command “ACCS”, unit [% (percentage relative to the maximum PTP acceleration for SCARA axis set in axis-parameter No. 134)]

Deceleration setting command “DCLS”, unit [% (percentage relative to the maximum PTP deceleration for SCARA axis set in axis-parameter No. 135)]

(Example)

| | | |
|---|---|---|
| ACCS | 50 | Set the acceleration for PTP operation to 50% of the maximum PTP acceleration. |
| DCLS | 50 | Set the deceleration for PTP operation to 50% of the maximum PTP deceleration. |
| VELS | 50 | Set the speed for PTP operation to 50% of the maximum PTP speed. |
| MOVP | 2 | Move to position 2 via PTP operation. |

3) Note on PTP operation

The arm system may change during an operation depending on the operation area, arm system control command or position data arm system.

Refer to “[4] Arm System”.

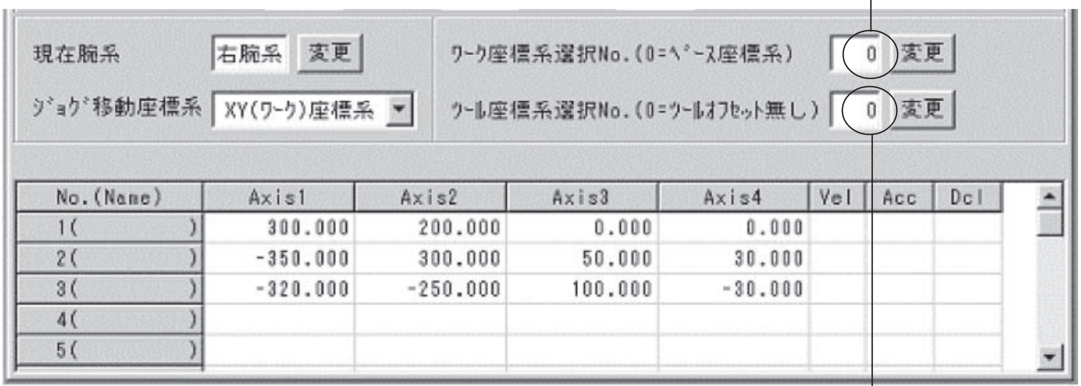
[3] Settings of Each Axis

(1) Base coordinate system

1) Positioning on Base Coordinate System

To select a work coordinate system number in the SEL program, use a SLWK command.
The work coordinate system selection number that has been set will remain effective even after the program ends or after the system-memory backup battery has been set and power has been reconnected. (For XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD, MSEL-PCX/PGX battery is not necessary.)
The figure below shows a part of the position edit screen in the PC software for horizontal articulated robot.
In this example, the following teaching settings are assumed:
Position data for Position No. 1: X = 300, Y = 200, Z = 0, R = 0
Position data for Position No. 2: X = -350, Y = 300, Z = 50, R = 30
Position data for Position No. 3: X = -320, Y = -250, Z = 100, R = -30

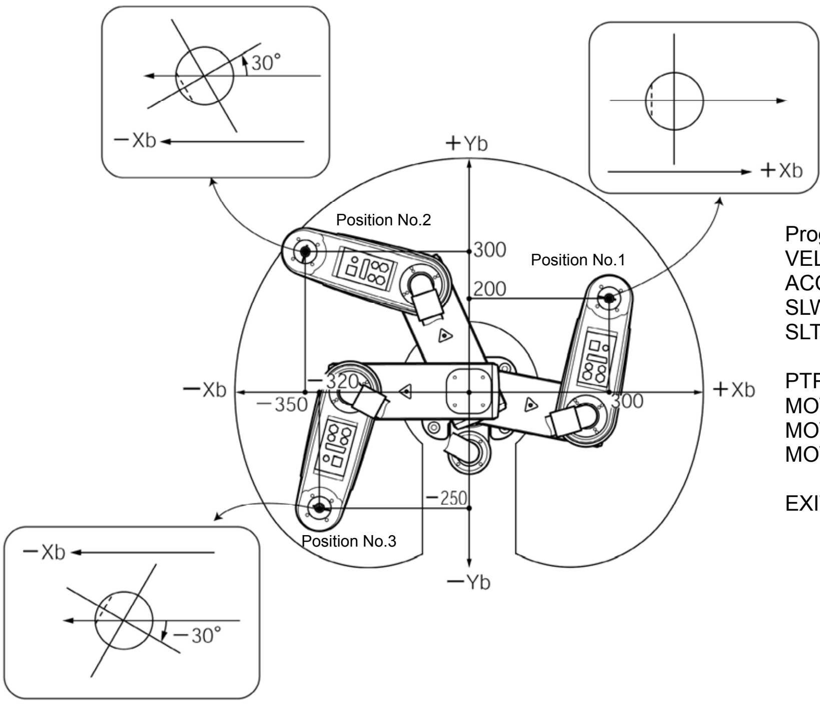
The selected work coordinate system number is displayed.
Work Coordinate System No. 0 = Base coordinate system




| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 1( ) | 300.000 | 200.000 | 0.000 | 0.000 | | | |
| 2( ) | -350.000 | 300.000 | 50.000 | 30.000 | | | |
| 3( ) | -320.000 | -250.000 | 100.000 | -30.000 | | | |
| 4( ) | | | | | | | |
| 5( ) | | | | | | | |

The selected tool coordinate system number is displayed. In the case of Tool Coordinate System No. 0, the center of the tool installation surface is positioned.

2) Position to the position data shown above via PTP operation.




Program example

```
VELS    50
ACCS    50
SLWK    0     Select Work Coordinate System No. 0
SLTL    0     Select Tool Coordinate System No. 0

PTPR          Specify right arm of PTP target arm system
MOVP    1
MOVP    2
MOVP    3

EXIT
```

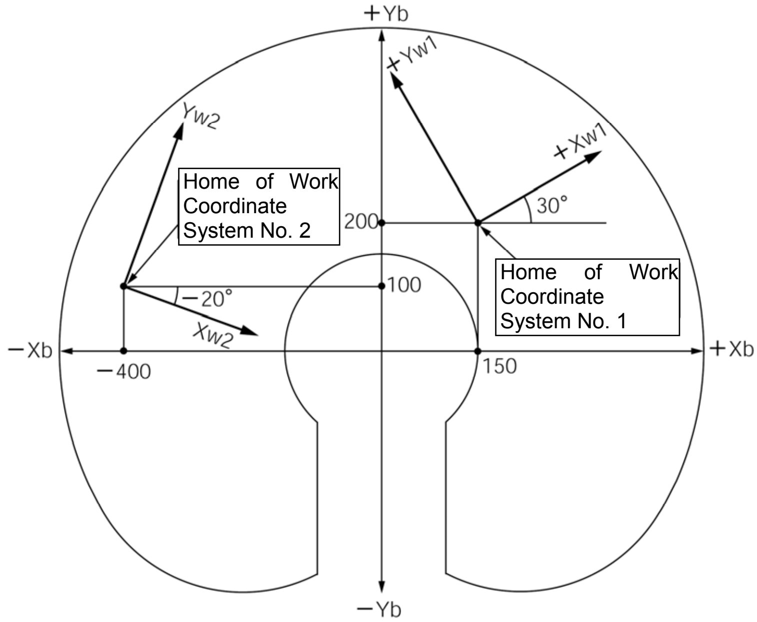
(2) Work coordinate system

1) Setting of Work Coordinate System

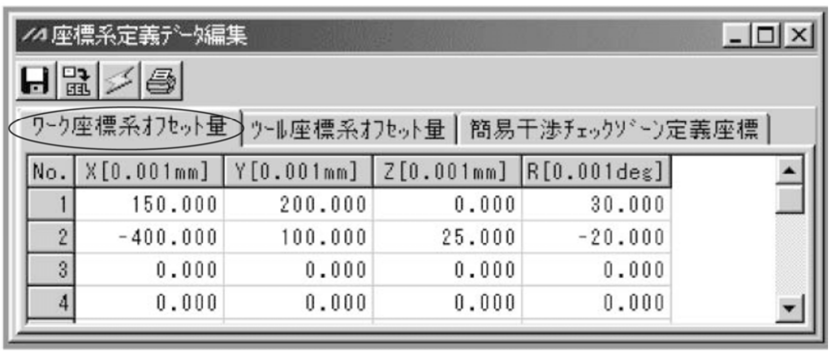
Set the offset relative to the base coordinate system.

(Example) Setting example of work coordinate system

Define Work Coordinate System No. 1 and No. 2 as shown below.




For the offset of Work Coordinate System No. 1, set Xofw1 = 150, Yofw1 = 200, Zofw1 = 0 and Rofw1 = 30.
For the offset of Work Coordinate System No. 2, set Xofw2 = -400, Yofw2 = 100, Zofw2 = 25 and Rofw2 = -20.
Shown below is the edit screen for work coordinate system definition data in the PC software for horizontal articulated robot, where Work Coordinate System No. 1 and No. 2 have been set.

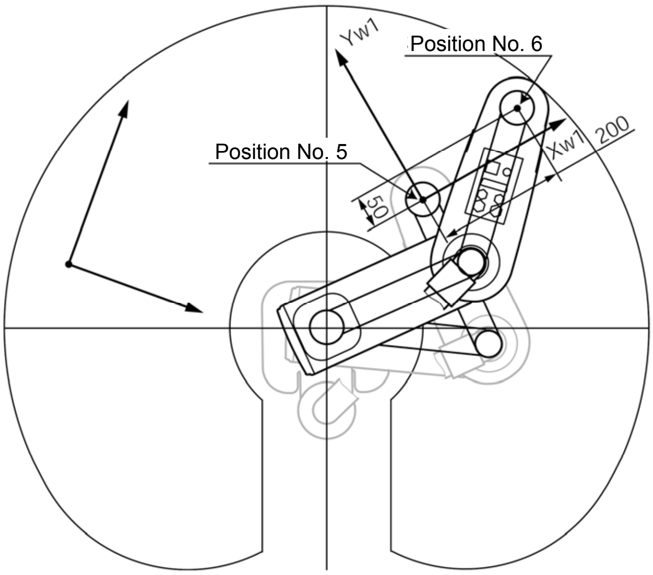



| No. | X[0.001mm] | Y[0.001mm] | Z[0.001mm] | R[0.001deg] |
|---|---|---|---|---|
| 1 | 150.000 | 200.000 | 0.000 | 30.000 |
| 2 | -400.000 | 100.000 | 25.000 | -20.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.000 | 0.000 |

* To set a work coordinate system offset in the SEL program, use a DFWK command.

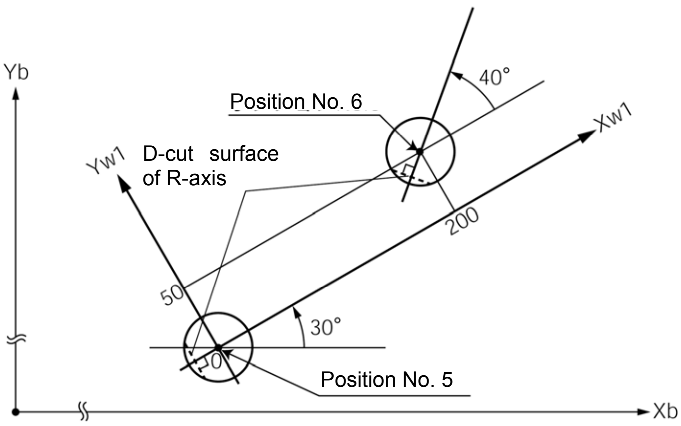
2) Positioning on Work Coordinate System
Perform positioning after selecting the work coordinate system you want to use.
To select a work coordinate system number in the SEL program, use a SLWK command.
The work coordinate system selection number that has been selected will remain effective even after the program ends or after the system-memory backup battery has been set and power has been reconnected. (For XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD, MSEL-PCX/PGX battery is not necessary.)
(Example 1) Position to Position No. 5 and No. 6 via PTP operation on Work Coordinate System No. 1.


| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 4( ) | | | | | | | |
| 5( ) | 0.000 | 0.000 | 0.000 | 0.000 | | | |
| 6( ) | 200.000 | 50.000 | 20.000 | 40.000 | | | |
| 7( ) | | | | | | | |




Program example

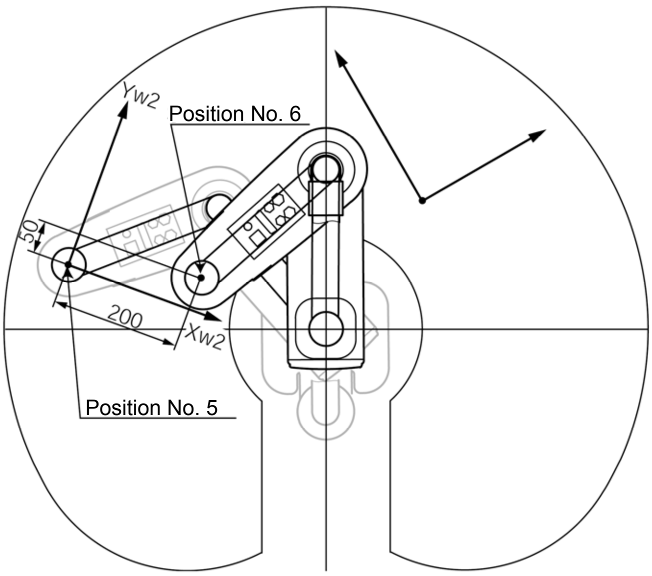
```
  :
  :
  :
SLWK   1   Select Work Coordinate System No. 1
SLTL   0   Select Tool Coordinate System No. 0
PTPR       Specify right arm of PTP target arm system
MOVP   5   Move to Position No. 5.
MOVP   6   Move to Position No. 6.
  :
  :
  :
```




The R-axis position is shown to the left (viewed from above).
The Z-axis position is as follows:
Position No. 5 Zb = 0
Position No. 6 Zb = 20

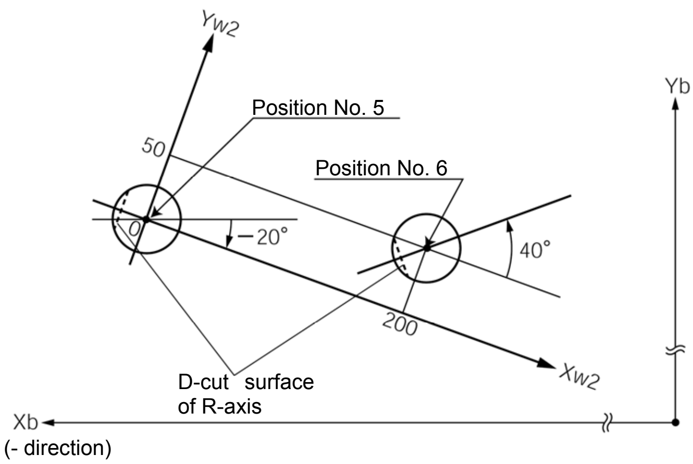
(Example 2) Position to Position No. 5 and No. 6 via PTP operation on Work Coordinate System No. 2.


| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 4( ) | | | | | | | |
| 5( ) | 0.000 | 0.000 | 0.000 | 0.000 | | | |
| 6( ) | 200.000 | 50.000 | 20.000 | 40.000 | | | |
| 7( ) | | | | | | | |




Program example

```
     :
     :
SLWK    2    Select Work Coordinate System No. 2
SLTL    0    Select Tool Coordinate System No. 0
PTPR         Specify right arm of PTP target arm system
MOVP    5    Move to Position No. 5.
MOVP    6    Move to Position No. 6.
     :
     :
```



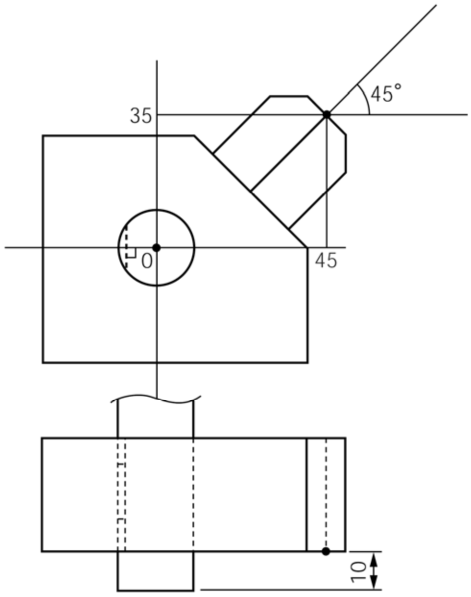

The R-axis position is shown to the left (viewed from above).
The Z-axis position is as follows:
Position No. 5 Zb = 25
Position No. 6 Zb = 45

(3) Tool coordinate system

1) Setting of Tool Coordinate System

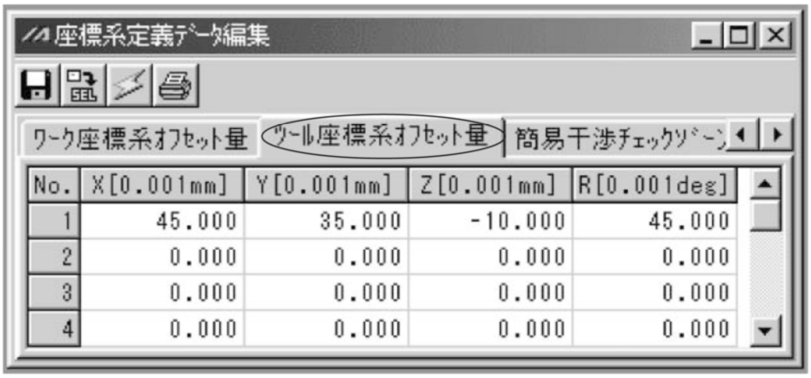
Set the offset from the center of the tool installation surface to tip of the tool.

(Example) Setting example of tool coordinate system
Define Tool Coordinate System No. 1 as shown below.




Offsets under Tool Coordinate System No. 1: Xoft1 = 45, Yoft1 = 35, Zoft1 = -10, Roft1 = 45

Shown below is the edit screen for tool coordinate system definition data in the PC software for horizontal articulated robot, where Tool Coordinate System No. 1 has been set.



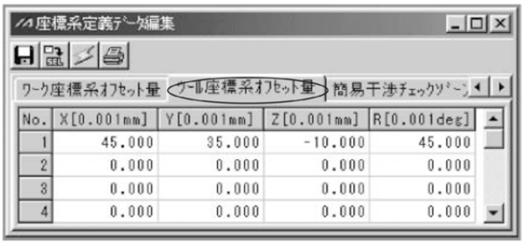

| No. | X[0.001mm] | Y[0.001mm] | Z[0.001mm] | R[0.001deg] |
|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 |
| 2 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.000 | 0.000 |

* To set a tool coordinate system offset in the SEL program, use a DFTL command.

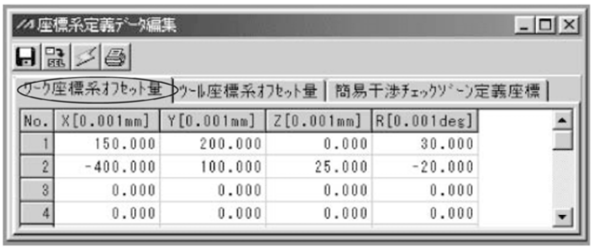
2) Positioning Using Tool Coordinate System Offset
Perform positioning after selecting the tool coordinate system you want to use.
To use a tool coordinate system number in the SEL program, use a SLTL command.
The tool coordinate system selection number that has been selected will remain effective even after the program ends or after the system-memory backup battery has been set and power has been reconnected. (For XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD, MSEL-PCX/PGX battery is not necessary.)

> The orientation of the tool coordinate system axis is always the same as the orientation of the base coordinate system axis on Three-Axis Type SCARA ROBOT (IXP-3N****). As there is no R axis, the control of the orientation (posture) of the tool cannot be conducted. Therefore, there may be a case the tool end may not be the reachable point at positioning. Also, the setting of the tool coordinate system offset on the R axis will be ignored.

(Example 1) Position the tip of the tool on Tool Coordinate System No. 1 to Position No. 5 and No. 6 on Work Coordinate System No. 1 via PTP operation.

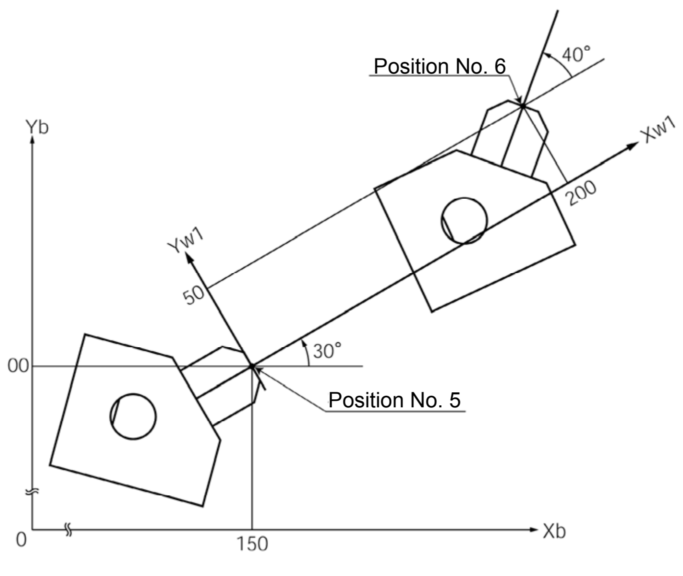



| No. | X[0.001mm] | Y[0.001mm] | Z[0.001mm] | R[0.001deg] |
|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 |
| 2 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.000 | 0.000 |




| No. | X[0.001mm] | Y[0.001mm] | Z[0.001mm] | R[0.001deg] |
|---|---|---|---|---|
| 1 | 150.000 | 200.000 | 0.000 | 30.000 |
| 2 | -400.000 | 100.000 | 25.000 | -20.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.000 | 0.000 |


| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 4( ) | | | | | | | |
| 5( ) | 0.000 | 0.000 | 0.000 | 0.000 | | | |
| 6( ) | 200.000 | 50.000 | 20.000 | 40.000 | | | |
| 7( ) | | | | | | | |

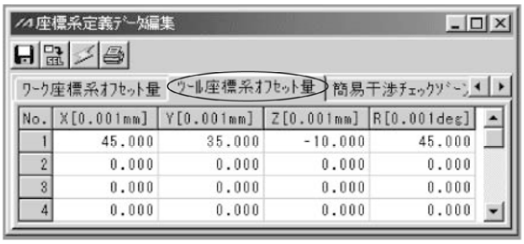



Program example

```
:
:
SLWK    1    Select Work Coordinate System No. 1
SLTL    1    Select Tool Coordinate System No. 1
PTPR         Specify right arm of PTP target arm system
MOVP    5    Move to Position No. 5.
MOVP    6    Move to Position No. 6.
:
:
```

The Z-axis position at the tip of the tool is as follows:
Position No. 5    Zb = 0
Position No. 6    Zb = 20
The figure on the left is viewed from above.

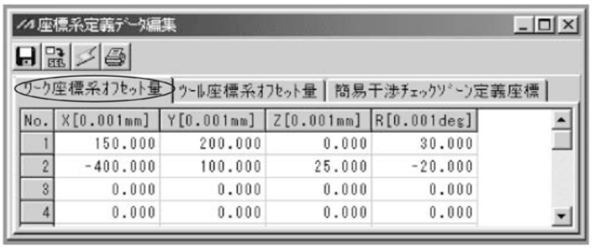
(Example 2) Position the tip of the tool on Tool Coordinate System No. 1 to Position No. 5 and No. 6 on Work Coordinate System No. 2 via PTP operation.



座標系定義データ編集

ワーク座標系オフセット量 | ツール座標系オフセット量 | 簡易干渉チェックゾーン

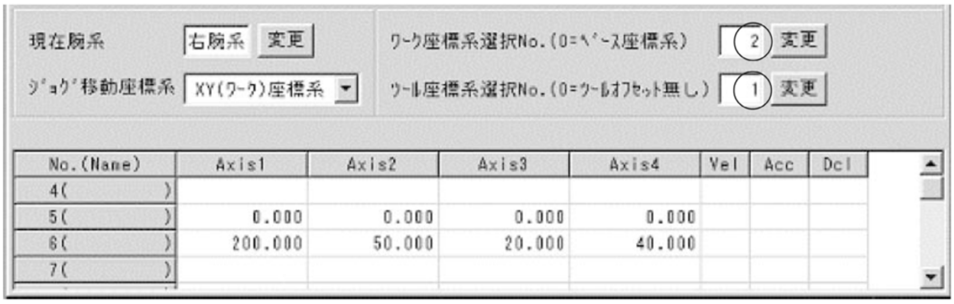
| No. | X[0.001mm] | Y[0.001mm] | Z[0.001mm] | R[0.001deg] |
|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 |
| 2 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.000 | 0.000 |



座標系定義データ編集

ワーク座標系オフセット量 | ツール座標系オフセット量 | 簡易干渉チェックゾーン定義座標

| No. | X[0.001mm] | Y[0.001mm] | Z[0.001mm] | R[0.001deg] |
|---|---|---|---|---|
| 1 | 150.000 | 200.000 | 0.000 | 30.000 |
| 2 | -400.000 | 100.000 | 25.000 | -20.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.000 | 0.000 |



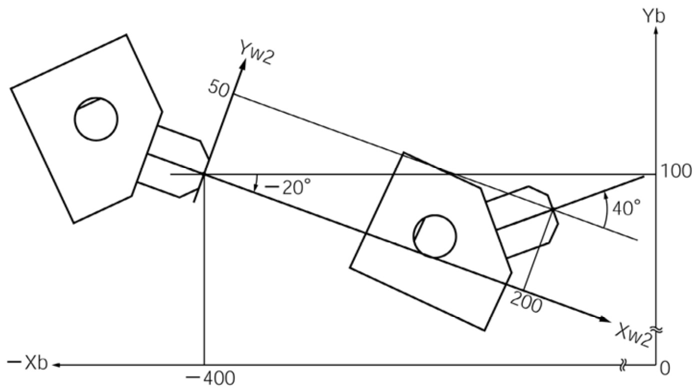
現在腕系 右腕系 変更

ジョグ移動座標系 XY(ワーク)座標系

ワーク座標系選択No.(0=ベース座標系) 2 変更

ツール座標系選択No.(0=ツールオフセット無し) 1 変更

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 4( ) | | | | | | | |
| 5( ) | 0.000 | 0.000 | 0.000 | 0.000 | | | |
| 6( ) | 200.000 | 50.000 | 20.000 | 40.000 | | | |
| 7( ) | | | | | | | |




Program example

```
      :
      :
      :
SLWK     2     Select Work Coordinate System No. 2
SLTL     1     Select Tool Coordinate System No. 1
PTPR           Specify right arm of PTP target arm system
MOVP     5     Move to Position No. 5.
MOVP     6     Move to Position No. 6.
      :
      :
      :
```

The Z-axis position at the tip of the tool is as follows:

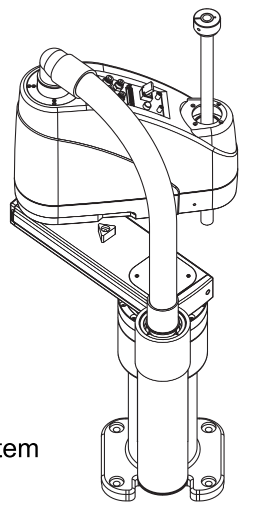
Position No. 5 Zb = 25

Position No. 6 Zb = 45

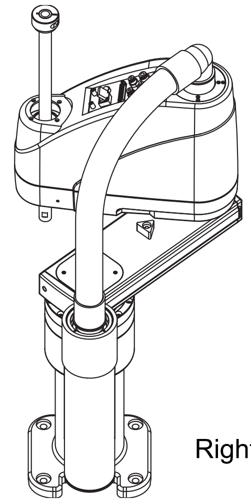
[4] Arm System

(1) Right arm system/left arm system

Robot postures are classified into two types: right arm system and left arm system.



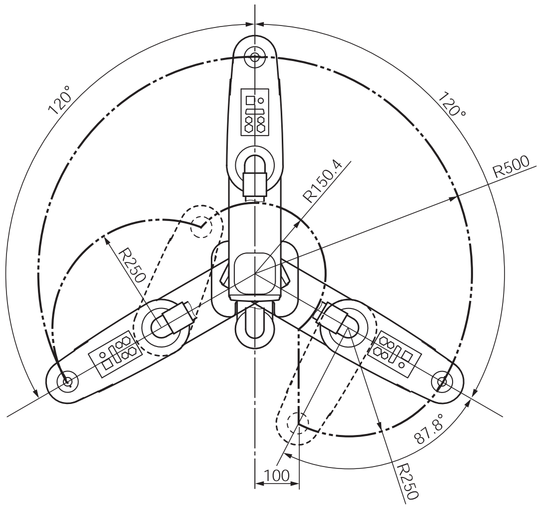
Left arm system



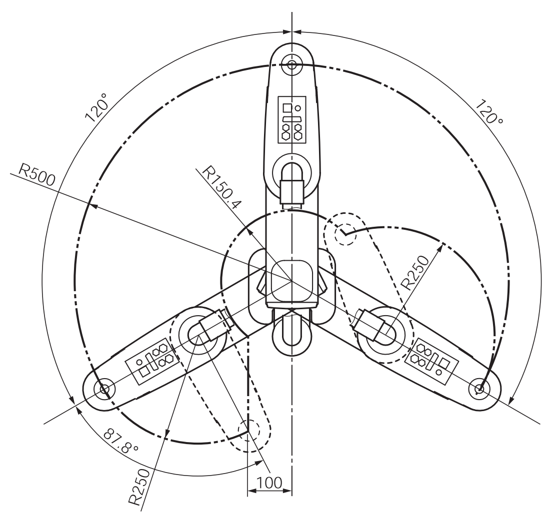
Right arm system

Right arm system : Condition where arms 1 and 2 extend straight and arm 2 is positioned in the CCW direction.
Left arm system : Condition where arms 1 and 2 extend straight and arm 2 is positioned in the CW direction.
The conditions of robot arms are expressed by assuming them as human arms.

The operation area is different between the right arm system and left arm system.
The figure below shows the operation area of each arm system of a robot whose arm length is 500mm.




Operation area of left arm system



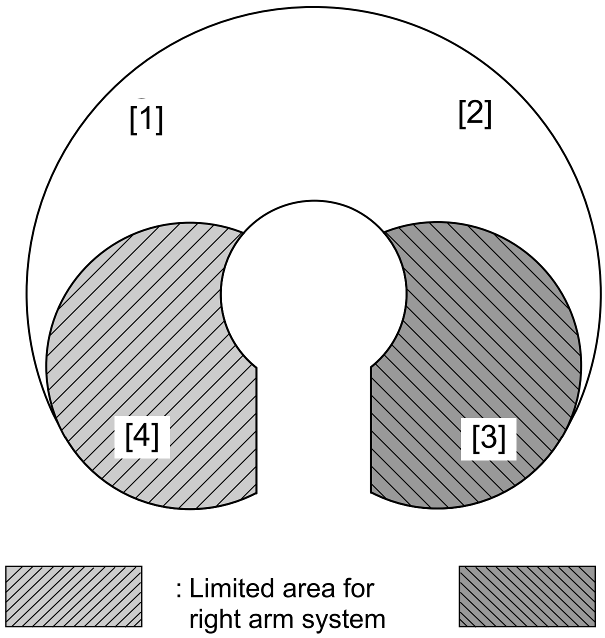

Operation area of right arm system

(2) Arm control commands (dedicated SCARA commands)
The left arm system is defined as “opposite arm system” of the right arm system, and vice versa.
The actual arm system currently used is defined as “current arm system”.
The arm system scheduled to be used for positioning to the target under a movement command is defined as “target arm system”.

Commands that are used to control the robot arm system include PTPD, PTPE, PTPR, PTPL, RIGH and LEFT.
PTPD, PTPE, PTPR and PTPL are control declarations for the target arm system of PTP operation, so they remain valid throughout the program once declared. In the case of CP operation where the arm system does not change, operation is performed based on the current arm system without being affected by the above commands.
Only one of PTPD, PTPE, PTPR and PTPL, whichever is executed last, remains valid.
RIGH and LEFT are control commands for the current arm system.

(3) Arm system control commands and arm system changes
Arm system commands and how the arm system changes during PTP operation as a result of their declaration are explained.

Set position No. 1 to 4 as shown below ([1] to [4]).
Try moving the actuator using a MOVP command (PTP operation) in the order of 1 → 2 → 3 → 2 → 1 → 4.
Move the robot while it is positioned at position No. 1.
Position No. 3 is inside the limited area for left arm system (positioning to this position is not possible if the right arm system is used).
Position No. 4 is inside the limited area for right arm system (positioning to this position is not possible if the left arm system is used).




:
:
:
Arm system control command

```
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4
EXIT
```

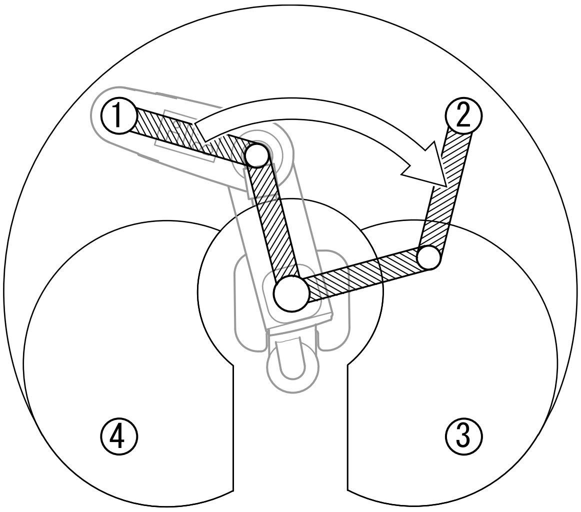
How the arm system changes with an arm system control command is explained for each command.

In the figure, the black arrows indicate movements where the arm system changes.
White arrows indicate movements where the arm system does not change.
The shaded arm represents the right arm system. The unshaded arm represents the left arm system.

[PTPD]

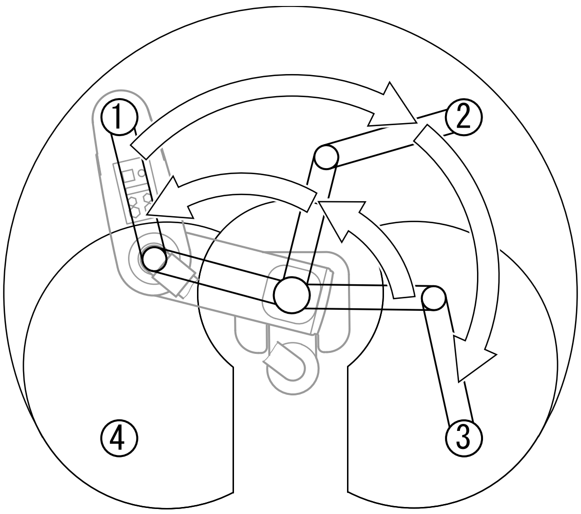
Following the execution of a PTPD command, the robot performs positioning by moving according to the current arm system. The PTPD command prohibits situations where the current arm system is opposite the target arm system. An attempt to move to an area to which positioning is impossible without changing to the opposite arm system generates an error "C73: Target path soft limit over error". Even when a PTPD command is not executed, this command is already effective on the robot when the program is started.

1) Starting from right arm system




```
:
:
:
PTPD
MOVP    2
MOVP    3 ⇒ C73 error occurs.
```

2) Starting from left arm system

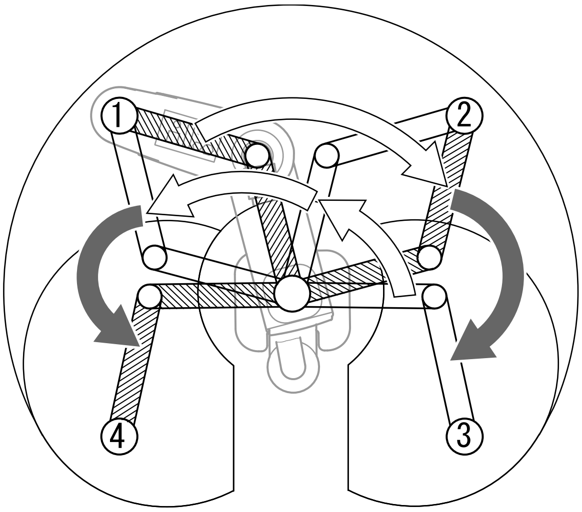



```
:
:
:
PTPD
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

[PTPE]

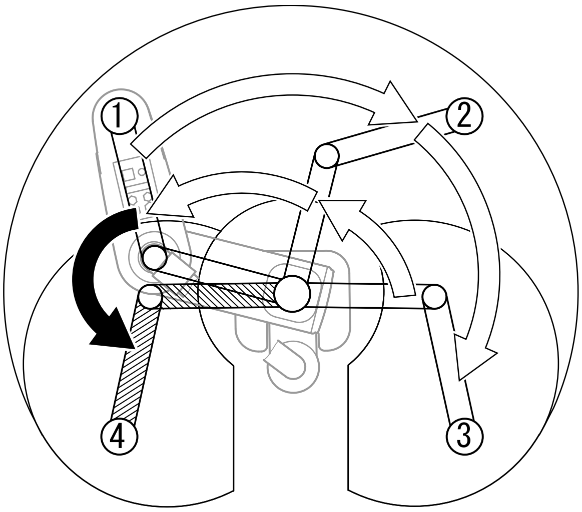
Following the execution of a PTPE command, the robot gives priority to the current arm system for movement and positioning. The PTPE command permits situations where the current arm system is opposite the target arm system. Therefore, it is permitted to move to an area to which positioning is impossible without changing to the opposite arm system.
To prohibit moving to the area for opposite arm system after permitting such movement, execute a PTPD command.

① Starting from right arm system




```
:
:
:
PTPE
MOVP   2
MOVP   3
MOVP   2
MOVP   1
MOVP   4
EXIT
```

② Starting from left arm system

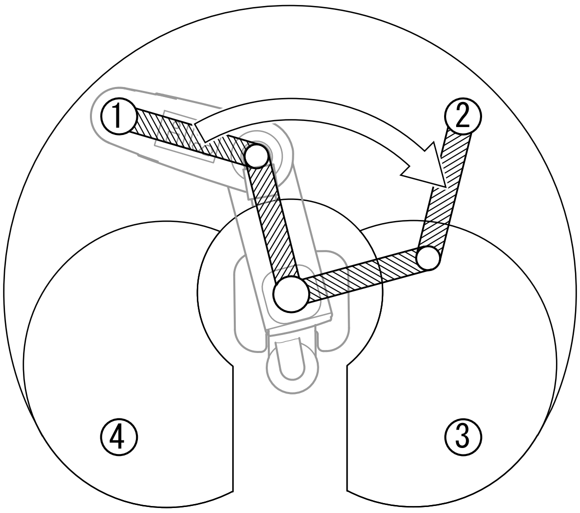



```
:
:
:
PTPE
MOVP   2
MOVP   3
MOVP   2
MOVP   1
MOVP   4
EXIT
```

[PTPR]

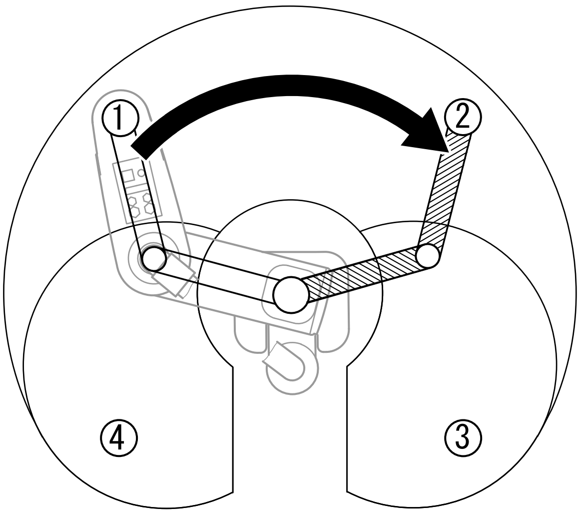
Following the execution of a PTPR command, the robot performs positioning according to the right arm system. The PTPR command limits the target arm system to the right arm system. Therefore, an attempt to move to an area to which positioning is impossible without changing to the left arm system generates a “C73: Target path soft limit over error”.
Executing a PTPR command alone does not initiate any arm movement.
When a PTP movement command is executed following the execution of a PTPR command and while the current arm system is the left arm system, the robot moves as it changes from the left arm system to right arm system and performs positioning according to the right arm system.

① Starting from right arm system




```
  :
  :
  :
PTPR
MOVP    2
MOVP    3 ⇒ C73 error occurs.
```

② Starting from left arm system

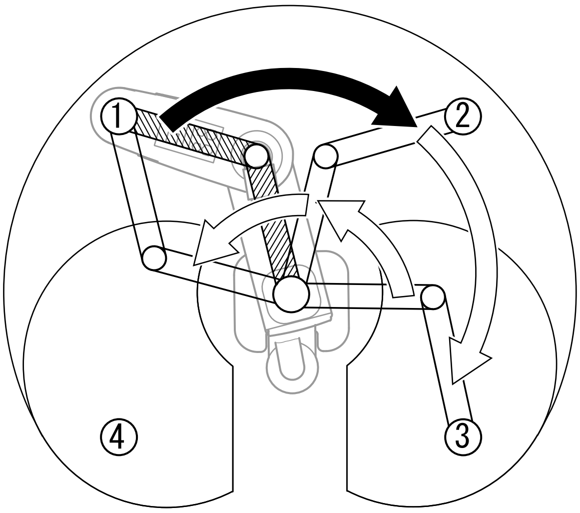



```
  :
  :
  :
PTPR
MOVP    2
MOVP    3 ⇒ C73 error occurs.
```

[PTPL]

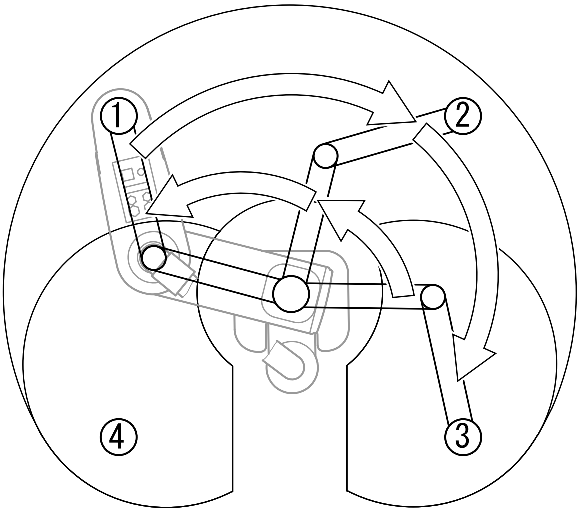
Following the execution of a PTPL command, the robot performs positioning according to the left arm system. The PTPL command limits the target arm system to the left arm system. Therefore, an attempt to move to an area to which positioning is impossible without changing to the right arm system generates a “C73: Target path soft limit over error”.
Executing a PTPL command alone does not initiate any arm movement.
When a PTP movement command is executed following the execution of a PTPL command and while the current arm system is the right arm system, the robot moves as it changes from the right arm system to left arm system and performs positioning according to the left arm system.

① Starting from right arm system




```
:
:
:
PTPL
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

② Starting from left arm system

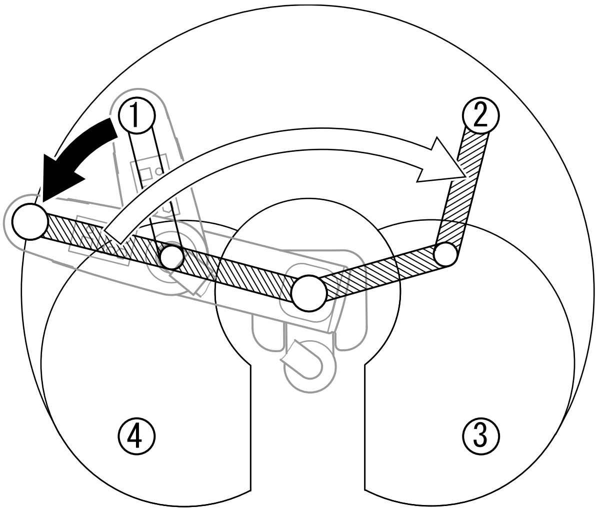



```
:
:
:
PTPL
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

[RIGH]

The RIGH command changes the current arm system to the right arm system.
When a RIGH command is executed while the current arm system is the left arm system, arm 2 operates in such a way that both arms 1 and 2 form a straight line.
Executing a RIGH command while the current arm system is the right arm system does not initiate any arm movement.

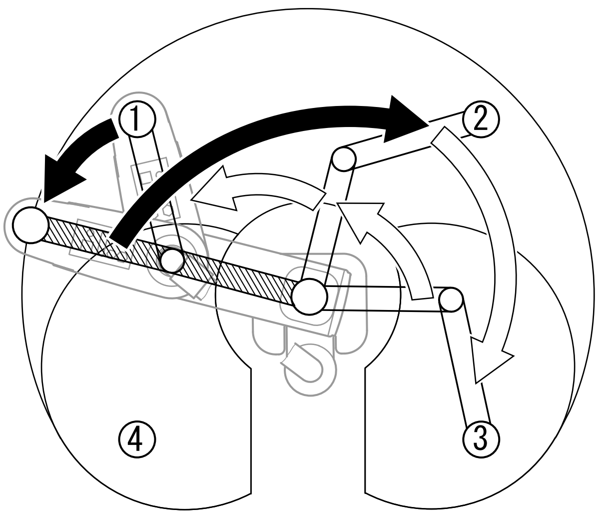
① Starting from left arm system




```
:
:
:
RIGH
MOVP    2
MOVP    3 ⇒ C73 error occurs.
```

In the above example, the PTPD command is effective because no arm system control command other than RIGH is set.
The RIGH command only controls the current arm system. It does not limit positioning via PTP operation to the right arm system. The arm system used for positioning varies depending on the control declaration of target arm system (PTPD, PTPE, PTPR, PTPL).
Accordingly, the specific operation that takes place after the execution of a RIGH command varies depending on the control declaration of target arm system which is currently effective.

② RIGH command at PTPL command execution

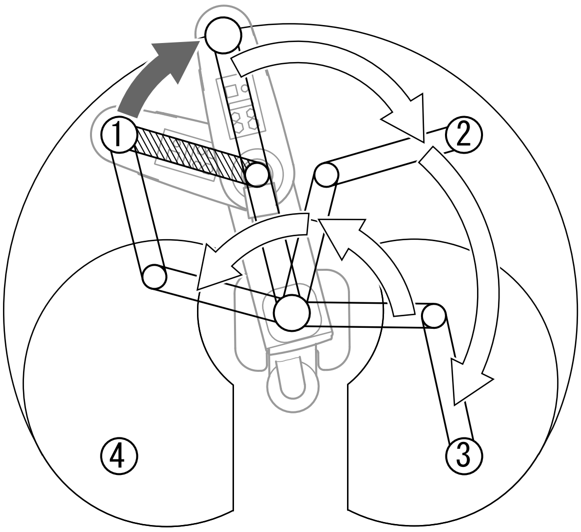



```
:
:
:
PTPL
:
:
:
RIGH
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

[LEFT]

The LEFT command changes the current arm system to the left arm system.
When a LEFT command is executed while the current arm system is the right arm system, arm 2 operates in such a way that both arms 1 and 2 form a straight line.
Executing a LEFT command while the current arm system is the left arm system does not initiate any arm movement.

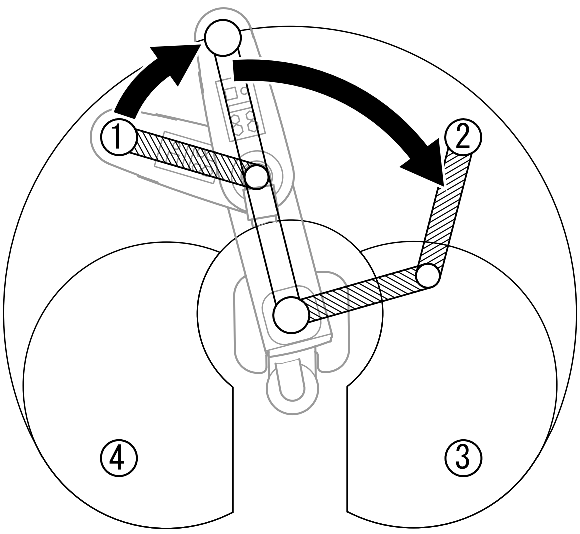
① Starting from right arm system




```
:
:
:
LEFT
MOVP    2
MOVP    3
MOVP    2
MOVP    1
MOVP    4 ⇒ C73 error occurs.
```

In the above example, the PTPD command is effective because no arm system control command other than LEFT is set.
The LEFT command only controls the current arm system. It does not limit positioning via PTP operation to the left arm system. The arm system used for positioning varies depending on the control declaration of target arm system (PTPD, PTPE, PTPR, PTPL).
Accordingly, the specific operation that takes place after the execution of a LEFT command varies depending on the control declaration of target arm system which is currently effective.

② LEFT command at PTPR command execution




```
:
:
:
PTPR
:
:
:
LEFT
MOVP    2
MOVP    3 ⇒ C73 error occurs.
```

[4] PTP Acceleration/Deceleration Optimization Function

IXA Type, IX-***H Type and IXP Type operates in the optimum acceleration / deceleration speed during the PTP operation.

(Note) Those such as IX-NNN5020 do not operate in the optimum acceleration/deceleration speed during the PTP operation. For those models, the maximum acceleration/deceleration speed during the PTP operation relies on the values set in Axis Parameters No. 134 “Maximum PTP acceleration for SCARA axis” and No. 135 “Maximum PTP deceleration for SCARA axis”.
The acceleration for PTP operation corresponds to the ratio (%) set according to the ACCS command and DCLS command.

(1) Function overview
PTP optimum acceleration/deceleration is the automatic adjustment function to obtain the optimized acceleration and deceleration responding to the conditions of such facts as the tip load. In PTP optimum acceleration/deceleration, it is necessary to set the tip load mass with the WGHT Command as well as the acceleration and deceleration ratios with ACCS and DCLS Commands set for the existing models. Set an appropriate load mass according to the load to be transported, etc.
The calculation formats of PTP acceleration and deceleration in PTP optimum acceleration/deceleration are as shown below:

- PTP acceleration = Maximum acceleration determined by the load mass, etc. × ACCS command [%]
- PTP deceleration = Maximum deceleration determined by the load mass, etc. × DCLS command [%]

* The WGHT command is supported by XSEL-PX/QX controllers of main application Ver.0.45 or later.
This command can be input in PC software of Ver.7.5.0.0 or later or on teaching pendants TB-02(D) : first edition or later, TB-01(D) : first edition or later, SEL-T (D) of Ver.1.11 or later. (Not applicable for IA-T-X(0))

Caution

- PTP optimum acceleration/deceleration would not work with the ideal acceleration or deceleration unless the setting of the WGHT Command according the actual robot tip load is conducted. Make sure to setup the tip load mass setting in the WGHT Command.

- PTP optimum acceleration/deceleration function is effective only for the PTP operation. It cannot be operated with the optimized acceleration or deceleration for CP operation or direct-movement axis.

- If an overload error occurs, lower the acceleration setting and/or deceleration setting as deemed appropriate or make other adjustment such as providing a stopping time after acceleration/deceleration to prevent an overload error from occurring.

[5] Horizontal Movement Optimizing Function responding to Z-axis position

SCARA Robot (IX-***H) can utilize the horizontal movement optimizing function.

(Note) Note that the horizontal movement Z-position optimization function is not available for those such as IX-NNN5020. (Using this function would generate a “D8A: Internal parameter error for acceleration/deceleration optimization or horizontal movement Z-position optimization function”.)

(1) Function overview

Horizontal movement optimizing function by Z-axis is the function to optimize the horizontal movement condition based on Z-axis position and the tip load mass.
This function can be set effective/ineffective in the all axes parameter No. 51. When a parameter setting change is made, make sure to reset the software or reboot the system after the flash ROM writing is complete.
The tip load mass setting by the WGHT Command is required while the SCARA Z-axis position and horizontal movement optimizing function are effective. Set the load mass setting accordingly following the transporting work figure.

● All-axis common parameters

| No. | Parameter name | Default value (reference) | Input range | Unit | Access privilege | Remarks |
|---|---|---|---|---|---|---|
| 51 | SCARA-axis control 1 | 0H | 0H to FFFFFFFFH | | F | Bits 8 to 11: Z-axis Position -> horizontal movement optimized (PTP)<br>(0: Disable, 1: Enable)<br>(Available only on high-speed SCARA robots with main application Ver.0.45 or later)<br>Bits 12 to 15: Z-axis Position -> horizontal movement optimized (CP)<br>(0: Disable, 1: Enable)<br>* Disabling this function is recommended if the CP operation requires constant speed, path precision and attainment of specified speed.<br>(Available only on high-speed SCARA robots with main application Ver.0.45 or later) |

* The WGHT command is supported by XSEL-PX/QX controllers of main application Ver.0.45 or later.
This command can be input in PC software of Ver.7.5.0.0 or later or on teaching pendants TB-02(D) : first edition or later, TB-01(D) : first edition or later, SEL-T (D) of Ver.1.11 or later. (Not applicable for IA-T-X(0))

⚠ **Caution**

- It is necessary to set the tip load mass with the WGHT Command while the horizontal movement optimizing function by Z position is activated. An appropriate result could not be gained unless the mass setting according the actual robot tip load is conducted.

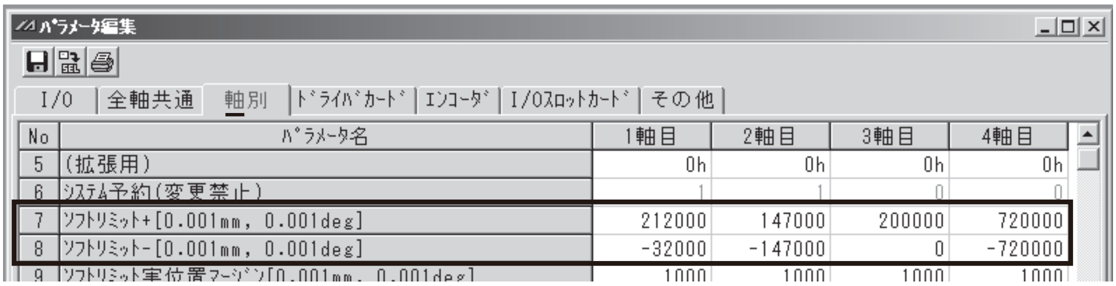
- When the horizontal movement optimizing function by Z position is activated, the speed may not reach the set speed due to the robot load mass or movement position. Make the horizontal movement optimization function invalid if it is required to reach the set speed.
  - * When also it is indicated the Operation 1 = 0 (prioritized to reach set speed automatic division) for DIS (divide distance setting for spline movement) and DIG (arc angle setting), the horizontal movement optimization function should be prioritized and may not reach the set speed.

- When operating individually with the PATH, CIR, ARC, CIRS, ARCS or PSPL Command while the horizontal movement optimization (CP) by Z position is activated, the movement speed during the command may vary due to the robot load mass or the movement position. In a continuous operation with the continuous operation related commands (PATH, PSPL, CIR2, ARC2, CIRS, ARCS, CIRS, ARCD, ARCC, CIR, ARC Commands, etc.), the movement speed between the commands may vary due to the operational condition. Make the horizontal movement optimization (CP) invalid if an evenly paced speed is required.

- When the horizontal movement optimization (CP) by Z axis is activated, the track of CP operation may slightly vary due to the robot load mass and movement position. If accuracy in the track is required, make the horizontal movement optimization (CP) invalid.

[6] Soft Limit

The soft limit is set in axis-specific parameter No. 7 and 8. Below is an example of a screen showing the soft limits for IX5020 (arm length 500mm, Z-axis 200mm).


パラメータ編集

I/O | 全軸共通 | 軸別 | ドライバカード | エンコーダ | I/Oスロットカード | その他

| No | パラメータ名 | 1軸目 | 2軸目 | 3軸目 | 4軸目 |
|---|---|---|---|---|---|
| 5 | (拡張用) | 0h | 0h | 0h | 0h |
| 6 | システム予約(変更禁止) | 1 | 1 | 0 | 0 |
| 7 | ソフトリミット+[0.001mm, 0.001deg] | 212000 | 147000 | 200000 | 720000 |
| 8 | ソフトリミット-[0.001mm, 0.001deg] | -32000 | -147000 | 0 | -720000 |
| 9 | ソフトリミット実位置マージン[0.001mm, 0.001deg] | 1000 | 1000 | 1000 | 1000 |

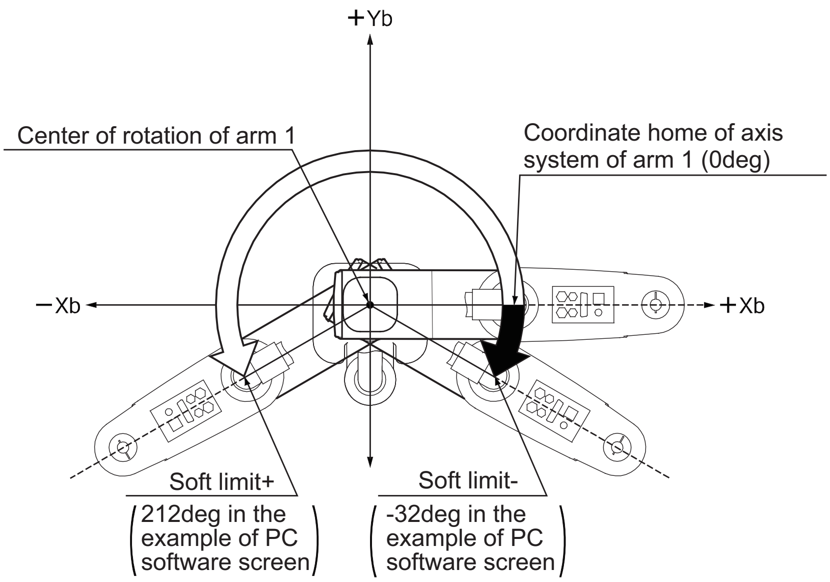
The soft limit parameters are set by coordinate values according to each axis system.
Axis 1 corresponds to arm 1, axis 2 corresponds to arm 2, axis 3 corresponds to Z-axis, and axis 4 corresponds to R-axis.
The setting units is [0.001deg] for arm 1, arm 2 and R-axis (rotational axis). The setting unit for Z-axis is [0.001mm].
The soft limits are used to limit the range of operation of arm 1, arm 2, Z-axis or R-axis from the coordinate home of each axis system. It is not affected by the work coordinates system or tool coordinate systems.

(Note) These parameters have been set to the maximum limits of range of operation at the factory. Do not set values that would enlarge the range of operation.

(1) Coordinates on each axis system and soft limits

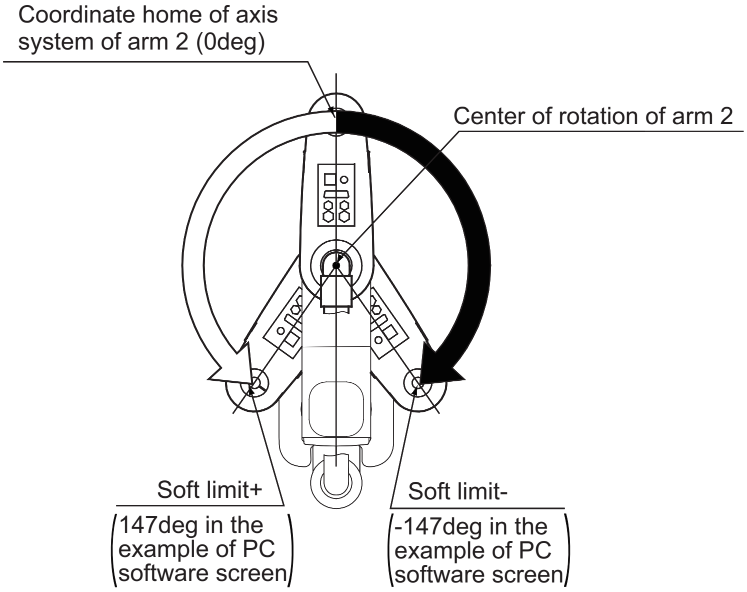
[Soft limits for arm 1]

The arm 1 position at which the arm faces the +Xb direction defines the coordinate home of the axis system of arm 1 (0deg).
This position is not affected by the arm 2 position.
Operating angles in the counterclockwise direction (positive direction) from this coordinate home of axis system are limited by the soft limit+ (axis 1 of axis-specific parameter No. 7).
Operating angles in the clockwise direction (negative direction) are limited by the soft limit- (axis 1 of axis-specific parameter No. 8).



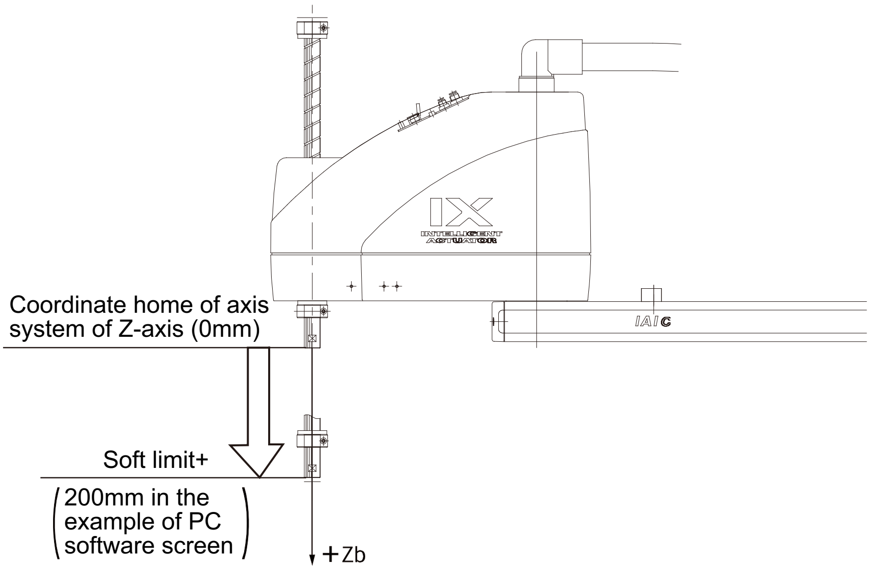
[Soft limits for arm 2]

The arm 2 position at which the arm forms a straight line with arm 1 defines the coordinate home of the axis system of arm 2 (0deg).
This position is not affected by the arm 1 position.
Operating angles in the counterclockwise direction (positive direction) from this coordinate home of axis system are limited by the soft limit+ (axis 2 of axis-specific parameter No. 7).
Operating angles in the clockwise direction (negative direction) are limited by the soft limit- (axis 2 of axis-specific parameter No. 8).



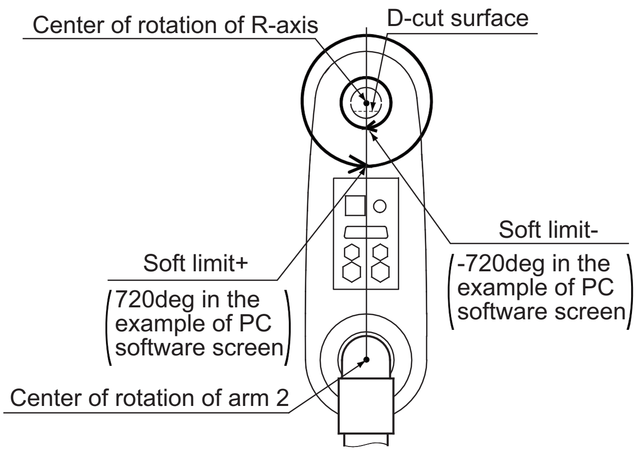

[Soft limits for Z-axis]

The Z-axis position at which the mechanical stopper attached to the Z-axis is approx. 5mm below the mechanical end at the bottom of arm 2 defines the coordinate home of the axis system of Z-axis (0mm). This position is the same as the Axis 3 = 0mm position on the base coordinate system. (On actuators of clean-room specification and dust-proof/splash-proof specification, this mechanical stopper is not visible because it is located inside bellows.)
Movements in the downward direction (positive direction) from this coordinate home of axis system are limited by the soft limit+ (axis 3 of axis-specific parameter No. 7). Movements in the upward direction (negative direction) from this coordinate home of axis system are limited by the soft limit- (axis 3 of axis-specific parameter No. 8). (The relationship is reversed on actuators of inverse specification.)



[Soft limits for R-axis]

The R-axis position at which the D-cut surface at the tip of the axis faces the center of rotation of arm 2 defines the coordinate home of the axis system of R-axis (0deg). This position is not affected by the arm 1 or arm 2 position.



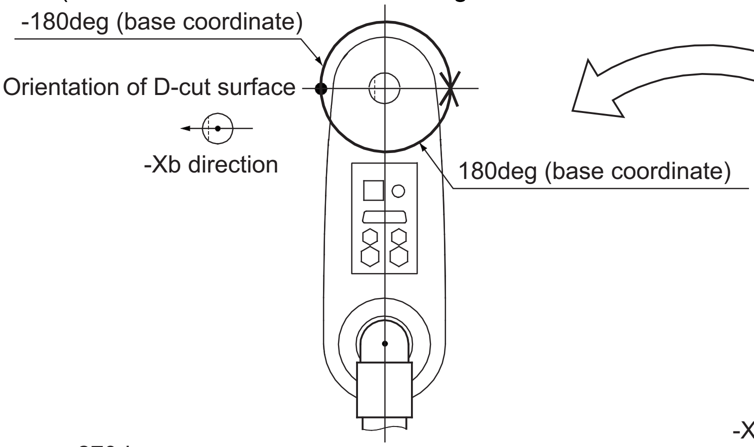

Operating angles in the counterclockwise direction (positive direction) from this coordinate home of axis system are limited by the soft limit+ (axis 4 of axis-specific parameter No. 7). Operating angles in the clockwise direction (negative direction) are limited by the soft limit- (axis 4 of axis-specific parameter No. 8).

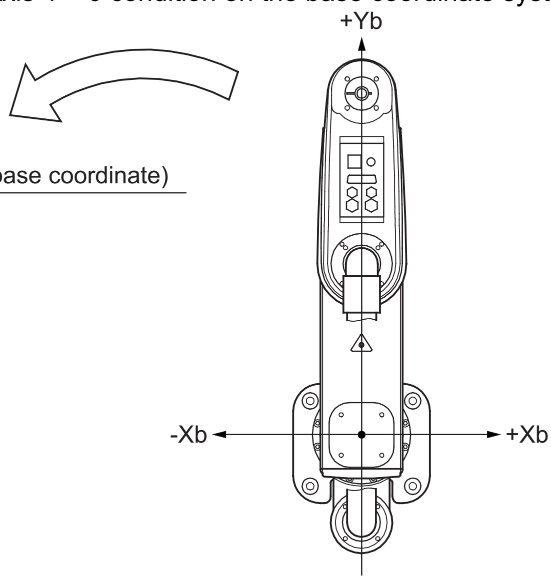
When limiting the operating range of the R-axis, you must pay attention to the difference between the base coordinate system and this axis system.

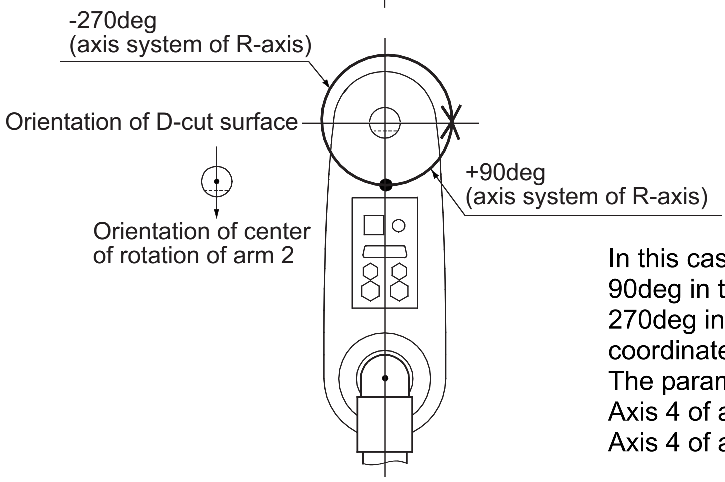
(Example)

Limit the range of operation of the R-axis to ±180 from the position shown below.

(Limit the R-axis to within ±180deg in the Axis 4 = 0 condition on the base coordinate system.)



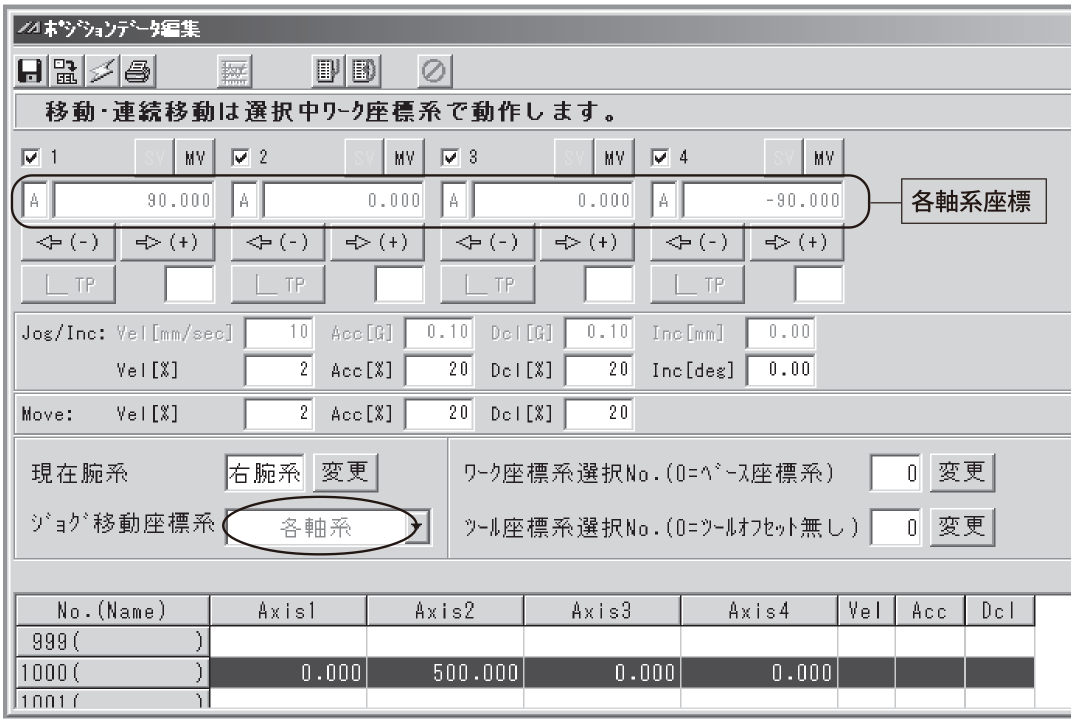

In this case, the R-axis is limited to within 90deg in the positive direction and within 270deg in the negative direction from the coordinate home of the axis system of R-axis.

The parameters below are set as follows:

Axis 4 of axis-specific parameter No. 7 = 90000

Axis 4 of axis-specific parameter No. 8 = -270000

(2) Monitoring of axis system coordinates
You can use the PC software or teaching pendant to monitor axis system coordinates. Shown below is an example of a PC software screen. When the jog movement coordinate system is selected for each axis system in the position data edit window, the currently displayed position switch to a coordinate based on the selected axis system.




(An IX5020 (arm length 500mm, Z-axis 200mm) is located at the position of Axis 1 = 0, Axis 2 = 500, Axis 3 = 0, Axis 4 = 0 on the base coordinate system.)

(Note) Position data cannot be loaded in each axis system.

[For details on the specific operating procedure, refer to the Instruction Manual for your PC software or teaching pendant.]

[7] Simple Contact Check Zone

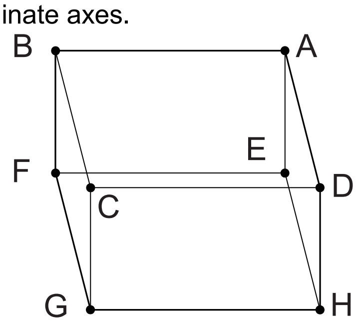
The simple contact check zone is an area you must set when checking for contact between the robot and nearby equipment.
When tool coordinate system No. 0 (= tool coordinate system offset 0) is selected, you can detect an entry into the simple contact check zone by the center position of the tool mounting surface. When any one of tool coordinate system No. 1 to 127 (= tool coordinate system offset enabled) is selected, you can detect a similar entry by the tool tip position.

[Notes on use of simple contact check zone]

- An entry into the simple contact check zone by the center position of the tool mounting surface (when tool coordinate system No. 0 is selected) or tool tip position (when any one of tool coordinate system No. 1 to 127 is selected) is detected. An entry by the outer periphery of the R-axis or any part of the tool other than its tip is not detected.
- This function does not prevent an entry into the simple contact check zone. It only detects an entry after it has occurred.
- An entry cannot be detected reliably unless the applicable position remains inside the simple contact check zone for 5msec or more. This function is intended to provide a means for simple check by low-speed operation.
- The path changes between high-speed operation (actual operation) and low-speed operation. Ensure a sufficient margin to avoid contact. (The robot tends to pass on the inner side of the path during high-speed operation compared to low-speed operation.)
- The coordinates defining the simple contact check zone are always recognized as data of the base coordinate system (work coordinate system selection No. 0). Take note that changing the work coordinate system does not change the position of the simple contact check zone.
  If the coordinates defining the simple contact check zone are changed, it will take 5msec before the check result according to the new coordinates is reflected.
- In PTP operation, the robot does not move along a fixed path. Conduct test operation at low speed to confirm absence of contact near an obstacle (including a part of the robot), and then gradually raise the speed to an appropriate level.
- If physical output port numbers or global flag numbers are duplicated, chattering occurs and operation results become unstable. Do not specify duplicate numbers.
- Use of the simple contact check zone consumes significant CPU power. When this function is not used, disable the function by setting 0 for the applicable “physical output port number/global flag number” and “error type”.
- The simple interference check zone becomes available after the home-operation complete or the absolute coordinate confirmation. Note that interference check cannot be held when home-return operation is incomplete or the absolute coordinate is unconfirmed.
- In Physical Output Port Number and Global Flag Number of Simple Interference Check Zone Definition, numbers to duplicate with those set in System Output Port / Flag Numbers (Output Function Select, Linear Axis Zone, etc.) Error No. 906 “Input and Output Port / Flag Number Error” generates if any duplicated number is indicated.

[Setting of simple contact check zone]

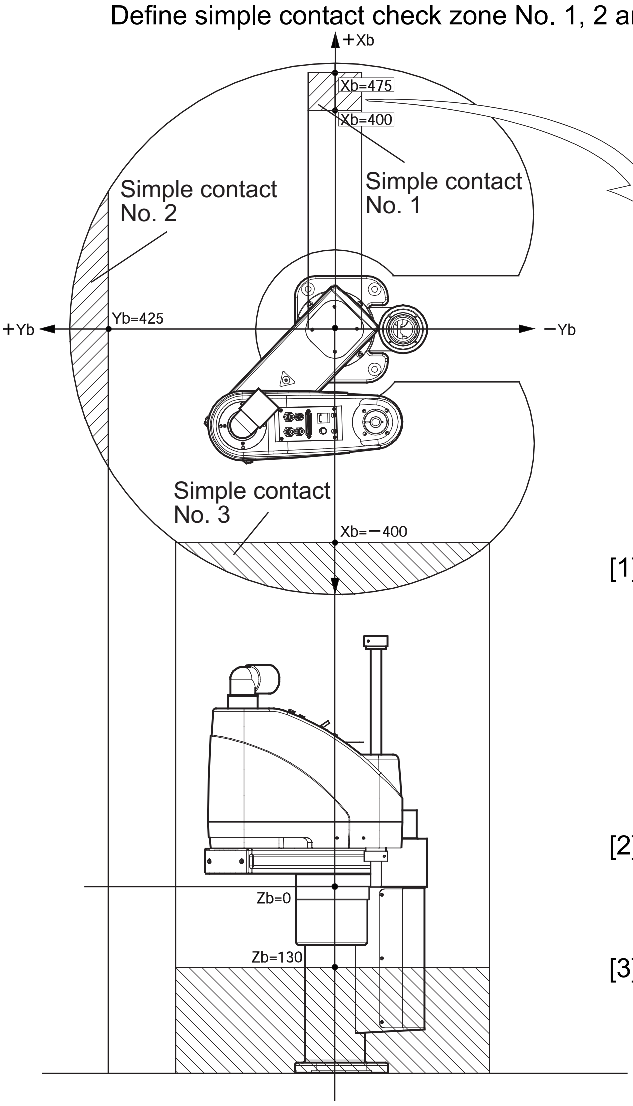
Set the simple contact check zone using position data of the base coordinate system.
Enter the maximum and minimum coordinate values of the simple contact check zone.
Set the boundary surfaces of the simple contact check zone in parallel with the base coordinate axes.

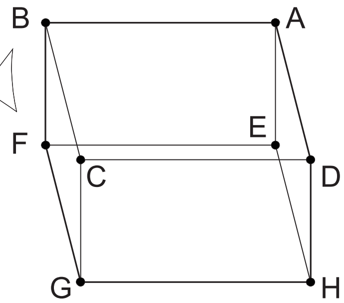



To set a rectangular solid like the one shown to the left as the simple contact check zone, enter the coordinate values of two points according to a combination of A-G, B-H, C-E or D-F.

(Example) Setting example of simple contact check zones

Define simple contact check zone No. 1, 2 and 3 as shown below.



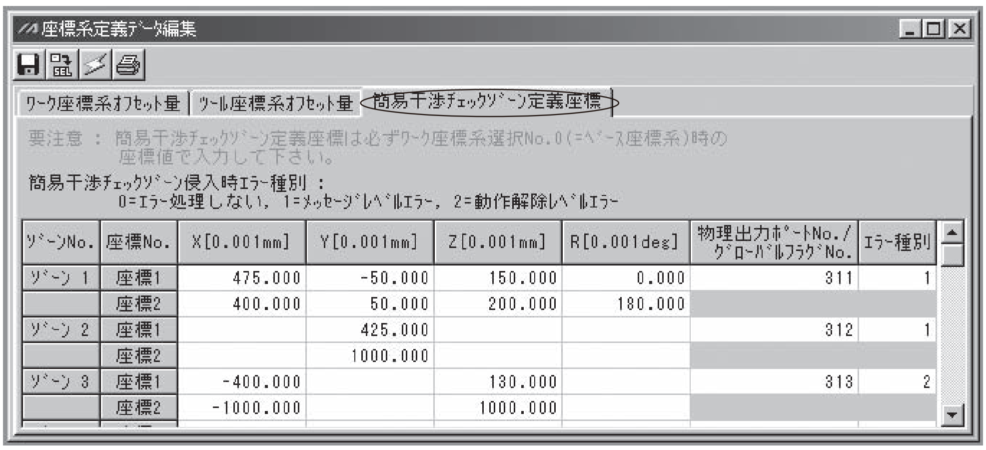




[1] For simple contact check zone No. 1, set a rectangular solid as the simple contact check zone.
A : Xb = 475, Yb = -50, Zb = 150, Rb = 0
G : Xb = 400, Yb = 50, Zb = 200, Rb = 180
An entry into this rectangular solid is not detected if Rb is outside the range of 0 to 180°.

[2] For simple contact check zone No. 2, set an area where Yb is 425mm or more as the simple contact check zone.

[3] For simple contact check zone No. 3, set an area where Xb is -400mm or less and Zb is 130mm or more as the simple contact check zone.

Shown below is the screenshot of the edit window for the simple contact check zone definition data in PC software dedicated for SCARA Robot assuming the simple contact check zones No. 1, No. 2 and No. 3 are set.


座標系定義ﾃﾞｰﾀ編集

ﾜｰｸ座標系ｵﾌｾｯﾄ量 | ﾂｰﾙ座標系ｵﾌｾｯﾄ量 | 簡易干渉ﾁｪｯｸｿﾞｰﾝ定義座標

要注意 ： 簡易干渉ﾁｪｯｸｿﾞｰﾝ定義座標は必ずﾜｰｸ座標系選択No.0(=ﾍﾞｰｽ座標系)時の座標値で入力して下さい。

簡易干渉ﾁｪｯｸｿﾞｰﾝ侵入時ｴﾗｰ種別 ：
0=ｴﾗｰ処理しない， 1=ﾒｯｾｰｼﾞﾚﾍﾞﾙｴﾗｰ， 2=動作解除ﾚﾍﾞﾙｴﾗｰ

| ｿﾞｰﾝNo. | 座標No. | X[0.001mm] | Y[0.001mm] | Z[0.001mm] | R[0.001deg] | 物理出力ﾎﾟｰﾄNo./ｸﾞﾛｰﾊﾞﾙﾌﾗｸﾞNo. | ｴﾗｰ種別 |
|---|---|---|---|---|---|---|---|
| ｿﾞｰﾝ 1 | 座標1 | 475.000 | -50.000 | 150.000 | 0.000 | 311 | 1 |
| | 座標2 | 400.000 | 50.000 | 200.000 | 180.000 | | |
| ｿﾞｰﾝ 2 | 座標1 | | 425.000 | | | 312 | 1 |
| | 座標2 | | 1000.000 | | | | |
| ｿﾞｰﾝ 3 | 座標1 | -400.000 | | 130.000 | | 313 | 2 |
| | 座標2 | -1000.000 | | 1000.000 | | | |

- As for simple contact check zone No. 1, an entry into this rectangular solid is not detected if the Rb is outside the range of 0 to 180°. To detect an entry into this zone regardless of the R-axis coordinate value, leave the coordinate 1 and 2 fields for zone 1 and R blank.

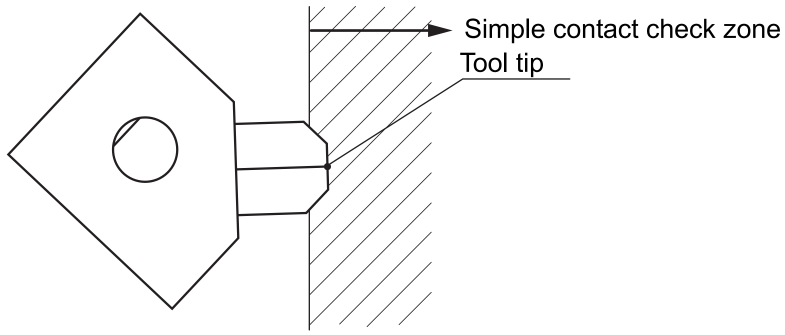
- If either the maximum value or minimum value is not limited, as is the case with simple contact check zone No. 2 and 3, enter a value outside the range of operation (such as 1000 for zone 2, and 1000 or -1000 for zone 3).

- The maximum value and minimum value can be set under either coordinate 1 or 2.

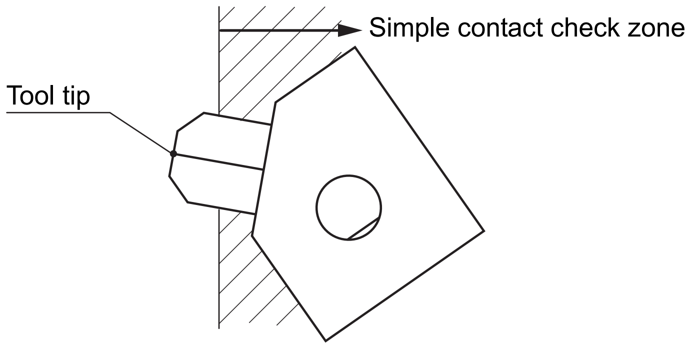
- According to the above settings, output port No. 311 turns ON upon entry into simple contact check zone No. 1, output port No. 312 turns ON upon entry into simple contact check zone No. 2, and output port No. 313 turns ON upon entry into simple contact check zone No. 3.

* Use a DFIF command if you want to set a simple contact check zone within the SEL program.

[Note on detection while tool coordinate system is selected]
While the tool coordinate system is selected, this function detects an entry of the tool tip, not the center of the mounting surface, into the simple contact check zone.




Depending on the movement path, a part of the tool other than its tip may enter the simple contact check zone, as shown below. Exercise due caution because in this case, the entry will not be detected until the tool tip enters the simple contact check zone.



● Caution

In XSEL-RXD/SXD, XSEL-RAXD/SAXD, the definitions of SCARA axes (Axes 1 to 4) are to be set to Axes 1 to 4 and SCARA axes (Axes 5 to 8) to Axes 5 to 8. SCARA axes (Axes 1 to 4) and SCARA axes (Axes 5 to 8) cannot be set in one zone number at the same time. (10 zone definitions are required in total for 2 units of SCARA.)


Each coordinate axis number expresses the meaning listed below.

•Axis 1: Interference domain data of X-axis for SCARA axes (Axes 1 to 4)
•Axis 2: Interference domain data of Y-axis for SCARA axes (Axes 1 to 4)
•Axis 3: Interference domain data of A-axis for SCARA axes (Axes 1 to 4)
•Axis 4: Interference domain data of R-axis for SCARA axes (Axes 1 to 4)
•Axis 5: Interference domain data of X-axis for SCARA axes (Axes 5 to 8)
•Axis 6: Interference domain data of Y-axis for SCARA axes (Axes 5 to 8)
•Axis 7: Interference domain data of Z-axis for SCARA axes (Axes 5 to 8)
•Axis 8: Interference domain data of R-axis for SCARA axes (Axes 5 to 8)

# 2. Connection with Host System

When transferring the data between the host system (PLC, etc.), it can be selected from the following methods(Note 1):

1) Use 24V DC I/O.
2) Use the serial communication (RS232C).
3) Use the Fieldbus communication(Note 2) (option). … This is able to control like I/O.

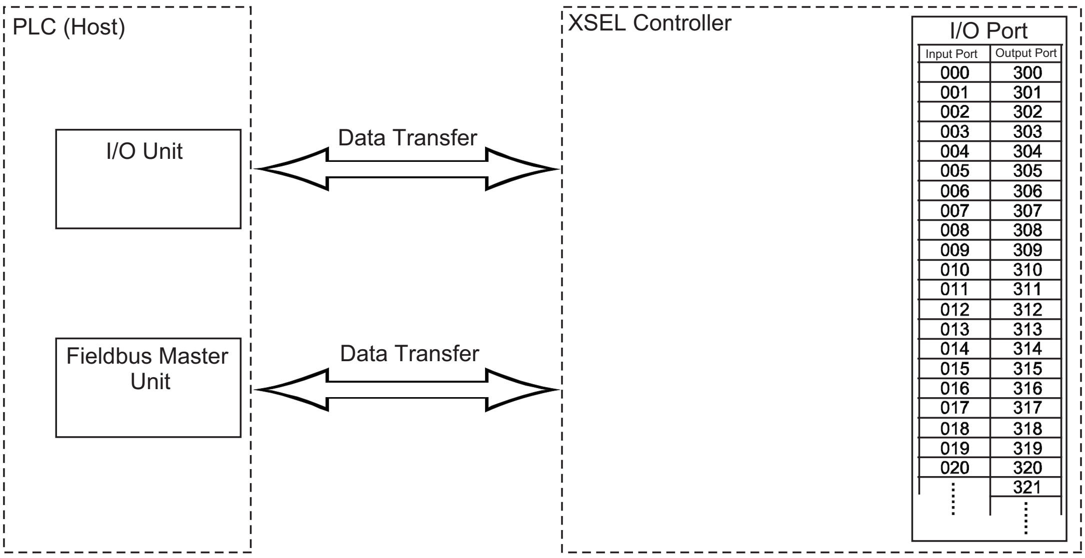
(Note 1) It includes the optional functions.

(Note 2) A dedicated PCB is required separately.
The types of applicable Fieldbus are CC-Link, DeviceNet, PROFIBUS, PROFINET-IO (MSEL-PC/PG/PCX/PGX), Ethernet (for XSEL-J/X/JX/KX/P/Q/PX/QX/TT only), EtherNet/IP (for XSEL-R*/S*, SSEL, ASEL, PSEL, TTA, MSEL-PC/PG/PCX/PGX only) and EtherCAT (for XSEL-R*/S*, TTA, MSEL-PC/PG/PCX/PGX only). [For details, refer to the Fieldbus Instruction Manual provided separately and the Instruction Manual for the host system.]

## 2.1 I/O Signal

There are 2 types of input and output signals as shown below.

1) Input and Output I/O Port
2) Virtual I/O Port




| | Port No. | Function | | Port No. | Function |
|---|---|---|---|---|---|
| | 000 | Program Start | | 300 | Alarm Output |
| | 001 | General-purpose Input | | 301 | Ready Output |
| | 002 | General-purpose Input | | 302 | Emergency Stop Output |
| | 003 | General-purpose Input | | 303 | General-purpose Output |
| | 004 | General-purpose Input | | 304 | General-purpose Output |
| | 005 | General-purpose Input | | 305 | General-purpose Output |
| | 006 | General-purpose Input | | 306 | General-purpose Output |
| | 007 | Program Specification (PRG No.1) | | 307 | General-purpose Output |
| Input | 008 | Program Specification (PRG No.2) | Output | 308 | General-purpose Output |
| | 009 | Program Specification (PRG No.4) | | 309 | General-purpose Output |
| | 010 | Program Specification (PRG No.8) | | 310 | General-purpose Output |
| | 011 | Program Specification (PRG No.10) | | 311 | General-purpose Output |
| | 012 | Program Specification (PRG No.20) | | 312 | General-purpose Output |
| | 013 | Program Specification (PRG No.40) | | 313 | General-purpose Output |
| | 014 | General-purpose Input | | 314 | General-purpose Output |
| | 015 | General-purpose Input | | 315 | General-purpose Output |
| | ⋮ | ⋮ | | ⋮ | ⋮ |

(Note) The numbers of I/O ports are:

Input: 000 to 299 (300 points max.)
Output: 300 to 599 (300 points max.)

### 2.1.1 XSEL-J/K Type Controllers

XSEL-J/K type controllers
- XSEL-J/K/KE/KT/KET
- XSEL-JX/KX/KETX

[1] Input and Output I/O Port
With XSEL-J/K type controllers, the assignments of input and output functions to I/O ports are fixed and cannot be changed.

I/O Signal Table

Input

| Pin No. | Wire color | Port No. | Standard (factory) setting Can be changed by I/O parameter | I/O parameter | |
|---|---|---|---|---|---|
| 1 | Brown-1 | | K, KX types : Cannot be connected.<br>J, JX types : +24V input | | |
| 2 | Red-1 | 000 | Program start | No. 30 | 0: General-purpose input<br>1: Program start (BCD specification)<br>2: Program start (binary specification) |
| 3 | Orange-1 | 001 | General-purpose input | No. 31 | 0: General-purpose input<br>1: Soft reset signal |
| 4 | Yellow-1 | 002 | General-purpose input | No. 32 | 0: General-purpose input<br>1: Servo ON signal |
| 5 | Green-1 | 003 | General-purpose input | No. 33 | 0: General-purpose input<br>1: Auto program start upon power-ON reset or software reset in AUTO mode<br>2: Auto program start signal |
| 6 | Blue-1 | 004 | General-purpose input | No. 34 | 0: General-purpose input<br>1: Software interlock of all servo axes (OFF level) |
| 7 | Purple-1 | 005 | General-purpose input | No. 35 | 0: General-purpose input<br>1: Operation pause cancellation input (ON edge) |
| 8 | Gray-1 | 006 | General-purpose input | No. 36 | 0: General-purpose input<br>1: Operation pause signal (OFF level) |
| 9 | White-1 | 007 | Program number specification (MSB) | No. 37 | 0: General-purpose input<br>1: Program number specification (MSB) |
| 10 | Black-1 | 008 | Program number specification (bit 2) | No. 38 | 0: General-purpose input<br>1: Program number specification (bit 2) |
| 11 | Brown-2 | 009 | Program number specification (bit 3) | No. 39 | 0: General-purpose input<br>1: Program number specification (bit 3) |
| 12 | Red-2 | 010 | Program number specification (bit 4) | No. 40 | 0: General-purpose input<br>1: Program number specification (bit 4) |
| 13 | Orange-2 | 011 | Program number specification (bit 5) | No. 41 | 0: General-purpose input<br>1: Program number specification (bit 5) |
| 14 | Yellow-2 | 012 | Program number specification (bit 6) | No. 42 | 0: General-purpose input<br>1: Program number specification (bit 6) |
| 15 | Green-2 | 013 | Program number specification (LSB: bit 7) | No. 43 | 0: General-purpose input<br>1: Program number specification (LSB: bit 7) |
| 16 | Blue-2 | 014 | General-purpose input | No. 44 | 0: General-purpose input<br>1: Drive-source cutoff cancellation (ON edge) |
| 17 | Purple-2 | 015 | General-purpose input | No. 45 | 0: General-purpose input<br>The following settings are effective only with XSEL-J/K:<br>1: Home return of all effective axes (ON edge)<br>2: Home return of all effective incremental axes (ON edge) |
| 18 | Gray-2 | 016 | General-purpose input | | |
| 19 | White-2 | 017 | General-purpose input | | |
| 20 | Black-2 | 018 | General-purpose input | | |
| 21 | Brown-3 | 019 | General-purpose input | | |
| 22 | Red-3 | 020 | General-purpose input | | |
| 23 | Orange-3 | 021 | General-purpose input | | |
| 24 | Yellow-3 | 022 | General-purpose input | | |
| 25 | Green-3 | 023 | General-purpose input | | |
| 26 | Blue-3 | 024 | General-purpose input | | |
| 27 | Purple-3 | 025 | General-purpose input | | |
| 28 | Gray-3 | 026 | General-purpose input | | |
| 29 | White-3 | 027 | General-purpose input | | |
| 30 | Black-3 | 028 | General-purpose input | | |
| 31 | Brown-4 | 029 | General-purpose input | | |
| 32 | Red-4 | 030 | General-purpose input | | |
| 33 | Orange-4 | 031 | General-purpose input | | |

Output

| Pin No. | Wire color | Port No. | Standard (factory) setting Can be changed by I/O parameter | I/O parameter | |
|---|---|---|---|---|---|
| 34 | Yellow-4 | 300 | Output of operation-cancellation level or higher error (OFF) | No. 46 | 0: General-purpose output<br>1: Output of operation-cancellation level or higher error (ON)<br>2: Output of operation-cancellation level or higher error (OFF)<br>3: Output of operation-cancellation level or higher error + Emergency stop output (ON)<br>4: Output of operation-cancellation level or higher error + Emergency stop output (OFF) |
| 35 | Green-4 | 301 | READY output (PIO-trigger program operation enabled AND no cold-start level or higher error) (Main application version 0.20 or later) | No. 47 | 0: General-purpose output<br>1: READY output (PIO-trigger program operation enabled)<br>2: READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error)<br>(Main application Ver.0.20 or later)<br>3: READY output (PIO-trigger program operation enabled and no cold-start level or higher error)<br>(Main application Ver.0.20 or later) |
| 36 | Blue-4 | 302 | Emergency stop output (OFF) | No. 48 | 0: General-purpose output<br>2: Emergency stop output (ON)<br>3: Emergency stop output (OFF) |
| 37 | Purple-4 | 303 | General-purpose output | No. 49 | 0: General-purpose output<br>1: AUTO mode output<br>2: Auto operation output (When other parameter No. 12 is set to '1') |
| 38 | Gray-4 | 304 | General-purpose output | No. 50 | 0: General-purpose output<br>The following settings are effective only with XSEL-J/K:<br>1: Output when all effective axes are home (= 0)<br>2: Output when all effective axes have completed home return<br>3: Output when all effective axes are at home preset coordinate<br>(Main application Ver.0.21 or later)<br>* To move an actuator of absolute encoder specification to coordinate 0 or the home preset coordinate, use a MOVP command instead of HOME command. |
| 39 | White-4 | 305 | General-purpose output | No. 51 | 0: General-purpose output<br>2: Axis 1 servo ON output<br>(Main application Ver.0.44 or later) |
| 40 | Black-4 | 306 | General-purpose output | No. 52 | 0: General-purpose output<br>2: Axis 2 servo ON output<br>(Main application Ver.0.44 or later) |
| 41 | Brown-5 | 307 | General-purpose output | No. 53 | 0: General-purpose output<br>2: Axis 3 servo ON output<br>(Main application Ver.0.44 or later) |
| 42 | Red-5 | 308 | General-purpose output | No. 54 | 0: General-purpose output<br>2: Axis 4 servo ON output<br>(Main application Ver.0.44 or later) |
| 43 | Orange-5 | 309 | General-purpose output | No. 55 | |
| 44 | Yellow-5 | 310 | General-purpose output | No. 56 | |
| 45 | Green-5 | 311 | General-purpose output | No. 57 | |
| 46 | Blue-5 | 312 | General-purpose output | No. 58 | |
| 47 | Purple-5 | 313 | General-purpose output | No. 59 | 0: General-purpose output<br>1: System-memory backup battery voltage low alarm level or lower |
| 48 | Gray-5 | 314 | General-purpose output | No. 60 | 0: General-purpose output<br>1: Absolute-battery backup battery voltage low alarm level or lower<br>(OR check of all axes. If an error level is detected, this output is retained until power-ON reset or software reset.)<br>(Main application Ver.0.28 or later) |
| 49 | White-5 | 315 | General-purpose output | No. 61 | |
| 50 | Black-5 | | K, KX types : Need not be connected.<br>J, JX types : 0V input | | |

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, "Input filtering period".

[2] Virtual I/O Ports

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

XSEL-J/K Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7000 | Always OFF |
| 7001 | Always ON |
| 7002 | Voltage low warning for system-memory backup battery |
| 7003 | Abnormal voltage of system-memory backup battery |
| 7004 | (For future expansion = Use strictly prohibited) |
| 7005 | (For future expansion = Use strictly prohibited) |
| 7006 | Top-level system error = Message level error is present |
| 7007 | Top-level system error = Operation-cancellation level error is present |
| 7008 | Top-level system error = Cold-start level error is present |
| 7009 | (For future expansion = Use strictly prohibited) |
| 7010 | Drive-source cutoff factor is present (including when waiting for cutoff reset input) |
| 7011 | Latch signal indicating that all-operation-cancellation factor is present (latch signal for recognizing 1-shot cancellation factor; latch is cancelled by 7300-ON) |
| 7012 | All-operation-pause factor is present (including when waiting for restart switch signal) (Valid only during automatic operation recognition) |
| 7013 | All-servo-axis-interlock factor is present (all-operation-pause factor + interlock input-port factor) |
| 7014 | (For future expansion = Use strictly prohibited) |
| 7015 | Voltage low warning for axis-1 absolute-data backup battery (main application version 0.28 or later) |
| 7016 | Abnormal voltage of axis-1 absolute-data backup battery (latched until power-on reset or software reset) (main application version 0.28 or later) |
| 7017 | Voltage low warning for axis-2 absolute-data backup battery (main application version 0.28 or later) |
| 7018 | Abnormal voltage of axis-2 absolute-data backup battery (latched until power-on reset or software reset) (main application version 0.28 or later) |
| 7019 | Voltage low warning for axis-3 absolute-data backup battery (main application version 0.28 or later) |
| 7020 | Abnormal voltage of axis-3 absolute-data backup battery (latched until power-on reset or software reset) (main application version 0.28 or later) |
| 7021 | Voltage low warning for axis-4 absolute-data backup battery (main application version 0.28 or later) |
| 7022 | Abnormal voltage of axis-4 absolute-data backup battery (latched until power-on reset or software reset) (main application version 0.28 or later) |
| 7023 to 7030 | For future expansion = Use strictly prohibited |
| 7031 | Reading SIO CH1 (standard SIO) (reception ready) (*OFF if used for PC/TP connection) (main application versions 0.41 or later) |
| 7032 | Reading SIO CH2 (expanded SIO) (reception ready) (main application versions 0.41 or later) |
| 7033 | Reading SIO CH3 (expanded SIO) (reception ready) (main application versions 0.41 or later) |
| 7034 | Reading SIO CH4 (expanded SIO) (reception ready) (main application versions 0.41 or later) |
| 7035 | Reading SIO CH5 (expanded SIO) (reception ready) (main application versions 0.41 or later) |
| 7036 | Reading SIO CH6 (expanded SIO) (reception ready) (main application versions 0.41 or later) |
| 7037 | Reading SIO CH7 (expanded SIO) (reception ready) (main application versions 0.41 or later) |
| 7038 to 7040 | (For future expansion = Use strictly prohibited) |
| 7041 to 7070 | (For future expansion = Use strictly prohibited) |
| 7071 | In AUTO mode (main application version 0.87 or later) |
| 7072 | During automatic operation (main application version 0.87 or later) |
| 7073 to 7100 | (For future expansion = Use strictly prohibited) |
| 7101 | Running program No. 01 (including during pause) |
| ~ | ~ |
| 7164 | Running program No. 64 (including during pause) |
| 7165 to 7299 | (For future expansion = Use strictly prohibited) |

XSEL-J/K Virtual Output Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7300 | Latch cancellation output for a latch signal indicating that all-operation-cancellation factor is present (7011) (latch is cancelled only when operation-cancellation factor is no longer present) (7300 will be turned OFF following an attempt to cancel latch.) |
| 7301 to 7380 | (For future expansion = Use strictly prohibited) |
| 7381 to 7399 | (For future expansion = Use strictly prohibited) |
| 7400 to 7599 | (For future expansion = Use strictly prohibited) |

XSEL-JX/KX Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7000 | Always OFF |
| 7001 | Always ON |
| 7002 | System-memory backup battery voltage low warning |
| 7003 | System-memory backup battery voltage error |
| 7004 | (Reserved by the system = Use is strictly prohibited) |
| 7005 | (Reserved by the system = Use is strictly prohibited) |
| 7006 | Critical system error = A message level error is present. |
| 7007 | Critical system error = An operation-cancellation level error is present. |
| 7008 | Critical system error = A cold-start level error is present. |
| 7009 | (Reserved by the system = Use is strictly prohibited) |
| 7010 | A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input). |
| 7011 | A latch signal indicating that a cause of all-operation cancellation is present.<br>(This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON) |
| 7012 | A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed).<br>(Effective only in the auto operation recognition mode) |
| 7013 | A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port) |
| 7014 | (Reserved by the system = Use is strictly prohibited) |
| 7015 | Axis 1 absolute-data backup battery voltage low warning |
| 7016 | Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7017 | Axis 2 absolute-data backup battery voltage low warning |
| 7018 | Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7019 | Axis 3 absolute-data backup battery voltage low warning |
| 7020 | Axis 3 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7021 | Axis 4 absolute-data backup battery voltage low warning |
| 7022 | Axis 4 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7023 to 7030 | (For future expansion = Use is strictly prohibited) |
| 7031 | Reading SIO CH1 (standard SIO) (Receive ready) (*OFF if a PC/TP is connected) |
| 7032 | Reading SIO CH2 (standard SIO) (Receive ready) |
| 7033 | Reading SIO CH3 (standard SIO) (Receive ready) |
| 7034 | Reading SIO CH4 (standard SIO) (Receive ready) |
| 7035 | Reading SIO CH5 (standard SIO) (Receive ready) |
| 7036 | Reading SIO CH6 (standard SIO) (Receive ready) |
| 7037 | Reading SIO CH7 (standard SIO) (Receive ready) |
| 7038 to 7070 | (Reserved by the system = Use is strictly prohibited) |
| 7071 | In the AUTO mode (Main application Ver.0.34 or later) |
| 7072 | During auto operation (Main application Ver.0.34 or later) |
| 7073 to 7100 | (Reserved by the system = Use is strictly prohibited) |
| 7101 | Program No. 01 is being executed (or paused). |
| ~ | ~ |
| 7164 | Program No. 64 is being executed (or paused). |
| 7165 to 7299 | (For future expansion = Use is strictly prohibited) |

XSEL-JX/KX Virtual Output Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7300 | A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.)<br>(7300 is turned OFF after latch cancellation is attempted.) |
| 7301 to 7380 | (For future expansion = Use is strictly prohibited) |
| 7381 to 7399 | (Reserved by the system = Use is strictly prohibited) |
| 7400 to 7599 | (For future expansion = Use is strictly prohibited) |

## 2.1.2 XSEL-P/Q/PCT/QCT Controllers

[1] Input and Output I/O Port

With XSEL-P/Q/PCT/QCT controllers, input and output functions can be assigned to input and output ports as desired.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
You can also use I/O parameters 331 to 346 (output function selections 300 (area 2) to 315 (area 2)) to set output functions and then use I/O parameters 315 to 330 to set the port numbers to assign the respective functions to.

Input

| Pin No. | Wire color | Port No. | Standard (factory-set) function |
|---|---|---|---|
| 1 | Brown-1 | | +24V input |
| 2 | Red-1 | 000 | Program start |
| 3 | Orange-1 | 001 | General-purpose input |
| 4 | Yellow-1 | 002 | General-purpose input |
| 5 | Green-1 | 003 | General-purpose input |
| 6 | Blue-1 | 004 | General-purpose input |
| 7 | Purple-1 | 005 | General-purpose input |
| 8 | Gray-1 | 006 | General-purpose input |
| 9 | White-1 | 007 | Program number specification (MSB) |
| 10 | Black-1 | 008 | Program number specification (bit 2) |
| 11 | Brown-2 | 009 | Program number specification (bit 3) |
| 12 | Red-2 | 010 | Program number specification (bit 4) |
| 13 | Orange-2 | 011 | Program number specification (bit 5) |
| 14 | Yellow-2 | 012 | Program number specification (bit 6) |
| 15 | Green-2 | 013 | Program number specification<br>(LSB: bit 7) |
| 16 | Blue-2 | 014 | General-purpose input |
| 17 | Purple-2 | 015 | General-purpose input |
| 18 | Gray-2 | 016 | General-purpose input |
| 19 | White-2 | 017 | General-purpose input |
| 20 | Black-2 | 018 | General-purpose input |
| 21 | Brown-3 | 019 | General-purpose input |
| 22 | Red-3 | 020 | General-purpose input |
| 23 | Orange-3 | 021 | General-purpose input |
| 24 | Yellow-3 | 022 | General-purpose input |
| 25 | Green-3 | 023 | General-purpose input |
| 26 | Blue-3 | 024 | General-purpose input |
| 27 | Purple-3 | 025 | General-purpose input |
| 28 | Gray-3 | 026 | General-purpose input |
| 29 | White-3 | 027 | General-purpose input |
| 30 | Black-3 | 028 | General-purpose input |
| 31 | Brown-4 | 029 | General-purpose input |
| 32 | Red-4 | 030 | General-purpose input |
| 33 | Orange-4 | 031 | General-purpose input |

| Remarks | | |
|---|---|---|
| Inputs are set as shown in the table prior to the shipment, but you can change these input functions by setting applicable I/O parameters. | | |
| Parameter No. | Parameter name | Function |
| No. 30 | Input function selection 000 | 0: General-purpose input<br>1: Program start (input ports 007 to 013, BCD specification)<br>2: Program start (input ports 007 to 013, binary specification)<br>3: Program start (input ports 008 to 014, BCD specification)<br>4: Program start (input ports 008 to 014, binary specification) |
| No. 31 | Input function selection 001 | 0: General-purpose input<br>1: Soft reset signal |
| No. 32 | Input function selection 002 | 0: General-purpose input<br>1: Servo ON signal |
| No. 33 | Input function selection 003 | 0: General-purpose input<br>1: Auto program start upon power-ON reset or software reset in AUTO mode<br>2: Auto program start signal |
| No. 34 | Input function selection 004 | 0: General-purpose input<br>1: Software interlock of all servo axes (OFF level) |
| No. 35 | Input function selection 005 | 0: General-purpose input<br>1: Operation pause cancellation input (ON edge) |
| No. 36 | Input function selection 006 | 0: General-purpose input<br>1: Operation pause signal (OFF level) |
| No. 37 | Input function selection 007 | 0: General-purpose input<br>1: Program number specification (MSB) |
| No. 38 | Input function selection 008 | 0: General-purpose input<br>1: Program number specification (bit 2) |
| No. 39 | Input function selection 009 | 0: General-purpose input<br>1: Program number specification (bit 3) |
| No. 40 | Input function selection 010 | 0: General-purpose input<br>1: Program number specification (bit 4) |
| No. 41 | Input function selection 011 | 0: General-purpose input<br>1: Program number specification (bit 5) |
| No. 42 | Input function selection 012 | 0: General-purpose input<br>1: Program number specification (bit 6) |
| No. 43 | Input function selection 013 | 0: General-purpose input<br>1: Program number specification (LSB: bit 7) |
| No. 44 | Input function selection 014 | 0: General-purpose input<br>1: Drive-source cutoff cancellation (ON edge) |
| No. 45 | Input function selection 015 | 0: General-purpose input<br>1: Home return of all effective axes (ON edge)<br>2: Home return of all effective incremental axes (ON edge) |

Output

| Pin No. | Wire color | Port No. | Standard Setting (in the delivery) Function | Parameter No. *1 | Parameter Name | Function |
|---|---|---|---|---|---|---|
| | | | | Remarks: When the unit is delivered, the output is set as shown in the table. However, the output function can be changed using the I/O parameter setting. | | |
| 34 | YW-4 | 300 | Error Output at the Operation Cancellation Level or more (OFF) | No.46<br>No.331 | Output Function Selection 300<br>Output Function Selection 300 (Area 2) | 0: Universal Output<br>1: Error Output at the Operation Cancellation Level or more (ON)<br>2: Error Output at the Operation Cancellation Level or more (OFF)<br>3: Error Output at the Operation Cancellation Level or more + Emergency-stop output (ON)<br>4: Error Output at the Operation Cancellation Level or more + Emergency-stop output (OFF) |
| 35 | GN-4 | 301 | READY Output (PIO Trigger Program Operation Available and without occurrence of any error at the cold start level or more) (Main Application Ver. 0.20 or later) | No.47<br>No.332 | Output Function Selection 301<br>Output Function Selection 301 (Area 2) | 0: Universal Input<br>1: READY Output (PIO Trigger Program Operation Available)<br>2: READY Output (PIO Trigger Program Operation Available)and without occurrence of any error at the operation cancellation level or more<br>3: READY Output (PIO Trigger Program Operation Available)and READY Output (PIO Trigger Program Operation Available, and without occurrence of any error at the cold start level or more or more level or more |
| 36 | BL-4 | 302 | Emergency-stop output (OFF) | No.48<br>No.333 | Output Function Selection 302<br>Output Function Selection 302 (Area 2) | 0: Universal Input<br>2: Emergency-stop output (ON)<br>3: Emergency-stop output (OFF) |
| 37 | PL-4 | 303 | Universal Output | No.49<br>No.334 | Output Function Selection 303<br>Output Function Selection 303 (Area 2) | 0: Universal Output<br>1: AUTO Mode Output<br>2: Output during the Automatic Operation (In addition, when the parameter No. 12 is set to “1”) |
| 38 | GY-4 | 304 | Universal Output | No.50<br>No.335 | Output Function Selection 304<br>Output Function Selection 304 (Area 2) | 0: Universal Output<br>1: Output at the time of “All Effective Axes Homing (=0)”<br>2: Output when all the effective axes homing is completed<br>3: Output when all the effective axes home preset coordinates are set<br>* When the actuator applicable to the absolute encoder is moved to the coordinates “0” or home preset coordinates, use “MOVE” order, not “HOME” order. |
| 39 | WT-4 | 305 | Universal Output | No.51<br>No.336 | Output Function Selection 305<br>Output Function Selection 305 (Area 2) | 0: Universal Output<br>1: Axis 1 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 1 servo ON |
| 40 | BK-4 | 306 | Universal Output | No.52<br>No.337 | Output Function Selection 306<br>Output Function Selection 306 (Area 2) | 0: Universal Output<br>1: Axis 2 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 2 servo ON |
| 41 | BR-5 | 307 | Universal Output | No.53<br>No.338 | Output Function Selection 307<br>Output Function Selection 307 (Area 2) | 0: Universal Output<br>1: Axis 3 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 3 servo ON |
| 42 | RD-5 | 308 | Universal Output | No.54<br>No.339 | Output Function Selection 308<br>Output Function Selection 308 (Area 2) | 0: Universal Output<br>1: Axis 4 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 4 servo ON |
| 43 | OR-5 | 309 | Universal Output | No.55<br>No.340 | Output Function Selection 309<br>Output Function Selection 309 (Area 2) | 0: Universal Output<br>1: Axis 5 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 5 servo ON |
| 44 | YW-5 | 310 | Universal Output | No.56<br>No.341 | Output Function Selection 310<br>Output Function Selection 310 (Area 2) | 0: Universal Output<br>1: Axis 6 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 6 servo ON |
| 45 | GN-5 | 311 | Universal Output | No.57<br>No.342 | Output Function Selection 311<br>Output Function Selection 311 (Area 2) | |
| 46 | BL-5 | 312 | Universal Output | No.58<br>No.343 | Output Function Selection 312<br>Output Function Selection 312 (Area 2) | |

| Pin No. | Wire color | Port No. | Standard Setting (in the delivery) Function |
|---|---|---|---|
| 47 | PL-5 | 313 | Universal Output |
| 48 | GY-5 | 314 | Universal Output |
| 49 | WT-5 | 315 | Universal Output |
| 50 | BK-5 | | 0V Output |

| Remarks | | |
|---|---|---|
| When the unit is delivered, the output is set as shown in the table.<br>However, the output function can be changed using the I/O parameter setting. | | |
| Parameter No. *1 | Parameter Name | Function |
| No.59<br>No.344 | Output Function Selection 313<br>Output Function Selection 313 (Area 2) | 0: Universal Output<br>1: System Memory Backup Battery Low Voltage Alarm Level or less |
| No.60<br>No.345 | Output Function Selection 314<br>Output Function Selection 314 (Area 2) | 0: Universal Output<br>1: Absolute Battery Backup Battery Low Voltage Alarm Level or less (All axes OR check: Error level detection is maintained until power ON reset and software reset) |
| No.61<br>No.346 | Output Function Selection 315<br>Output Function Selection 315 (Area 2) | |

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.

[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

XSEL-P/Q/PCT/QCT Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7000 | Always OFF |
| 7001 | Always ON |
| 7002 | System-memory backup battery voltage low warning |
| 7003 | System-memory backup battery voltage error |
| 7004 | (Reserved by the system = Use is strictly prohibited) |
| 7005 | (Reserved by the system = Use is strictly prohibited) |
| 7006 | Critical system error = A message level error is present. |
| 7007 | Critical system error = An operation-cancellation level error is present. |
| 7008 | Critical system error = A cold-start level error is present. |
| 7009 | (Reserved by the system = Use is strictly prohibited) |
| 7010 | A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input). |
| 7011 | A latch signal indicating that a cause of all-operation cancellation is present.<br>(This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON) |
| 7012 | A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed).<br>(Effective only in the auto operation recognition mode) |
| 7013 | A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port) |
| 7014 | (Reserved by the system = Use is strictly prohibited) |
| 7015 | Axis 1 absolute-data backup battery voltage low warning |
| 7016 | Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7017 | Axis 2 absolute-data backup battery voltage low warning (Main application version 0.28 or later) |
| 7018 | Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7019 | Axis 3 absolute-data backup battery voltage low warning |
| 7020 | Axis 3 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7021 | Axis 4 absolute-data backup battery voltage low warning |
| 7022 | Axis 4 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7023 | Axis 5 absolute-data backup battery voltage low warning (Effective only with 6-axis types) |
| 7024 | Axis 5 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)<br>(Effective only with 6-axis types) |
| 7025 | Axis 6 absolute-data backup battery voltage low warning (Effective only with 6-axis types) |
| 7026 | Axis 6 absolute-data backup battery voltage error (Latched until power-ON reset or software reset)<br>(Effective only with 6-axis types) |
| 7027 to 7040 | (Reserved by the system = Use is strictly prohibited) |
| 7041, 7042 | (For future expansion = Use is strictly prohibited) |
| 7043 | Axis 1 home return completion |
| 7044 | Axis 2 home return completion |
| 7045 | Axis 3 home return completion |
| 7046 | Axis 4 home return completion |
| 7047 | Axis 5 home return completion |
| 7048 | Axis 6 home return completion |
| 7049 to 7070 | (For future expansion = Use is strictly prohibited) |
| 7071 | In the AUTO mode |
| 7072 | During auto operation |
| 7073 to 7100 | (Reserved by the system = Use is strictly prohibited) |
| 7101 | Program No. 01 is being executed (or paused). |
| ~ | ~ |
| 7164 | Program No. 64 is being executed (or paused). |

XSEL-P/Q/PCT/QCT Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7165 | Program No. 65 is being executed (or paused).<br>(Controller with increased memory capacity (with gateway function) only) |
| ~ | ~ |
| 7228 | Program No. 128 is being executed (or paused).<br>(Controller with increased memory capacity (with gateway function) only) |
| 7229 to 7299 | (For future expansion = Use is strictly prohibited) |

XSEL-P/Q/PCT/QCT Virtual Output Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7300 | A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.)<br>(7300 is turned OFF after latch cancellation is attempted.) |
| 7301 | Axis 1 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 “Brake Output Control Method Select” = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 0.41 or later) |
| 7302 | Axis 2 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 “Brake Output Control Method Select” = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 0.41 or later) |
| 7303 | Axis 3 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 “Brake Output Control Method Select” = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 0.41 or later) |
| 7304 | Axis 4 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 “Brake Output Control Method Select” = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 0.41 or later) |
| 7305 | Axis 5 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 “Brake Output Control Method Select” = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 0.41 or later) |
| 7306 | Axis 6 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 “Brake Output Control Method Select” = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 0.41 or later) |
| 7307 to 7380 | (For future expansion = Use is strictly prohibited) |
| 7381 to 7399 | (Reserved by the system = Use is strictly prohibited) |
| 7400 to 7599 | (For future expansion = Use is strictly prohibited) |

## 2.1.3 XSEL-PX/QX Controllers

[1] Input and Output I/O Port

With XSEL-PX/QX type controllers, the assignments of input and output functions to I/O ports are fixed and cannot be changed.

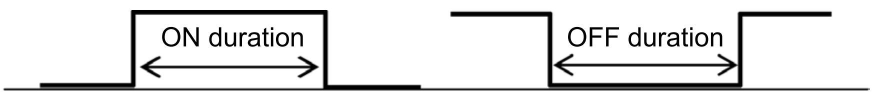
Input

| Pin No. | Wire color | Port No. | Standard (factory-set) function | I/O parameter | |
|---|---|---|---|---|---|
| | | | | Inputs are set as shown in the table prior to the shipment, but you can change these input functions by setting applicable I/O parameters. | |
| 1 | Brown-1 | | +24V input | | |
| 2 | Red-1 | 000 | Program start | No. 30 | 0: General-purpose input<br>1: Program start (input ports 007 to 013, BCD specification)<br>2: Program start (input ports 007 to 013, binary specification)<br>3: Program start (input ports 008 to 014, BCD specification)<br>4: Program start (input ports 008 to 014, binary specification) |
| 3 | Orange-1 | 001 | General-purpose input | No. 31 | 0: General-purpose input<br>1: Soft reset signal |
| 4 | Yellow-1 | 002 | General-purpose input | No. 32 | 0: General-purpose input<br>1: Servo ON signal |
| 5 | Green-1 | 003 | General-purpose input | No. 33 | 0: General-purpose input<br>1: Auto program start upon power-ON reset or software reset in AUTO mode<br>2: Auto program start signal |
| 6 | Blue-1 | 004 | General-purpose input | No. 34 | 0: General-purpose input<br>1: Software interlock of all servo axes (OFF level) |
| 7 | Purple-1 | 005 | General-purpose input | No. 35 | 0: General-purpose input<br>1: Operation pause cancellation input (ON edge) |
| 8 | Gray-1 | 006 | General-purpose input | No. 36 | 0: General-purpose input<br>1: Operation pause signal (OFF level) |
| 9 | White-1 | 007 | Program number specification (MSB) | No. 37 | 0: General-purpose input<br>1: Program number specification (MSB) |
| 10 | Black-1 | 008 | Program number specification (bit 2) | No. 38 | 0: General-purpose input<br>1: Program number specification (bit 2) |
| 11 | Brown-2 | 009 | Program number specification (bit 3) | No. 39 | 0: General-purpose input<br>1: Program number specification (bit 3) |
| 12 | Red-2 | 010 | Program number specification (bit 4) | No. 40 | 0: General-purpose input<br>1: Program number specification (bit 4) |
| 13 | Orange-2 | 011 | Program number specification (bit 5) | No. 41 | 0: General-purpose input<br>1: Program number specification (bit 5) |
| 14 | Yellow-2 | 012 | Program number specification (bit 6) | No. 42 | 0: General-purpose input<br>1: Program number specification (bit 6) |
| 15 | Green-2 | 013 | Program number specification (LSB: bit 7) | No. 43 | 0: General-purpose input<br>1: Program number specification (LSB: bit 7) |
| 16 | Blue-2 | 014 | General-purpose input | No. 44 | 0: General-purpose input<br>1: Drive-source cutoff cancellation (ON edge) |
| 17 | Purple-2 | 015 | General-purpose input | No. 45 | 0: General-purpose input<br>1: Home return of all effective axes (ON edge)<br>2: Home return of all effective incremental axes (ON edge) |
| 18 | Gray-2 | 016 | General-purpose input | | |
| 19 | White-2 | 017 | General-purpose input | | |
| 20 | Black-2 | 018 | General-purpose input | | |
| 21 | Brown-3 | 019 | General-purpose input | | |
| 22 | Red-3 | 020 | General-purpose input | | |
| 23 | Orange-3 | 021 | General-purpose input | | |
| 24 | Yellow-3 | 022 | General-purpose input | | |
| 25 | Green-3 | 023 | General-purpose input | | |
| 26 | Blue-3 | 024 | General-purpose input | | |
| 27 | Purple-3 | 025 | General-purpose input | | |
| 28 | Gray-3 | 026 | General-purpose input | | |
| 29 | White-3 | 027 | General-purpose input | | |
| 30 | Black-3 | 028 | General-purpose input | | |
| 31 | Brown-4 | 029 | General-purpose input | | |
| 32 | Red-4 | 030 | General-purpose input | | |
| 33 | Orange-4 | 031 | General-purpose input | | |

Output

| Pin No. | Wire color | Port No. | Standard (factory-set) function | | |
|---|---|---|---|---|---|
| 34 | Yellow-4 | 300 | Output of operation-cancellation level or higher error (OFF) | No. 46 | 0: General-purpose output<br>1: Output of operation-cancellation level or higher error (ON)<br>2: Output of operation-cancellation level or higher error (OFF)<br>3: Output of operation-cancellation level or higher error + Emergency stop output (ON)<br>4: Output of operation-cancellation level or higher error + Emergency stop output (OFF) |
| 35 | Green-4 | 301 | READY output (PIO-trigger program operation enabled AND no cold-start level or higher error) (Main application Ver.0.20 or later) | No. 47 | 0: General-purpose output<br>1: READY output (PIO-trigger program operation enabled)<br>2: READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error)<br>3: READY output (PIO-trigger program operation enabled AND no cold-start level or higher error) |
| 36 | Blue-4 | 302 | Emergency stop output (OFF) | No. 48 | 0: General-purpose output<br>2: Emergency stop output (ON)<br>3: Emergency stop output (OFF) |
| 37 | Purple-4 | 303 | General-purpose output | No. 49 | 0: General-purpose output<br>1: AUTO mode output<br>2: Auto operation output (When other parameter No. 12 is set to ‘1’) |
| 38 | Gray-4 | 304 | General-purpose output | No. 50 | 0: General-purpose output<br>The following settings are effective only with XSEL-J/K:<br>1: Output when all effective axes are home (= 0)<br>2: Output when all effective axes have completed home return<br>3: Output when all effective axes are at home preset coordinate<br>* To move an actuator of absolute encoder specification to coordinate 0 or the home preset coordinate, use a MOVP command instead of HOME command. |
| 39 | White-4 | 305 | General-purpose output | No. 51 | 0: General-purpose output<br>2: Axis 1 servo ON output |
| 40 | Black-4 | 306 | General-purpose output | No. 52 | 0: General-purpose output<br>2: Axis 2 servo ON output |
| 41 | Brown-5 | 307 | General-purpose output | No. 53 | 0: General-purpose output<br>2: Axis 3 servo ON output |
| 42 | Red-5 | 308 | General-purpose output | No. 54 | 0: General-purpose output<br>2: Axis 4 servo ON output |
| 43 | Orange-5 | 309 | General-purpose output | No. 55 | 0: General-purpose output<br>2: Axis 5 servo ON output |
| 44 | Yellow-5 | 310 | General-purpose output | No. 56 | 0: General-purpose output<br>2: Axis 6 servo ON output |
| 45 | Green-5 | 311 | General-purpose output | No. 57 | |
| 46 | Blue-5 | 312 | General-purpose output | No. 58 | |
| 47 | Purple-5 | 313 | General-purpose output | No. 59 | 0: General-purpose output<br>1: System-memory backup battery voltage low alarm level or lower |
| 48 | Gray-5 | 314 | General-purpose output | No. 60 | 0: General-purpose output<br>1: Absolute-battery backup battery voltage low alarm level or lower<br>(OR check of all axes. If an error level is detected, this output is retained until power-ON reset or software reset.) |
| 49 | White-5 | 315 | General-purpose output | No. 61 | |
| 50 | Black-5 | | 0V input | | |

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.



[2] Virtual I/O Port
Should be the same as XSEL-P/Q/PCT/QCT [Refer to 2.1.2 XSEL-P/Q/PCT/QCT]

XSEL-PX/QX Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7000 | Always OFF |
| 7001 | Always ON |
| 7002 | Voltage low warning for system memory backup battery |
| 7003 | Abnormal voltage of system memory backup battery |
| 7004 | (For future expansion = Use strictly prohibited) |
| 7005 | (For future expansion = Use strictly prohibited) |
| 7006 | Top level system error = Message level error is present |
| 7007 | Top level system error = Operation cancellation level error is present |
| 7008 | Top level system error = Cold start level error is present |
| 7009 | (For future expansion = Use strictly prohibited) |
| 7010 | Drive source cutoff factor is present (including when waiting for cutoff reset input) |
| 7011 | Latch signal indicating that all operation cancellation factor is present (latch signal for recognizing 1-shot cancellation factor; latch is cancelled by 7300 being ON) |
| 7012 | All operation pause factor is present (including when waiting for restart switch signal. Valid only during automatic operation recognition) |
| 7013 | All servo axis interlock factor is present (all operation pause factor + interlock input port factor) |
| 7014 | (For future expansion = Use strictly prohibited) |
| 7015 | Voltage low warning for axis 1 absolute data backup battery |
| 7016 | Abnormal voltage of axis 1 absolute data backup battery (latched until power on reset or software reset) |
| 7017 | Voltage low warning for axis 2 absolute data backup battery (main application version 0.28 or later) |
| 7018 | Abnormal voltage of axis 2 absolute data backup battery (latched until power on reset or software reset) |
| 7019 | Voltage low warning for axis 3 absolute data backup battery |
| 7020 | Abnormal voltage of axis 3 absolute data backup battery (latched until power on reset or software reset) |
| 7021 | Voltage low warning for axis 4 absolute data backup battery |
| 7022 | Abnormal voltage of axis 4 absolute data backup battery (latched until power on reset or software reset) |
| 7023 | Voltage low warning for axis 5 absolute data backup battery (valid only when the controller supports up to 6 axes) |
| 7024 | Abnormal voltage of axis 5 absolute data backup battery (latched until power on reset or software reset. Valid only when the controller supports up to 6 axes) |
| 7025 | Voltage low warning for axis 6 absolute data backup battery (valid only when the controller supports up to 6 axes) |
| 7026 | Abnormal voltage of axis 6 absolute data backup battery (latched until power on reset or software reset. Valid only when the controller supports up to 6 axes) |
| 7027 to 7040 | (For future expansion = Use strictly prohibited) |
| 7041 to 7070 | (For future expansion = Use strictly prohibited) |
| 7071 | In AUTO mode |
| 7072 | During automatic operation |
| 7073 to 7100 | (For future expansion = Use strictly prohibited) |
| 7101 | Running program No. 01 (including during pause) |
| ~ | ~ |
| 7164 | Running program No. 64 (including during pause) |
| 7165 | Running program No. 65 (including during pause)<br>(Controller with increased memory size (with gateway function) only) |
| ~ | ~ |
| 7228 | Running program No. 128 (including during pause)<br>(Controller with increased memory size (with gateway function) only) |
| 7229 to 7299 | (For future expansion = Use strictly prohibited) |

XSEL-PX/QX Virtual Output Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7300 | Latch cancellation output for a latch signal indicating that all operation cancellation factor is present (port 7011. The latch is cancelled only when operation cancellation factor is no longer present. 7300 will be turned OFF following an attempt to cancel latch) |
| 7301 to 7380 | (For future expansion = Use strictly prohibited) |
| 7381 to 7399 | (For future expansion = Use strictly prohibited) |
| 7400 to 7599 | (For future expansion = Use strictly prohibited) |

### 2.1.4 XSEL-R/S/RX/SX/RXD/SXD Controllers

[1] Input and Output I/O Port

With XSEL-R/S/RX/SX/RXD/SXD controllers, input and output functions can be assigned to input and output ports as desired.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
You can also use I/O parameters 331 to 346 (output function selections 300 (area 2) to 315 (area 2)) to set output functions and then use I/O parameters 315 to 330 to set the port numbers to assign the respective functions to.

Input

| Pin No. | Wire color | Port No. | Standard (factory-set) function |
|---|---|---|---|
| 1 | Brown-1 | | +24V input |
| 2 | Red-1 | 000 | Program start |
| 3 | Orange-1 | 001 | General-purpose input |
| 4 | Yellow-1 | 002 | General-purpose input |
| 5 | Green-1 | 003 | General-purpose input |
| 6 | Blue-1 | 004 | General-purpose input |
| 7 | Purple-1 | 005 | General-purpose input |
| 8 | Gray-1 | 006 | General-purpose input |
| 9 | White-1 | 007 | Program number specification (MSB) |
| 10 | Black-1 | 008 | Program number specification (bit 2) |
| 11 | Brown-2 | 009 | Program number specification (bit 3) |
| 12 | Red-2 | 010 | Program number specification (bit 4) |
| 13 | Orange-2 | 011 | Program number specification (bit 5) |
| 14 | Yellow-2 | 012 | Program number specification (bit 6) |
| 15 | Green-2 | 013 | Program number specification (LSB: bit 7) |
| 16 | Blue-2 | 014 | General-purpose input |
| 17 | Purple-2 | 015 | General-purpose input |
| 18 | Gray-2 | 016 | General-purpose input |
| 19 | White-2 | 017 | General-purpose input |
| 20 | Black-2 | 018 | General-purpose input |
| 21 | Brown-3 | 019 | General-purpose input |
| 22 | Red-3 | 020 | General-purpose input |
| 23 | Orange-3 | 021 | General-purpose input |
| 24 | Yellow-3 | 022 | General-purpose input |
| 25 | Green-3 | 023 | General-purpose input |
| 26 | Blue-3 | 024 | General-purpose input |
| 27 | Purple-3 | 025 | General-purpose input |
| 28 | Gray-3 | 026 | General-purpose input |
| 29 | White-3 | 027 | General-purpose input |
| 30 | Black-3 | 028 | General-purpose input |
| 31 | Brown-4 | 029 | General-purpose input |
| 32 | Red-4 | 030 | General-purpose input |
| 33 | Orange-4 | 031 | General-purpose input |

Remarks

Inputs are set as shown in the table prior to the shipment, but you can change these input functions by setting applicable I/O parameters.

| Parameter No. | Parameter name | Function |
|---|---|---|
| No. 30 | Input function selection 000 | 0: General-purpose input<br>1: Program start (input ports 007 to 013, BCD specification)<br>2: Program start (input ports 007 to 013, binary specification)<br>3: Program start (input ports 008 to 014, BCD specification)<br>4: Program start (input ports 008 to 014, binary specification) |
| No. 31 | Input function selection 001 | 0: General-purpose input<br>1: Soft reset signal |
| No. 32 | Input function selection 002 | 0: General-purpose input<br>1: Servo ON signal |
| No. 33 | Input function selection 003 | 0: General-purpose input<br>1: Auto program start upon power-ON reset or software reset in AUTO mode<br>2: Auto program start signal |
| No. 34 | Input function selection 004 | 0: General-purpose input<br>1: Software interlock of all servo axes (OFF level) |
| No. 35 | Input function selection 005 | 0: General-purpose input<br>1: Operation pause cancellation input (ON edge) |
| No. 36 | Input function selection 006 | 0: General-purpose input<br>1: Operation pause signal (OFF level) |
| No. 37 | Input function selection 007 | 0: General-purpose input<br>1: Program number specification (MSB) |
| No. 38 | Input function selection 008 | 0: General-purpose input<br>1: Program number specification (bit 2) |
| No. 39 | Input function selection 009 | 0: General-purpose input<br>1: Program number specification (bit 3) |
| No. 40 | Input function selection 010 | 0: General-purpose input<br>1: Program number specification (bit 4) |
| No. 41 | Input function selection 011 | 0: General-purpose input<br>1: Program number specification (bit 5) |
| No. 42 | Input function selection 012 | 0: General-purpose input<br>1: Program number specification (bit 6) |
| No. 43 | Input function selection 013 | 0: General-purpose input<br>1: Program number specification (LSB: bit 7) |
| No. 44 | Input function selection 014 | 0: General-purpose input<br>1: Drive-source cutoff cancellation (ON edge) |
| No. 45 | Input function selection 015 | 0: General-purpose input<br>1: Home return of all effective axes (ON edge)<br>2: Home return of all effective incremental axes (ON edge) |

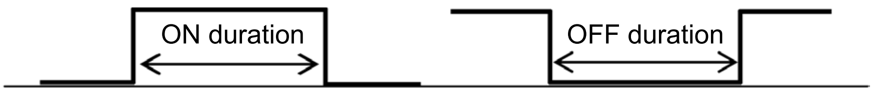
Output

| Pin No. | Wire color | Port No. | Standard Setting (in the delivery) Function | Parameter No. *1 | Parameter Name | Function |
|---|---|---|---|---|---|---|
| | | | | Remarks<br>When the unit is delivered, the output is set as shown in the table.<br>However, the output function can be changed using the I/O parameter setting. | | |
| 34 | YW-4 | 300 | Error Output at the Operation Cancellation Level or more (OFF) | No.46<br>No.331 | Output Function Selection 300<br>Output Function Selection 300 (Area 2) | 0: Universal Output<br>1: Error Output at the Operation Cancellation Level or more (ON)<br>2: Error Output at the Operation Cancellation Level or more (OFF)<br>3: Error Output at the Operation Cancellation Level or more + Emergency-stop output (ON)<br>4: Error Output at the Operation Cancellation Level or more + Emergency-stop output (OFF)<br>5: Error output of cold start level or more (ON)<br>6: Error output of cold start level or more (OFF)<br>7: Message level related to maintenance information alarm function (OFF) for error output of (Error No. 231 to 232) |
| 35 | GN-4 | 301 | READY Output (PIO Trigger Program Operation Available and without occurrence of any error at the cold start level or more) (Main Application Ver. 0.20 or later) | No.47<br>No.332 | Output Function Selection 301<br>Output Function Selection 301 (Area 2) | 0: Universal Input<br>1: READY Output (PIO Trigger Program Operation Available)<br>2: READY Output (PIO Trigger Program Operation Available)and without occurrence of any error at the operation cancellation level or more<br>3: READY Output (PIO Trigger Program Operation Available)and READY Output (PIO Trigger Program Operation Available, and without occurrence of any error at the cold start level or more or more level or more |
| 36 | BL-4 | 302 | Emergency-stop output (OFF) | No.48<br>No.333 | Output Function Selection 302<br>Output Function Selection 302 (Area 2) | 0: Universal Input<br>2: Emergency-stop output (ON)<br>3: Emergency-stop output (OFF) |
| 37 | PL-4 | 303 | Universal Output | No.49<br>No.334 | Output Function Selection 303<br>Output Function Selection 303 (Area 2) | 0: Universal Output<br>1: AUTO Mode Output<br>2: Output during the Automatic Operation (In addition, when the parameter No. 12 is set to "1") |
| 38 | GY-4 | 304 | Universal Output | No.50<br>No.335 | Output Function Selection 304<br>Output Function Selection 304 (Area 2) | 0: Universal Output<br>1: Output at the time of "All Effective Axes Homing (=0)"<br>2: Output when all the effective axes homing is completed (Coordinates determined)<br>3: Output when all the effective axes home preset coordinates are set |
| 39 | WT-4 | 305 | Universal Output | No.51<br>No.336 | Output Function Selection 305<br>Output Function Selection 305 (Area 2) | 0: Universal Output<br>1: Axis 1 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 1 servo ON |
| 40 | BK-4 | 306 | Universal Output | No.52<br>No.337 | Output Function Selection 306<br>Output Function Selection 306 (Area 2) | 0: Universal Output<br>1: Axis 2 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 2 servo ON |
| 41 | BR-5 | 307 | Universal Output | No.53<br>No.338 | Output Function Selection 307<br>Output Function Selection 307 (Area 2) | 0: Universal Output<br>1: Axis 3 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 3 servo ON |
| 42 | RD-5 | 308 | Universal Output | No.54<br>No.339 | Output Function Selection 308<br>Output Function Selection 308 (Area 2) | 0: Universal Output<br>1: Axis 4 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 4 servo ON |
| 43 | OR-5 | 309 | Universal Output | No.55<br>No.340 | Output Function Selection 309<br>Output Function Selection 309 (Area 2) | 0: Universal Output<br>1: Axis 5 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 5 servo ON |
| 44 | YW-5 | 310 | Universal Output | No.56<br>No.341 | Output Function Selection 310<br>Output Function Selection 310 (Area 2) | 0: Universal Output<br>1: Axis 6 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 6 servo ON |
| 45 | GN-5 | 311 | Universal Output | No.57<br>No.342 | Output Function Selection 311<br>Output Function Selection 311 (Area 2) | 0: Universal Output<br>1: Axis 7 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 7 servo ON (system monitoring task output) |
| 46 | BL-5 | 312 | Universal Output | No.58<br>No.343 | Output Function Selection 312<br>Output Function Selection 312 (Area 2) | 0: Universal Output<br>1: Axis 8 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 8 servo ON (system monitoring task output) |

| Pin No. | Wire color | Port No. | Standard Setting (in the delivery) Function |
|---|---|---|---|
| 47 | PL-5 | 313 | Universal Output |
| 48 | GY-5 | 314 | Universal Output |
| 49 | WT-5 | 315 | Universal Output |
| 50 | BK-5 | | 0V Output |

| Remarks | | |
|---|---|---|
| When the unit is delivered, the output is set as shown in the table.<br>However, the output function can be changed using the I/O parameter setting. | | |
| Parameter No. *1 | Parameter Name | Function |
| No.59<br>No.344 | Output Function Selection 313<br>Output Function Selection 313 (Area 2) | 0: Universal Output<br>1: Reserved by the system |
| No.60<br>No.345 | Output Function Selection 314<br>Output Function Selection 314 (Area 2) | 0: Universal Output<br>1: Voltage low warning for absolute-data backup battery |
| No.61<br>No.346 | Output Function Selection 315<br>Output Function Selection 315 (Area 2) | 0: Universal Output |

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, “Input filtering period”.



[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

XSEL-R/S/RX/SX/RXD/SXD Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7000 | Always OFF |
| 7001 | Always ON |
| 7002 to 7005 | (Reserved by the system = Use is strictly prohibited) |
| 7006 | Critical system error = A message level error is present. |
| 7007 | Critical system error = An operation-cancellation level error is present. |
| 7008 | Critical system error = A cold-start level error is present. |
| 7009 | (Reserved by the system = Use is strictly prohibited) |
| 7010 | A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input). |
| 7011 | A latch signal indicating that a cause of all-operation cancellation is present.<br>(This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON) |
| 7012 | A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed).<br>(Effective only in the auto operation recognition mode) |
| 7013 | A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port) |
| 7014 | (Reserved by the system = Use is strictly prohibited) |
| 7015 | Axis 1 absolute-data backup battery voltage low warning |
| 7016 | Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7017 | Axis 2 absolute-data backup battery voltage low warning |
| 7018 | Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7019 | Axis 3 absolute-data backup battery voltage low warning |
| 7020 | Axis 3 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7021 | Axis 4 absolute-data backup battery voltage low warning |
| 7022 | Axis 4 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7023 | Axis 5 absolute-data backup battery voltage low warning |
| 7024 | Axis 5 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7025 | Axis 6 absolute-data backup battery voltage low warning |
| 7025 | Axis 6 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7027 | Axis 7 absolute-data backup battery voltage low warning |
| 7028 | Axis 7 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7029 | Axis 8 absolute-data backup battery voltage low warning |
| 7030 | Axis 8 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7031 to 7040 | (Reserved by the system = Use is strictly prohibited) |
| 7041 to 7042 | (For future expansion = Use is strictly prohibited) |
| 7043 | Axis 1 home return completion |
| 7044 | Axis 2 home return completion |
| ~ | ~ |
| 7048 | Axis 6 home return completion |
| 7049 | Axis 7 home return completion |
| 7050 | Axis 8 home return completion |
| 7051 to 7069 | (For future expansion = Use is strictly prohibited) |
| 7070 | (Reserved by the system = Use is strictly prohibited) |
| 7071 | In the AUTO mode |
| 7072 | During auto operation |
| 7073 to 7074 | (For future expansion = Use is strictly prohibited) |
| 7075 | During Tracking Conveyor Speed Drop Detection |

XSEL-R/S/RX/SX/RXD/SXD Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7076 | Tracking Operation Complete Work Position Arrival Latch Signal (Latches until “TRAC 1 nnn” or “Tracking Operation Execution Program Finish”) |
| 7077 | In Tracking Conveyor Tracking Complete Range |
| 7078 | Tracking Reversed Operation Detected Work Position Arrival Latch Signal (Latches until “TRAC 1 nnn” or “Tracking Operation Execution Program Finish”) |
| 7079 | During Tracking Mode (Work detection valid) (for SCARA controller only) |
| 7080 | During Tracking Operation (including a pause in tracking operation) |
| 7081 to 7100 | (Reserved by the system = Use is strictly prohibited) |
| 7101 | Program No. 01 is being executed (or paused). |
| 7102 | Program No. 02 is being executed (or paused). |
| 7103 | Program No. 03 is being executed (or paused). |
| ~ | ~ |
| 7227 | Program No. 127 is being executed (or paused). |
| 7228 | Program No. 128 is being executed (or paused). |
| 7229 to 7299 | (Reserved by the system = Use is strictly prohibited) |

XSEL-R/S/RX/SX/RXD/SXD Virtual Output Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7300 | A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.)<br>(7300 is turned OFF after latch cancellation is attempted.) |
| 7301 to 7380 | (For future expansion = Use is strictly prohibited) |
| 7381 to 7399 | (Reserved by the system = Use is strictly prohibited) |
| 7400 to 7599 | (For future expansion = Use is strictly prohibited) |

## 2.1.5 XSEL-RA/SA/RAX/SAX/RAXD/SAXD Controllers

[1] Input and Output I/O Port

With XSEL-RA/SA/RAX/SAX/RAXD/SAXD controllers, input and output functions can be assigned to input and output ports as desired.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
You can also use I/O parameters 331 to 346 (output function selections 300 (area 2) to 315 (area 2)) to set output functions and then use I/O parameters 315 to 330 to set the port numbers to assign the respective functions to.

Input

| Pin No. | Wire color | Port No. | Standard (factory-set) function | Parameter No. | Parameter name | Function |
|---|---|---|---|---|---|---|
| | | | | Remarks<br>Inputs are set as shown in the table prior to the shipment, but you can change these input functions by setting applicable I/O parameters. | | |
| 1 | Brown-1 | | +24V input | | | |
| 2 | Red-1 | 000 | Program start | No. 30 | Input function selection 000 | 0: General-purpose input<br>1: Program start (input ports 007 to 013, BCD specification)<br>2: Program start (input ports 007 to 013, binary specification)<br>3: Program start (input ports 008 to 014, BCD specification)<br>4: Program start (input ports 008 to 014, binary specification) |
| 3 | Orange-1 | 001 | General-purpose input | No. 31 | Input function selection 001 | 0: General-purpose input<br>1: Soft reset signal |
| 4 | Yellow-1 | 002 | General-purpose input | No. 32 | Input function selection 002 | 0: General-purpose input<br>1: Servo ON signal |
| 5 | Green-1 | 003 | General-purpose input | No. 33 | Input function selection 003 | 0: General-purpose input<br>1: Auto program start upon power-ON reset or software reset in AUTO mode<br>2: Auto program start signal |
| 6 | Blue-1 | 004 | General-purpose input | No. 34 | Input function selection 004 | 0: General-purpose input<br>1: Software interlock of all servo axes (OFF level) |
| 7 | Purple-1 | 005 | General-purpose input | No. 35 | Input function selection 005 | 0: General-purpose input<br>1: Operation pause cancellation input (ON edge) |
| 8 | Gray-1 | 006 | General-purpose input | No. 36 | Input function selection 006 | 0: General-purpose input<br>1: Operation pause signal (OFF level) |
| 9 | White-1 | 007 | Program number specification (MSB) | No. 37 | Input function selection 007 | 0: General-purpose input<br>1: Program number specification (MSB) |
| 10 | Black-1 | 008 | Program number specification (bit 2) | No. 38 | Input function selection 008 | 0: General-purpose input<br>1: Program number specification (bit 2) |
| 11 | Brown-2 | 009 | Program number specification (bit 3) | No. 39 | Input function selection 009 | 0: General-purpose input<br>1: Program number specification (bit 3) |
| 12 | Red-2 | 010 | Program number specification (bit 4) | No. 40 | Input function selection 010 | 0: General-purpose input<br>1: Program number specification (bit 4) |
| 13 | Orange-2 | 011 | Program number specification (bit 5) | No. 41 | Input function selection 011 | 0: General-purpose input<br>1: Program number specification (bit 5) |
| 14 | Yellow-2 | 012 | Program number specification (bit 6) | No. 42 | Input function selection 012 | 0: General-purpose input<br>1: Program number specification (bit 6) |
| 15 | Green-2 | 013 | Program number specification (LSB: bit 7) | No. 43 | Input function selection 013 | 0: General-purpose input<br>1: Program number specification (LSB: bit 7) |
| 16 | Blue-2 | 014 | General-purpose input | No. 44 | Input function selection 014 | 0: General-purpose input<br>1: Drive-source cutoff cancellation (ON edge) |
| 17 | Purple-2 | 015 | General-purpose input | No. 45 | Input function selection 015 | 0: General-purpose input<br>1: Home return of all effective axes (ON edge)<br>2: Home return of all effective incremental axes (ON edge) |
| 18 | Gray-2 | 016 | General-purpose input | | | |
| 19 | White-2 | 017 | General-purpose input | | | |
| 20 | Black-2 | 018 | General-purpose input | | | |
| 21 | Brown-3 | 019 | General-purpose input | | | |
| 22 | Red-3 | 020 | General-purpose input | | | |
| 23 | Orange-3 | 021 | General-purpose input | | | |
| 24 | Yellow-3 | 022 | General-purpose input | | | |
| 25 | Green-3 | 023 | General-purpose input | | | |
| 26 | Blue-3 | 024 | General-purpose input | | | |
| 27 | Purple-3 | 025 | General-purpose input | | | |
| 28 | Gray-3 | 026 | General-purpose input | | | |
| 29 | White-3 | 027 | General-purpose input | | | |
| 30 | Black-3 | 028 | General-purpose input | | | |
| 31 | Brown-4 | 029 | General-purpose input | | | |
| 32 | Red-4 | 030 | General-purpose input | | | |
| 33 | Orange-4 | 031 | General-purpose input | | | |

Output

| Pin No. | Wire color | Port No. | Standard Setting (in the delivery) Function | Parameter No. *1 | Parameter Name | Function |
|---|---|---|---|---|---|---|
| | | | | Remarks<br>When the unit is delivered, the output is set as shown in the table.<br>However, the output function can be changed using the I/O parameter setting. | | |
| 34 | YW-4 | 300 | Error Output at the Operation Cancellation Level or more (OFF) | No.46<br>No.331 | Output Function Selection 300<br>Output Function Selection 300 (Area 2) | 0: Universal Output<br>1: Error Output at the Operation Cancellation Level or more (ON)<br>2: Error Output at the Operation Cancellation Level or more (OFF)<br>3: Error Output at the Operation Cancellation Level or more + Emergency-stop output (ON)<br>4: Error Output at the Operation Cancellation Level or more + Emergency-stop output (OFF)<br>5: Error output of cold start level or more (ON)<br>6: Error output of cold start level or more (OFF)<br>7: Message level related to maintenance information alarm function (OFF) for error output of (Error No. 231 to 232) |
| 35 | GN-4 | 301 | READY Output (PIO Trigger Program Operation Available and without occurrence of any error at the cold start level or more) (Main Application Ver. 0.20 or later) | No.47<br>No.332 | Output Function Selection 301<br>Output Function Selection 301 (Area 2) | 0: Universal Input<br>1: READY Output (PIO Trigger Program Operation Available)<br>2: READY Output (PIO Trigger Program Operation Available)and without occurrence of any error at the operation cancellation level or more<br>3: READY Output (PIO Trigger Program Operation Available)and READY Output (PIO Trigger Program Operation Available, and without occurrence of any error at the cold start level or more or more level or more |
| 36 | BL-4 | 302 | Emergency-stop output (OFF) | No.48<br>No.333 | Output Function Selection 302<br>Output Function Selection 302 (Area 2) | 0: Universal Input<br>2: Emergency-stop output (ON)<br>3: Emergency-stop output (OFF) |
| 37 | PL-4 | 303 | Universal Output | No.49<br>No.334 | Output Function Selection 303<br>Output Function Selection 303 (Area 2) | 0: Universal Output<br>1: AUTO Mode Output<br>2: Output during the Automatic Operation (In addition, when the parameter No. 12 is set to "1") |
| 38 | GY-4 | 304 | Universal Output | No.50<br>No.335 | Output Function Selection 304<br>Output Function Selection 304 (Area 2) | 0: Universal Output<br>1: Output at the time of "All Effective Axes Homing (=0)"<br>2: Output when all the effective axes homing is completed (Coordinates determined)<br>3: Output when all the effective axes home preset coordinates are set |
| 39 | WT-4 | 305 | Universal Output | No.51<br>No.336 | Output Function Selection 305<br>Output Function Selection 305 (Area 2) | 0: Universal Output<br>1: Axis 1 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 1 servo ON |
| 40 | BK-4 | 306 | Universal Output | No.52<br>No.337 | Output Function Selection 306<br>Output Function Selection 306 (Area 2) | 0: Universal Output<br>1: Axis 2 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 2 servo ON |
| 41 | BR-5 | 307 | Universal Output | No.53<br>No.338 | Output Function Selection 307<br>Output Function Selection 307 (Area 2) | 0: Universal Output<br>1: Axis 3 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 3 servo ON |
| 42 | RD-5 | 308 | Universal Output | No.54<br>No.339 | Output Function Selection 308<br>Output Function Selection 308 (Area 2) | 0: Universal Output<br>1: Axis 4 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 4 servo ON |
| 43 | OR-5 | 309 | Universal Output | No.55<br>No.340 | Output Function Selection 309<br>Output Function Selection 309 (Area 2) | 0: Universal Output<br>1: Axis 5 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 5 servo ON |
| 44 | YW-5 | 310 | Universal Output | No.56<br>No.341 | Output Function Selection 310<br>Output Function Selection 310 (Area 2) | 0: Universal Output<br>1: Axis 6 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 6 servo ON |
| 45 | GN-5 | 311 | Universal Output | No.57<br>No.342 | Output Function Selection 311<br>Output Function Selection 311 (Area 2) | 0: Universal Output<br>1: Axis 7 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 7 servo ON (system monitoring task output) |
| 46 | BL-5 | 312 | Universal Output | No.58<br>No.343 | Output Function Selection 312<br>Output Function Selection 312 (Area 2) | 0: Universal Output<br>1: Axis 8 in-position output (turned OFF when pressing missed)<br>2: Output during the Axis 8 servo ON (system monitoring task output) |

| Pin No. | Wire color | Port No. | Standard Setting (in the delivery) Function |
|---|---|---|---|
| 47 | PL-5 | 313 | Universal Output |
| 48 | GY-5 | 314 | Universal Output |
| 49 | WT-5 | 315 | Universal Output |
| 50 | BK-5 | | 0V Output |

| Remarks | | |
|---|---|---|
| When the unit is delivered, the output is set as shown in the table. However, the output function can be changed using the I/O parameter setting. | | |
| Parameter No. *1 | Parameter Name | Function |
| No.59<br>No.344 | Output Function Selection 313<br>Output Function Selection 313 (Area 2) | 0: Universal Output<br>1: Reserved by the system |
| No.60<br>No.345 | Output Function Selection 314<br>Output Function Selection 314 (Area 2) | 0: Universal Output<br>1: Voltage low warning for absolute-data backup battery |
| No.61<br>No.346 | Output Function Selection 315<br>Output Function Selection 315 (Area 2) | 0: Universal Output |

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, "Input filtering period".

[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

XSEL-RA/SA/RAX/SAX/RAXD/SAXD Virtual Input Ports (Internal Flags)

| Port No. | Input/ Output | Function |
|---|---|---|
| 7000 | Input | Always OFF |
| 7001 | | Always ON |
| 7002 to 7005 | | (Reserved by the system = Use is strictly prohibited) |
| 7006 | | Critical system error = A message level error is present. |
| 7007 | | Critical system error = An operation-cancellation level error is present. |
| 7008 | | Critical system error = A cold-start level error is present. |
| 7009 | | (Reserved by the system = Use is strictly prohibited) |
| 7010 | | A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input). |
| 7011 | | A latch signal indicating that a cause of all-operation cancellation is present.<br>(This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON) |
| 7012 | | A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed).<br>(Effective only in the auto operation recognition mode) |
| 7013 | | A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port) |
| 7014 | | (Reserved by the system = Use is strictly prohibited) |
| 7015 | | Axis 1 absolute-data backup battery voltage low warning |
| 7016 | | Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7017 | | Axis 2 absolute-data backup battery voltage low warning |
| 7018 | | Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7019 | | Axis 3 absolute-data backup battery voltage low warning |
| 7020 | | Axis 3 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7021 | | Axis 4 absolute-data backup battery voltage low warning |
| 7022 | | Axis 4 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7023 | | Axis 5 absolute-data backup battery voltage low warning |
| 7024 | | Axis 5 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7025 | | Axis 6 absolute-data backup battery voltage low warning |
| 7025 | | Axis 6 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7027 | | Axis 7 absolute-data backup battery voltage low warning |
| 7028 | | Axis 7 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7029 | | Axis 8 absolute-data backup battery voltage low warning |
| 7030 | | Axis 8 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7031 to 7040 | | (Reserved by the system = Use is strictly prohibited) |
| 7041 to 7042 | | (For future expansion = Use is strictly prohibited) |
| 7043 | | Axis 1 home return completion |
| 7044 | | Axis 2 home return completion |
| ~ | | ~ |
| 7048 | | Axis 6 home return completion |
| 7049 | | Axis 7 home return completion |
| 7050 | | Axis 8 home return completion |
| 7051 | | Axis 1 servo ON completion (Main application V1.10 or later) |
| 7052 | | Axis 2 servo ON completion (Main application V1.10 or later) |
| ~ | | ~ |
| 7056 | | Axis 6 servo ON completion (Main application V1.10 or later) |
| 7057 | | Axis 7 servo ON completion (Main application V1.10 or later) |
| 7058 | | Axis 8 servo ON completion (Main application V1.10 or later) |
| 7059 | | In Compliance Mode (Base Coordinate System Compliance Mode)<br>(Axis 1 to 4 SCARA robot) (Main application V1.13 or later) |
| 7060 to 7063 | | (Reserved by the system = Use is strictly prohibited) |

XSEL-RA/SA/RAX/SAX/RAXD/SAXD Virtual Input Ports (Internal Flags)

| Port No. | Input/ Output | Function |
|---|---|---|
| 7064 | Input | In Compliance Mode (Base Coordinate System Compliance Mode)<br>(Axis 5 to 8 SCARA robot) (Main application V1.13 or later) |
| 7065 to 7068 | | (Reserved by the system = Use is strictly prohibited) |
| 7069 | | (For future expansion = Use is strictly prohibited) |
| 7070 | | (Reserved by the system = Use is strictly prohibited) |
| 7071 | | In the AUTO mode |
| 7072 | | During auto operation |
| 7073 to 7074 | | (For future expansion = Use is strictly prohibited) |
| 7075 | | During Tracking Conveyor Speed Drop Detection |
| 7076 | | Tracking Operation Complete Work Position Arrival Latch Signal (Latches until “TRAC 1 nnn” or “Tracking Operation Execution Program Finish”) |
| 7077 | | In Tracking Conveyor Tracking Complete Range |
| 7078 | | Tracking Reversed Operation Detected Work Position Arrival Latch Signal (Latches until “TRAC 1 nnn” or “Tracking Operation Execution Program Finish”) |
| 7079 | | During Tracking Mode (Work detection valid) (for SCARA controller only) |
| 7080 | | During Tracking Operation (including a pause in tracking operation) |
| 7081 to 7100 | | (Reserved by the system = Use is strictly prohibited) |
| 7101 | | Program No. 01 is being executed (or paused). |
| 7102 | | Program No. 02 is being executed (or paused). |
| 7103 | | Program No. 03 is being executed (or paused). |
| ~ | | ~ |
| 7227 | | Program No. 127 is being executed (or paused). |
| 7228 | | Program No. 128 is being executed (or paused). |
| 7229 to 7299 | | (Reserved by the system = Use is strictly prohibited) |

## XSEL-RA/SA/RAX/SAX/RAXD/SAXD Virtual Input Ports (Internal Flags)

| Port No. | Input/ Output | Function |
|---|---|---|
| 7300 | Output | A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.) (7300 is turned OFF after latch cancellation is attempted.) |
| 7301 | | Axis 1 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 "Brake Output Control Method Select" = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 1.41 or later) |
| 7302 | | Axis 2 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 "Brake Output Control Method Select" = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 1.41 or later) |
| 7303 | | Axis 3 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 "Brake Output Control Method Select" = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 1.41 or later) |
| 7304 | | Axis 4 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 "Brake Output Control Method Select" = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 1.41 or later) |
| 7305 | | Axis 5 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 "Brake Output Control Method Select" = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 1.41 or later) |
| 7306 | | Axis 6 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 "Brake Output Control Method Select" = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted.<br>(main application versions 1.41 or later) |
| 7307 | | Axis 7 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 "Brake Output Control Method Select" = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted. |
| 7308 | | Axis 8 Gripper Release Output (On; Releasing Command, Off; Gripping Command)<br>* Each Axis Parameter No. 103 "Brake Output Control Method Select" = 1 (Valid when User Control)<br>The purpose of signal may differ depending on the actuator to be mounted. |
| 7309 | Input | (For future expansion = Use is strictly prohibited) |
| 7310 | | In Use of 1st Axis Collision Detection Feature (Main application V1.10 or later) |
| 7311 | | In Use of 2nd Axis Collision Detection Feature (Main application V1.10 or later) |
| ~ | | ~ |
| 7315 | | In Use of 6th Axis Collision Detection Feature (Main application V1.10 or later) |
| 7316 | | In Use of 7th Axis Collision Detection Feature (Main application V1.10 or later) |
| 7317 | | In Use of 8th Axis Collision Detection Feature (Main application V1.10 or later) |
| 7318 to 7380 | | (For future expansion = Use is strictly prohibited) |
| 7381 to 7399 | | (Reserved by the system = Use is strictly prohibited) |
| 7400 | | (For future expansion = Use is strictly prohibited) |
| 7401 | | Program No. 129 is being executed (or paused) |
| 7402 | | Program No. 130 is being executed (or paused) |
| 7403 | | Program No. 131 is being executed (or paused) |
| ~ | | ~ |
| 7526 | | Program No. 254 is being executed (or paused) |
| 7527 | | Program No. 255 is being executed (or paused) |
| 7528 to 7380 | Output | (For future expansion = Use is strictly prohibited) |

### 2.1.6 SSEL, ASEL, PSEL Controllers

[1] Input and Output I/O Port

With SSEL, ASEL and PSEL controllers, input and output functions can be assigned to input and output ports as desired.
For input ports, set input function setting values (0 to 23) in input function selections 000 to 015 (I/O parameters 30 to 45) corresponding to port No. 000 to 015 or input function selections 016 to 023 (I/O parameters 251 to 258) corresponding to port No. 16 to 23, and the set functions will be assigned.
For output ports, set output function setting values (0 to 17, 24, 25) in output function selections 300 to 307 (I/O parameters 46 to 53) corresponding to port No. 300 to 307, and the set functions will be assigned.

Program mode
Input

| Pin No. | Wire color | Port No. | Standard (factory-set) function | Parameter No. | Parameter name | Input function setting value (factory setting) |
|---|---|---|---|---|---|---|
| 1A | Brown 1 | - | I/O power supply +24 V | - | - | - |
| 1B | Red 1 | 016 | Program No. 1 selection (MSB) | 251 | Input function selection 016 | 9 |
| 2A | Orange 1 | 017 | Program No. 2 selection (bit 2) | 252 | Input function selection 017 | 10 |
| 2B | Yellow 1 | 018 | Program No. 4 selection (bit 3) | 253 | Input function selection 018 | 11 |
| 3A | Green 1 | 019 | Program No. 8 selection (bit 4) | 254 | Input function selection 019 | 12 |
| 3B | Blue 1 | 020 | Program No. 10 selection (bit 5) | 255 | Input function selection 020 | 13 |
| 4A | Purple 1 | 021 | Program No. 20 selection (bit 6) | 256 | Input function selection 021 | 14 |
| 4B | Gray 1 | 022 | Program No. 40 selection (LSB: bit 7) | 257 | Input function selection 022 | 15 |
| 5A | White 1 | 023 | Software reset | 258 | Input function selection 023 | 3 |
| 5B | Black 1 | 000 | Program start | 30 | Input function selection 000 | 1 |
| 6A | Brown 2 | 001 | General-purpose input | 31 | Input function selection 001 | 0 |
| 6B | Red 2 | 002 | General-purpose input | 32 | Input function selection 002 | 0 |
| 7A | Orange 2 | 003 | General-purpose input | 33 | Input function selection 003 | 0 |
| 7B | Yellow 2 | 004 | General-purpose input | 34 | Input function selection 004 | 0 |
| 8A | Green 2 | 005 | General-purpose input | 35 | Input function selection 005 | 0 |
| 8B | Blue 2 | 006 | General-purpose input | 36 | Input function selection 006 | 0 |
| 9A | Purple 2 | 007 | General-purpose input | 37 | Input function selection 007 | 0 |
| 9B | Gray 2 | 008 | General-purpose input | 38 | Input function selection 008 | 0 |
| 10A | White 2 | 009 | General-purpose input | 39 | Input function selection 009 | 0 |
| 10B | Black 2 | 010 | General-purpose input | 40 | Input function selection 010 | 0 |
| 11A | Brown 3 | 011 | General-purpose input | 41 | Input function selection 011 | 0 |
| 11B | Red 3 | 012 | General-purpose input | 42 | Input function selection 012 | 0 |
| 12A | Orange 3 | 013 | General-purpose input | 43 | Input function selection 013 | 0 |
| 12B | Yellow 3 | 014 | General-purpose input | 44 | Input function selection 014 | 0 |
| 13A | Green 3 | 015 | General-purpose input | 45 | Input function selection 015 | 0 |

| Input function setting value | Function |
|---|---|
| 0 | General-purpose input |
| 1 | Program start (BCD) (ON edge) signal |
| 2 | Program start (BIN) (ON edge) signal |
| 3 | Soft reset signal (ON for 1sec) |
| 4 | Servo ON signal (ON edge) |
| 5 | Auto program start signal (ON edge) |
| 6 | All-servo-axis software interlock (OFF level) |
| 7 | Operation pause cancellation input (ON edge) |
| 8 | Operation pause signal (OFF level) |
| 9 | Program number specification (MSB) |
| 10 | Program number specification (bit 2) |
| 11 | Program number specification (bit 3) |
| 12 | Program number specification (bit 4) |
| 13 | Program number specification (bit 5) |
| 14 | Program number specification (bit 6) |
| 15 | Program number specification (LSB: bit 7) |
| 16 | Error reset (ON edge) |
| 17 | Drive-source cutoff cancellation input (ON edge) |
| 18 | All-effective-axis home return command signal (ON edge) |
| 19 | All-effective-incremental-axis home return (ON edge) |
| 20 | PC/teaching pendant servo movement command acceptance input |
| 21 | Remote mode control input |
| 22 | Axis 1 forced brake release input |
| 23 | Axis 2 forced brake release input |
| 24 to 27 | Reserved by the system |
| 24 | Program number specification (bit 8) |
| 25 | Program number specification (bit 9) |

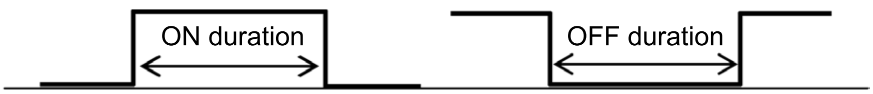
Program mode
Output

| Pin No. | Wire color | Port No. | Standard (factory-set) function | Parameter No. | Parameter name | Input function setting value (factory setting) |
|---|---|---|---|---|---|---|
| 13B | Blue 3 | 300 | Alarm output | 46 | Output function selection 300 | 2 |
| 14A | Purple 3 | 301 | Ready output | 47 | Output function selection 301 | 7 |
| 14B | Gray 3 | 302 | General-purpose output | 48 | Output function selection 302 | 0 |
| 15A | White 3 | 303 | General-purpose output | 49 | Output function selection 303 | 0 |
| 15B | Black 3 | 304 | General-purpose output | 50 | Output function selection 304 | 0 |
| 16A | Brown 4 | 305 | General-purpose output | 51 | Output function selection 305 | 0 |
| 16B | Red 4 | 306 | General-purpose output | 52 | Output function selection 306 | 0 |
| 17A | Orange 4 | 307 | General-purpose output | 53 | Output function selection 307 | 0 |
| 17B | Yellow 4 | N | I/O power supply 0V | - | - | - |

| Input function setting value | Function |
|---|---|
| 0 | General-purpose input |
| 1 | Output of operation-cancellation level or higher error (ON) |
| 2 | Output of operation-cancellation level or higher error (OFF) |
| 3 | Output of operation-cancellation level or higher error + Emergency stop output (ON) |
| 4 | Output of operation-cancellation level or higher error + Emergency stop output (OFF) |
| 5 | READY output (PIO-trigger program operation enabled) |
| 6 | READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error) |
| 7 | READY output (PIO-trigger program operation enabled AND no cold-start level or higher error) |
| 8 | Emergency stop output (ON) |
| 9 | Emergency stop output (OFF) |
| 10 | AUTO mode output |
| 11 | Auto operation output |
| 12 | Output when all effective axes are home (= 0) |
| 13 | Output when all effective axes have completed home return |
| 14 | Output when all effective axes are at home preset coordinate |
| 15 | System-memory backup battery (optional) voltage low warning output |
| 16 | Absolute-data backup battery (optional) voltage low warning output |
| 17 | Drive-source cutoff (SDN) notification output |
| 24 | Axis 1 servo ON output |
| 25 | Axis 2 servo ON output |

*1 Output function setting values 1, 2, 3 and 4 cannot be assigned at the same time.
*2 Output function setting values 5, 6 and 7 cannot be assigned at the same time.

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, "Input filtering period".



[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

ASEL/PSEL/SSEL Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7000 | Always OFF |
| 7001 | Always ON |
| 7002 | System-memory backup battery voltage low warning |
| 7003 | System-memory backup battery voltage error |
| 7004 | (Reserved by the system = Use is strictly prohibited) |
| 7005 | (Reserved by the system = Use is strictly prohibited) |
| 7006 | Critical system error = A message level error is present. |
| 7007 | Critical system error = An operation-cancellation level error is present. |
| 7008 | Critical system error = A cold-start level error is present. |
| 7009 | (Reserved by the system = Use is strictly prohibited) |
| 7010 | A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input). |
| 7011 | A latch signal indicating that a cause of all-operation cancellation is present.<br>(This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON) |
| 7012 | A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed).<br>(Effective only in the auto operation recognition mode) |
| 7013 | A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port) |
| 7014 | (Reserved by the system = Use is strictly prohibited) |
| 7015 | Axis 1 absolute-data backup battery voltage low warning |
| 7016 | Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7017 | Axis 2 absolute-data backup battery voltage low warning |
| 7018 | Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7019 to 7026 | (For future expansion = Use is strictly prohibited) |
| 7027 to 7040 | (Reserved by the system = Use is strictly prohibited) |
| 7041, 7042 | (For future expansion = Use is strictly prohibited) |
| 7043 | Axis 1 home return completion |
| 7044 | Axis 2 home return completion |
| 7045 to 7070 | (For future expansion = Use is strictly prohibited) |
| 7071 | In the AUTO mode |
| 7072 | During auto operation |
| 7073 to 7100 | (Reserved by the system = Use is strictly prohibited) |
| 7101 | Program No. 01 is being executed (or paused). |
| ~ | ~ |
| 7164 | Program No. 64 is being executed (or paused). |
| 7165 | Program No. 65 is being executed (or paused). ··· Dedicated only for SSEL with expanded memory capacity |
| ~ | ~ |
| 7228 | Program No. 128 is being executed (or paused). Dedicated only for SSEL with expanded memory capacity |
| 7229 to 7299 | (For future expansion = Use is strictly prohibited) |

ASEL/PSEL/SSEL Virtual Output Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7300 | A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.)<br>(7300 is turned OFF after latch cancellation is attempted.) |
| 7301 to 7380 | (For future expansion = Use is strictly prohibited) |
| 7381 to 7399 | (Reserved by the system = Use is strictly prohibited) |
| 7400 to 7599 | (For future expansion = Use is strictly prohibited) |

### 2.1.7 Tabletop Robot TT/TTA

[1] Input and Output I/O Port

With the tabletop robot TT, input and output functions can be assigned to input and output ports as desired.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
You can also use I/O parameters 331 to 346 (output function selections 300 (area 2) to 315 (area 2)) to set output functions and then use I/O parameters 315 to 330 to set the port numbers to assign the respective functions to.

Input

| Pin No. | Wire color | Port No. | Standard (factory-set) function |
|---|---|---|---|
| 1 | Brown 1 | - | I/O power supply +24V |
| 2 | Red 1 | 016 | General-purpose input |
| 3 | Orange 1 | 017 | General-purpose input |
| 4 | Yellow 1 | 018 | General-purpose input |
| 5 | Green 1 | 019 | General-purpose input |
| 6 | Blue 1 | 020 | General-purpose input |
| 7 | Purple 1 | 021 | General-purpose input |
| 8 | Gray 1 | 022 | General-purpose input |
| 9 | White 1 | 023 | General-purpose input |
| 10 | Black 1 | 024 | General-purpose input |
| 11 | Brown 2 | 025 | General-purpose input |
| 12 | Red 2 | 026 | General-purpose input |
| 13 | Orange 2 | 027 | General-purpose input |
| 14 | Yellow 2 | 028 | General-purpose input |
| 15 | Green 2 | 029 | General-purpose input |
| 16 | Blue 2 | 030 | General-purpose input |
| 17 | Purple 2 | 031 | General-purpose input |

Remarks

Inputs are set as general-purpose inputs, but you can change these input functions by setting applicable I/O parameters.

| Parameter No. | Parameter name | Function |
|---|---|---|
| 30 | Input function selection 000*1 | 1: Program start |
| 31 | Input function selection 001 | 0: General-purpose input<br>1: Soft reset signal |
| 32 | Input function selection 002 | 0: General-purpose input<br>1: Soft reset signal |
| 33 | Input function selection 003 | 0: General-purpose input<br>1: Auto program start upon power-ON reset or software reset in AUTO mode<br>2: Auto program start signal |
| 34 | Input function selection 004 | 0: General-purpose input<br>1: Software interlock of all servo axes (OFF level) |
| 35 | Input function selection 005 | 0: General-purpose input<br>1: Operation pause cancellation input (ON edge) |
| 36 | Input function selection 006 | 0: General-purpose input<br>1: Operation pause signal (OFF level) |
| 37 | Input function selection 007*2 | 0: General-purpose input<br>1: Program number specification (LSB) |
| 38 | Input function selection 008*2 | 0: General-purpose input<br>1: Program number specification (bit 2) |
| 39 | Input function selection 009*2 | 0: General-purpose input<br>1: Program number specification (bit 3) |
| 40 | Input function selection 010*2 | 0: General-purpose input<br>1: Program number specification (bit 4) |
| 41 | Input function selection 011*2 | 0: General-purpose input<br>1: Program number specification (bit 5) |
| 42 | Input function selection 012*2 | 0: General-purpose input<br>1: Program number specification (bit 6) |
| 43 | Input function selection 013*2 | 0: General-purpose input<br>1: Program number specification (MSB: bit 7)<br>2: Error reset (ON edge) |
| 44 | Input function selection 014 | 0: General-purpose input<br>1: Drive-source cutoff cancellation (ON edge) |
| 45 | Input function selection 015 | 0: General-purpose input<br>1: Home return of all effective axes (ON edge)<br>2: Home return of all effective incremental axes (ON edge) |

*1 If input function selection 000 (program start) is assigned to a port other than No. 000, the start switch on the front panel is disabled.
*2 If input function selections 007 to 013 (program selection switches) are assigned to ports other than No. 007 to 013, the program selection switches on the front panel are disabled.

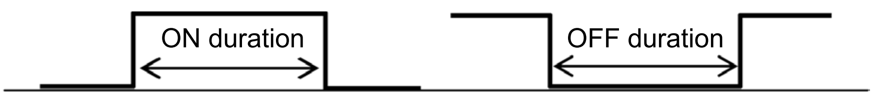
Output

| Pin No. | Wire color | Port No. | Standard (factory-set) function |
|---|---|---|---|
| 18 | Gray 2 | 316 | General-purpose output |
| 19 | White 2 | 317 | General-purpose output |
| 20 | Black 2 | 318 | General-purpose output |
| 21 | Brown 3 | 319 | General-purpose output |
| 22 | Red 3 | 320 | General-purpose output |
| 23 | Orange 3 | 321 | General-purpose output |
| 24 | Yellow 3 | 322 | General-purpose output |
| 25 | Green 3 | 323 | General-purpose output |
| 26 | Blue 3 | 324 | General-purpose output |
| 27 | Purple 3 | 325 | General-purpose output |
| 28 | Gray 3 | 326 | General-purpose output |
| 29 | White 3 | 327 | General-purpose output |
| 30 | Black 3 | 328 | General-purpose output |
| 31 | Brown 4 | 329 | General-purpose output |
| 32 | Red 4 | 330 | General-purpose output |
| 33 | Orange 4 | 331 | General-purpose output |
| 34 | Yellow 4 | - | I/O power supply 0 V |

| Remarks | | |
|---|---|---|
| Outputs are set as general-purpose outputs, but you can change these output functions by setting applicable I/O parameters. | | |
| Parameter No. | Parameter name | Function |
| 46<br>331 | Output function selection 300*3<br>Output function selection 300 (area 2) | 0: General-purpose output<br>1: Output of operation-cancellation level or higher error (ON)<br>2: Output of operation-cancellation level or higher error (OFF)<br>3: Output of operation-cancellation level or higher error + Emergency stop output (ON)<br>4: Output of operation-cancellation level or higher error + Emergency stop output (OFF) |
| 47<br>332 | Output function selection 301*3<br>Output function selection 301 (area 2) | 0: General-purpose output<br>1: READY output (PIO-trigger program operation enabled)<br>2: READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error)<br>3: READY output (PIO-trigger program operation enabled AND no cold-start level or higher error) |
| 48<br>333 | Output function selection 302*3<br>Output function selection 302 (area 2) | 0: General-purpose output<br>1: Emergency stop output (ON)<br>2: Emergency stop output (OFF) |
| 49<br>334 | Output function selection 303*3<br>Output function selection 303 (area 2) | 0: General-purpose output<br>1: AUTO mode output<br>2: Auto operation output (When other parameter No. 12 is set to '1') |
| 50<br>335 | Output function selection 304*3<br>Output function selection 304 (area 2) | 0: General-purpose output<br>1: Output when all effective axes are home (= 0)<br>2: Output when all effective axes have completed home return<br>3: Output when all effective axes are at home preset coordinate |
| 51<br>336 | Output function selection 305<br>Output function selection 305 (area 2) | 0: General-purpose output<br>2: Axis 1 servo ON output |
| 52<br>337 | Output function selection 306<br>Output function selection 306 (area 2) | 0: General-purpose output<br>2: Axis 2 servo ON output |
| 53<br>338 | Output function selection 307<br>Output function selection 307 (area 2) | 0: General-purpose input<br>2: Axis 3 servo ON output |
| 54<br>339 | Output function selection 308<br>Output function selection 308 (area 2) | 0: General-purpose output |
| 55<br>340 | Output function selection 309<br>Output function selection 309 (area 2) | 0: General-purpose output |
| 56<br>341 | Output function selection 310<br>Output function selection 310 (area 2) | 0: General-purpose output |
| 57<br>342 | Output function selection 311<br>Output function selection 311 (area 2) | 0: General-purpose output |
| 58<br>343 | Output function selection 312<br>Output function selection 312 (area 2) | 0: General-purpose output |
| 59<br>344 | Output function selection 313<br>Output function selection 313 (area 2) | 0: General-purpose output |
| 60<br>345 | Output function selection 314<br>Output function selection 314 (area 2) | 0: General-purpose output |
| 61<br>346 | Output function selection 315<br>Output function selection 315 (area 2) | 0: General-purpose output |

*3 Since output function selections 300 to 304 are assigned to LEDs in the panel window, the LEDs are disabled if parameters 46 to 50 are set as general-purpose outputs or port number assignments are changed using parameters 299 to 303.
If you want to output system signals to the I/O shown in the tables above, use output function selection area 2.

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, "Input filtering period".



[2] Virtual I/O Port

Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

TT Virtual Input Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7000 | Always OFF |
| 7001 | Always ON |
| 7002 | System-memory backup battery voltage low warning |
| 7003 | System-memory backup battery voltage error |
| 7004 | Reserved by the system = Use is prohibited |
| 7005 | Reserved by the system = Use is prohibited |
| 7006 | Critical system error = A message level error is present. |
| 7007 | Critical system error = An operation-cancellation level error is present. |
| 7008 | Critical system error = A cold-start level error is present. |
| 7009 | Reserved by the system = Use is prohibited |
| 7010 | A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input). |
| 7011 | A latch signal indicating that a cause of all-operation cancellation is present.<br>(This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON) |
| 7012 | A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed).<br>(Effective only in the auto operation recognition mode) |
| 7013 | A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port) |
| 7014 to 7050 | Reserved by the system = Use is prohibited |
| 7051 to 7070 | For future expansion = Use is prohibited |
| 7071 | In the AUTO mode |
| 7072 | During auto operation |
| 7073 to 7100 | Reserved by the system = Use is prohibited |
| 7101 | Program No. 01 is being executed (or paused). |
| ~ | ~ |
| 7164 | Program No. 64 is being executed (or paused). |
| 7165 to 7299 | For future expansion = Use is prohibited |

TT Virtual Output Ports (Internal Flags)

| Port No. | Function |
|---|---|
| 7300 | A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.)<br>(7300 is turned OFF after latch cancellation is attempted.) |
| 7301 to 7380 | (For future expansion = Use is strictly prohibited) |
| 7381 to 7399 | (Reserved by the system = Use is strictly prohibited) |
| 7400 to 7599 | (For future expansion = Use is strictly prohibited) |

TTA Virtual Input/Output Ports (Internal Flags)

| Port No. | Input/Output | Function |
|---|---|---|
| 7000 | Input | Always OFF |
| 7001 | | Always ON |
| 7002 | | Reserved by the system = Use is strictly prohibited |
| 7003 | | Reserved by the system = Use is strictly prohibited |
| 7004 | | Reserved by the system = Use is strictly prohibited |
| 7005 | | Reserved by the system = Use is strictly prohibited |
| 7006 | | Critical system error = A message level error is present. |
| 7007 | | Critical system error = An operation-cancellation level error is present. |
| 7008 | | Critical system error = A cold-start level error is present. |
| 7009 | | Reserved by the system = Use is strictly prohibited |
| 7010 | | A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input). |
| 7011 | | A latch signal indicating that a cause of all-operation cancellation is present.<br>(This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON) |
| 7012 | | A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed).<br>(Effective only in the auto operation recognition mode) |
| 7013 | | A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port) |
| 7014 | | Reserved by the system = Use is strictly prohibited |
| 7015 | | Axis 1 absolute-data backup battery voltage low warning (feature dedicated only for TTA-S) |
| 7016 | | Axis 1 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) |
| 7017 | | Axis 2 absolute-data backup battery voltage low warning (feature dedicated only for TTA-S) |
| 7018 | | Axis 2 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) (feature dedicated only for TTA-S) |
| 7019 | | Axis 3 absolute-data backup battery voltage low warning (feature dedicated only for TTA-S) |
| 7020 | | Axis 3 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) (feature dedicated only for TTA-S) |
| 7021 | | Axis 4 absolute-data backup battery voltage low warning (feature dedicated only for TTA-S) |
| 7022 | | Axis 4 absolute-data backup battery voltage error (Latched until power-ON reset or software reset) (feature dedicated only for TTA-S) |
| 7023 to 7042 | | Reserved by the system = Use is strictly prohibited |
| 7043 | | Axis 1 home return completion (Main application V2.00 or later) |
| 7044 | | Axis 2 home return completion (Main application V2.00 or later) |
| 7045 | | Axis 3 home return completion (Main application V2.00 or later) |
| 7046 | | Axis 4 home return completion (Main application V2.00 or later) |
| 7047 to 7050 | | Reserved by the system = Use is strictly prohibited |
| 7051 | | Axis 1 servo-ON completion (Main application V2.10 or later) |
| 7052 | | Axis 2 servo-ON completion (Main application V2.10 or later) |
| 7053 | | Axis 3 servo-ON completion (Main application V2.10 or later) |
| 7054 | | Axis 4 servo-ON completion (Main application V2.10 or later) |
| 7055 to 7070 | | For future expansion = Use is strictly prohibited |
| 7071 | | In the AUTO mode |
| 7072 | | During auto operation |
| 7073 to 7074 | | For future expansion = Use is strictly prohibited |
| 7075 to 7100 | | During Tracking Conveyor Speed Drop Detection |
| 7101 | | Program No. 01 is being executed (or paused). |
| ~ | | ~ |
| 7228 | | Program No. 128 is being executed (or paused). |
| 7229 to 7299 | | For future expansion = Use is strictly prohibited |

TTA Virtual Input/Output Ports (Internal Flags)

| Port No. | Input/Output | Function |
|---|---|---|
| 7300 | Output | A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.)<br>(7300 is turned OFF after latch cancellation is attempted.) |
| 7301 to 7380 | | (For future expansion = Use is strictly prohibited) |
| 7381 to 7399 | | (Reserved by the system = Use is strictly prohibited) |
| 7400 | | (For future expansion = Use is strictly prohibited) |
| 7401 | Input | Program No. 129 is being executed (or paused). |
| ~ | | ~ |
| 7527 | | Program No. 255 is being executed (or paused). |
| 7528 to 7599 | Output | (For future expansion = Use is strictly prohibited) |

## 2.1.8 MSEL Controller

[1] Input and Output I/O Port

In Input Function Select No. 000 to 015 and Output Function Select No. 300 to 315, dedicated functions can be set, and they can be assigned to desired input and output ports.
For input ports, set input functions using I/O parameters 30 to 45 (input function selections 000 to 015) and then use I/O parameters 283 to 298 to set the port numbers to assign the respective functions to.
For output ports, set output functions using I/O parameters 46 to 61 (output function selections 300 to 315) and then use I/O parameters 299 to 314 to set the port numbers to assign the respective functions to.
For other I/O port Nos., they can be used freely as the universal I/O port.

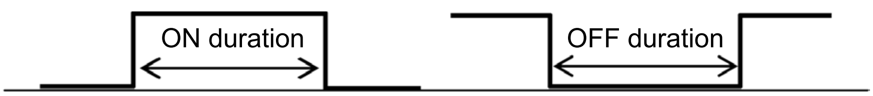
(1) Input Port Function Assignment

| Input function selection No. | Parameter No. | | Setting values | Input Signal Functions | Setting at delivery |
|---|---|---|---|---|---|
| 000 | No.30 | | 0 | General-purpose input | |
| | | | 1 | Program start (Indicated input port = 007 to 014, Start of program number indicated in BCD ... ON edge) | ○ |
| | | | 2 | Program start (Indicated input port = 007 to 014, Start of program number indicated in binary ... ON edge) | |
| 001 | No.31 | | 0 | General-purpose input | ○ |
| | | | 1 | Software reset signal (Kept on continuously for 1sec) | |
| 002 | No.32 | | 0 | General-purpose input | ○ |
| | | | 1 | Servo ON signal (Edge input) | |
| 003 | No.33 | | 0 | General-purpose input | |
| | | | 1 | Indicated program automatically starts by power-on reset / software reset in AUTO Mode | ○ |
| | | | 2 | Start of Auto Start Program (kept on continuously for 100ms) | |
| 004 | No.34 | | 0 | General-purpose input | ○ |
| | | | 1 | Software interlock on all servo axes (level signal with always being on) | |
| 005 | No.35 | | 0 | General-purpose input | ○ |
| | | | 1 | Operation pause cancellation input (On-edge process signal) | |
| 006 | No.36 | | 0 | General-purpose input | ○ |
| | | | 1 | Operation pause cancellation input (level signal with always being on) | |
| 007 | No.37 | Independent from value in Input Function Select 000 | 0 | General-purpose input | |
| | | When Input Function Select 000 = 1 | 1 | Program number specification 0 bit | ○ |
| | | When Input Function Select 000 = 2 | | | |
| 008 | No.38 | Independent from value in Input Function Select 000 | 0 | General-purpose input | |
| | | When Input Function Select 000 = 1 | 1 | Program number specification 1 bit | ○ |
| | | When Input Function Select 000 = 2 | | | |
| 009 | No.39 | Independent from value in Input Function Select 000 | 0 | General-purpose input | |
| | | When Input Function Select 000 = 1 | 1 | Program number specification 2 bit | ○ |
| | | When Input Function Select 000 = 2 | | | |
| 010 | No.40 | Independent from value in Input Function Select 000 | 0 | General-purpose input | |
| | | When Input Function Select 000 = 1 | 1 | Program number specification 3 bit | ○ |
| | | When Input Function Select 000 = 2 | | | |
| 011 | No.41 | Independent from value in Input Function Select 000 | 0 | General-purpose input | |
| | | When Input Function Select 000 = 1 | 1 | Program number specification 4 bit | ○ |
| | | When Input Function Select 000 = 2 | | | |
| 012 | No.42 | Independent from value in Input Function Select 000 | 0 | General-purpose input | |
| | | When Input Function Select 000 = 1 | 1 | Program number specification 5 bit | ○ |
| | | When Input Function Select 000 = 2 | | | |
| 013 | No.43 | Independent from value in Input Function Select 000 | 0 | General-purpose input | |
| | | When Input Function Select 000 = 1 | 1 | Program number specification 6 bit | ○ |
| | | When Input Function Select 000 = 2 | | | |
| 014 | No.44 | Independent from value in Input Function Select 000 | 0 | General-purpose input | |
| | | When Input Function Select 000 = 1 | 1 | Program number specification 7 bit | ○ |
| | | When Input Function Select 000 = 2 | | | |
| 015 | No.45 | | 0 | General-purpose input | ○ |

(2) Output Port Function Assignment

| Input function selection No. | Parameter No. | Setting values | Input Signal Functions | Setting at delivery |
|---|---|---|---|---|
| 300 | No.46 | 0 | General-purpose output | |
| | | 1 | Output of operation-cancellation level or higher error (ON) | |
| | | 2 | Output of operation-cancellation level or higher error (OFF) | ○ |
| | | 3 | Output of operation-cancellation level or higher error + Emergency stop output (ON) | |
| | | 4 | Output of operation-cancellation level or higher error + Emergency stop output (OFF) | |
| | | 5 | Error output of cold start level or more (ON) | |
| | | 6 | Error output of cold start level or more (OFF) | |
| | | 7 | Message level related to maintenance information alarm function (ON) for error output of (Error No. 231 to 232) | |
| | | 8 | Message level related to maintenance information alarm function (OFF) for error output of (Error No. 231 to 232) | |
| 301 | No.47 | 0 | General-purpose output | |
| | | 1 | READY output (PIO-trigger program operation enabled) | |
| | | 2 | READY output (PIO-trigger program operation enabled AND no operation-cancellation level or higher error) | |
| | | 3 | READY output (PIO-trigger program operation enabled AND no cold-start level or higher error)(ON) | ○ |
| 302 | No.48 | 0 | General-purpose output | |
| | | 1 | Emergency stop output（ON） | |
| | | 2 | Emergency stop output（OFF） | ○ |
| 303 | No.49 | 0 | General-purpose output | ○ |
| 304 | No.50 | 0 | General-purpose output | ○ |
| 305 | No.51 | 0 | General-purpose output | ○ |
| 306 | No.52 | 0 | General-purpose output | ○ |
| 307 | No.53 | 0 | General-purpose output | ○ |
| 308 | No.54 | 0 | General-purpose output | ○ |
| 309 | No.55 | 0 | General-purpose output | ○ |
| 310 | No.56 | 0 | General-purpose output | ○ |
| 311 | No.57 | 0 | General-purpose output | ○ |
| 312 | No.58 | 0 | General-purpose output | ○ |
| 313 | No.59 | 0 | General-purpose output | ○ |
| 314 | No.60 | 0 | General-purpose output | ○ |
| 315 | No.61 | 0 | General-purpose output | ○ |

- By default, the ON/OFF state of an input signal is recognized by the controller when the signal has remained ON/OFF for approx. 4msec or more.
- The setting for this ON/OFF duration can be changed using I/O parameter No. 20, "Input filtering period".



[2] Virtual I/O Port

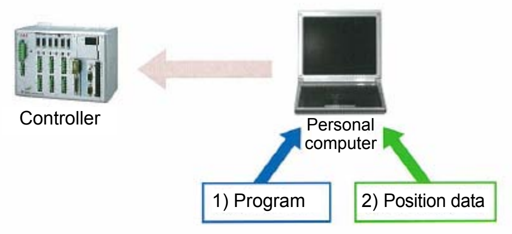
Virtual I/O ports are provided so that the controller can notify internal information. They are used to warn a low power-supply voltage, notify errors, etc. Use these ports as necessary.

MSEL Virtual Input Ports (Internal Flags)

| Port No. | Input/Output | Function |
|---|---|---|
| 7000 | Input | Always OFF |
| 7001 | | Always ON |
| 7002 | | Reserved by the system = Use is strictly prohibited |
| 7003 | | Reserved by the system = Use is strictly prohibited |
| 7004 | | Reserved by the system = Use is strictly prohibited |
| 7005 | | Reserved by the system = Use is strictly prohibited |
| 7006 | | Critical system error = A message level error is present. |
| 7007 | | Critical system error = An operation-cancellation level error is present. |
| 7008 | | Critical system error = A cold-start level error is present. |
| 7009 | | Reserved by the system = Use is strictly prohibited |
| 7010 | | A cause of drive-source cutoff is present (including a condition waiting for a cutoff cancellation input). |
| 7011 | | A latch signal indicating that a cause of all-operation cancellation is present.<br>(This latch signal is used to recognize a cause of 1-shot reset. Latch cancellation: 7300-ON) |
| 7012 | | A cause of all-operation pause is present (including a condition waiting for the restart switch to be pressed).<br>(Effective only in the auto operation recognition mode) |
| 7013 | | A cause of all-servo-axis interlock is present (cause of all-operation pause + cause of interlock input port) |
| 7014 to 7042 | | Reserved by the system = Use is strictly prohibited |
| 7043 | | Axis 1 home return completion (Main application V2.00 or later) |
| 7044 | | Axis 2 home return completion (Main application V2.00 or later) |
| 7045 | | Axis 3 home return completion (Main application V2.00 or later) |
| 7046 | | Axis 4 home return completion (Main application V2.00 or later) |
| 7047 to 7050 | | Reserved by the system = Use is strictly prohibited |
| 7051 | | Axis 1 servo-ON completion (Main application V2.10 or later) |
| 7052 | | Axis 2 servo-ON completion (Main application V2.10 or later) |
| 7053 | | Axis 3 servo-ON completion (Main application V2.10 or later) |
| 7054 | | Axis 4 servo-ON completion (Main application V2.10 or later) |
| 7055 to 7070 | | For future expansion = Use is strictly prohibited |
| 7071 | | In the AUTO mode |
| 7072 | | During auto operation |
| 7073 to 7074 | | For future expansion = Use is strictly prohibited |
| 7075 to 7100 | | During Tracking Conveyor Speed Drop Detection |
| 7101 | | Program No. 01 is being executed (or paused). |
| ~ | | ~ |
| 7228 | | Program No. 128 is being executed (or paused). |
| 7229 to 7299 | | For future expansion = Use is strictly prohibited |
| 7300 | Output | A latch cancellation signal is output to cancel the latch signal indicating a cause of all-operation cancellation (7011). (Unlatched only when the cause of operation cancellation is no longer present.)<br>(7300 is turned OFF after latch cancellation is attempted.) |
| 7301 to 7380 | | (For future expansion = Use is strictly prohibited) |
| 7381 to 7399 | | (Reserved by the system = Use is strictly prohibited) |
| 7400 | | (For future expansion = Use is strictly prohibited) |
| 7401 | Input | Program No. 129 is being executed (or paused). |
| ~ | | ~ |
| 7527 | | Program No. 255 is being executed (or paused). |
| 7528 to 7599 | Output | (For future expansion = Use is strictly prohibited) |

# 3. Program
## 3.1 Position Table and Program Format

To run the robot,
- Program
- Position data
  (information of positions for the robot to move)
  it is necessary to create 2 types of data as shown below and input them to the controller with using PC.



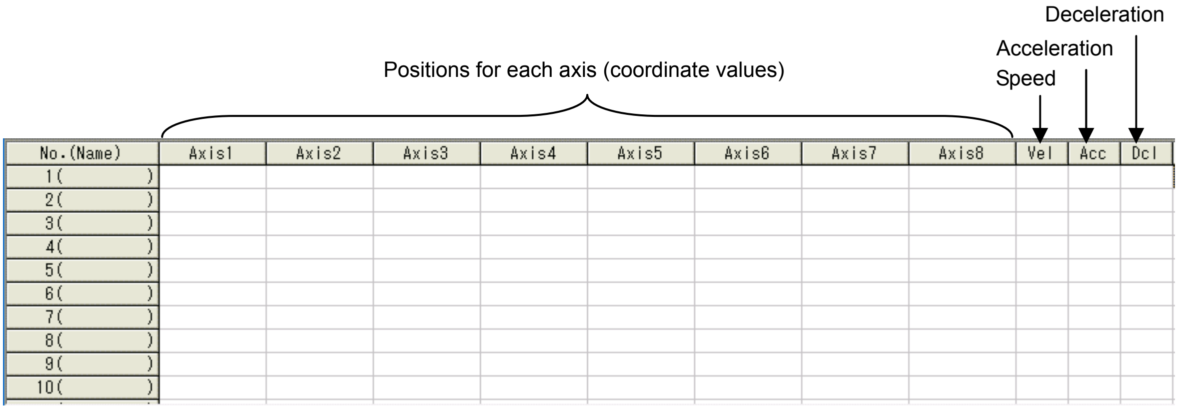

### 3.1.1 Position Table

The position data is to be input in the position table provided in the PC software.
(Note) • The edit window differs depending on the type of robot.
• The total number of settable positions differ depending on the controller

| Controller | Position Total Number |
|---|---|
| XSEL-P/Q/PCT/QCT, PX/QX | 20000 |
| XSEL-R/S/RX/SX/RXD/SXD | 53332 (1-axis specification)<br>40000 (2-axis specification)<br>32000 (3-axis specification)<br>26666 (4-axis specification)<br>22856 (5-axis specification)<br>20000 (6-axis specification)<br>17776 (7-axis specification)<br>16000 (8-axis specification) |
| XSEL-RA/SA/RAX/SAX/<br>RAXD/SAXD | 55000 (1-axis specification)<br>47142 (2-axis specification)<br>41250 (3-axis specification)<br>36666 (4-axis specification)<br>33000 (5-axis specification)<br>30000 (6-axis specification)<br>27500 (7-axis specification)<br>25384 (8-axis specification) |
| XSEL-J/K/KE/KT/KET<br>JX/KX/KETX | 3000 |
| SSEL | 20000 |
| ASEL/PSEL | 1500 |
| TT | 3000 |
| TTA | 30000 |
| MSEL | 30000 |

[1] Single/rectangular axes, TT robots

Set positions (coordinate values), speeds, accelerations and decelerations in the position table and store the table in the controller.




- Axes 1 to 8: Position (coordinate value)
  Set the positions (coordinates) for all the connected axes (8 axes at maximum).
  The setting range varies depending on the actuator.
  The maximum range is from -99999.999 to 99999.999.

| | Axis 1 | Axis 2 | Axis 3 | Axis 4 | Axis 5 | Axis 6 | Axis 7 | Axis 8 |
|---|---|---|---|---|---|---|---|---|
| Single/rectangular axes | Axis 1 | Axis 2 | Axis 3 | Axis 4 | Axis 5 | Axis 6 | Axis 7 | Axis 8 |

- Comment Column (shown on the right side of the position data)
  Applicable models : XSEL-RA/SA/RAX/SAX/RAXD/SAXD
  XSEL-R/S/RX/SX/RXD/SXD
  - Input is available up to 32 words with half width characters and 16 with full width at the maximum.
  - Comment can be added to Positions No. 1 to 10000.
  - Comment needs to be written to flash ROM after the data is transferred.

(Note 1) When having the software reset conducted or the power turned OFF without writing the position data to the flash ROM after a change is made to a comment, 22B "Position Data Comment Lost Error" will occur and the comment that a change was made will be deleted.
It is only the comment which is lost, and the program operation is available.

(Note 2) Comment is applicable also for PCLR (Position Data Clear) Command and PCPY (Position Data Copy) Command in SEL Program.
Therefore, in case that the position data with a comment being set is cleared by PCLR Command, or that a copy is made to the position with a comment being set in the area to copy from or area to copy to with PCPY Command, if the software reset is conducted or the power is turned OFF without the position data being written to the flash ROM, 22B "Position Data Comment Lost Error" will occur.
Please understand this well when using PCLR or PCPY.

- Vel (speed), Acc (acceleration), Dcl (deceleration)
  If Vel (speed), Acc (acceleration) and Dcl (deceleration) are set in the position data table, the values set in the position data table are given priority over the corresponding data set in the program.
  To make the Vel (speed), Acc (acceleration) and Dcl (deceleration) settings in the program effective, leave these fields blank.

The effective values of speed and acceleration are determined based on the priorities specified below.

| Priority | Speed | Acceleration (Deceleration) |
|---|---|---|
| 1 | Value set in the position data table | Value set in the position data table |
| 2 | Value set in the VEL command | Value set in the ACC (DCL) command |
| 3 | | All-axis parameter No. 11, "Default acceleration"<br>(All-axis parameter No. 12, "Default deceleration") |

The setting ranges vary depending on the actuator.

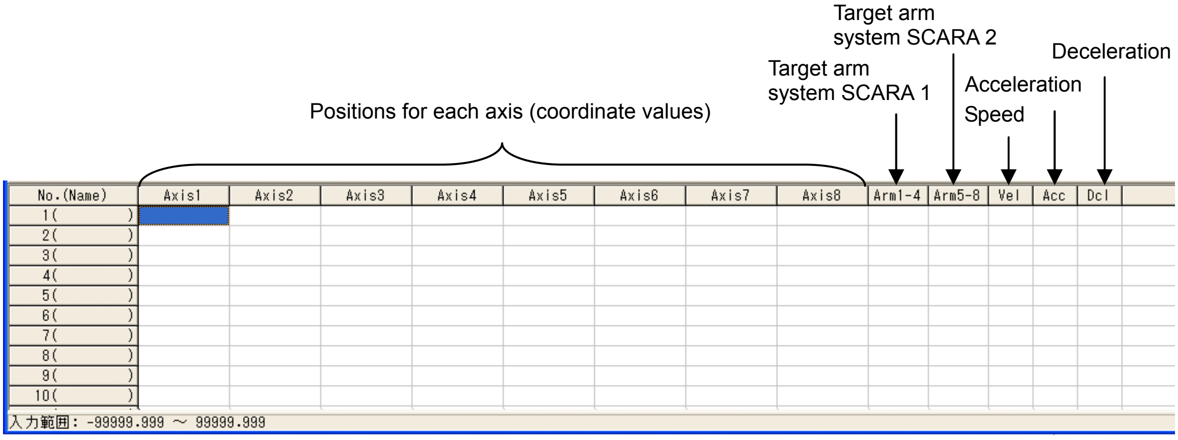
- Speed — 1 to the value of all-axis common parameter No. 21, “Maximum operating speed”
- Acceleration — 0.01 to the value of all-axis common parameter No. 22, “Maximum acceleration”
- Deceleration — 0.01 to the value of all-axis common parameter No. 23, “Maximum deceleration”

With rotational axes, values in mm are handled in degrees.
If Axis-specific parameter No. 1, “Axis operation type” is set to 1 (Rotational movement axis (angle control)), all values indicated in mm (including parameters, etc.) are converted to angles (in degrees).
These angles (in degrees) indicate angles (in degrees) of a rotating body at the end, as long as the gear ratio parameters (Axis-specific parameter No. 50, 51) are set correctly.

Example) Distance 1mm → 1deg
Speed 1mm/sec → 1deg/sec
Acceleration/deceleration 1G = $9807mm/sec^2$
→ $9807deg/sec^2$

- For XSEL-RA/SA/RAX/SAX/RAXD/SAXD, TTA and MSEL, there is the output operation data. Parameter select for TTA and MSEL) [Refer to 5.5 Position Output Operation Features]

[2] SCARA robots
Set positions (coordinate values), target arm system indications, speeds, accelerations and decelerations in the position table and store the table in the controller.




- Axes 1 to 8: Position (coordinate value)
  Set the positions (coordinates) for all the connected SCARA robots (8 axes at maximum).
  The setting range varies depending on the actuator.
  The maximum range is from -99999.999 to 99999.999.

| SCARA robots 1 | | | | SCARA robots 2 | | | |
|---|---|---|---|---|---|---|---|
| Axis 1 | Axis 2 | Axis 3 | Axis 4 | Axis 5 | Axis 6 | Axis 7 | Axis 8 |
| X-axis | Y-axis | Z-axis | R-axis | X-axis | Y-axis | Z-axis | R-axis |

- Arm1-4, Arm5-8: Target arm system indications (for XSEL-RX/SX/RXD/SXD, MSEL-PCX/PGX (Arm1-4) only)
  Indicate the arm system at PTP movement. (R: right arm system, L: left arm system)
  The indications become valid when:
  1) an operation is made with "Move" in the position data edit window or "Continuous Move" button, or
  2) when SEL program PTP movement command is executed using the position data

(Note 1) Unless otherwise the target arm system is indicated, the operation will be made with the current arm system.

(Note 2) The priority is put in the order as shown below to the indication of the target arm systems when PTP movement command is executed by SEL program.
1) Position data settings
2) SEL command (Settings for PTPR, PTPL, PTPD and PTPE)

(Note 3) Error No. B4D "Arm System Setting Error" would be issued if the current arm system and the target arm system are different at CP movement.

- There is a comment column. (XSEL-RA/SA/RAX/SAX/RAXD/SAXD)

- Vel (speed), Acc (acceleration), Dcl (deceleration)
  If Vel (speed), Acc (acceleration) and Dcl (deceleration) are set in the position data table, the values set in the position data table are given priority over the corresponding data set in the program.
  To make the Vel (speed), Acc (acceleration) and Dcl (deceleration) settings in the program effective, leave these fields blank.
  Only for CP operation (*1) the speeds, accelerations and decelerations set in the position table become effective.
  (*1) CP operation is a type of operation in which the axes interpolates with one another to move to the target position. CP operation may be performed in the form of linear movement, arc movement, etc. If the axes do not interpolate with one another, such operation is called "PTP operation".

The effective values of speed and acceleration are determined based on the priorities specified below.

| Priority | Speed | Acceleration (Deceleration) |
|---|---|---|
| 1 | Value set in the position data table | Value set in the position data table |
| 2 | Value set in the VEL command | Value set in the ACC (DCL) command |
| 3 | | All-axis parameter No. 11, "Default acceleration" (All-axis parameter No. 12, "Default deceleration") |

The setting ranges vary depending on the actuator.

- Speed 1 to the value of all-axis common parameter No. 21, "Maximum operating speed"
- Acceleration 0.01 to the value of all-axis common parameter No. 22, "Maximum acceleration"
- Deceleration 0.01 to the value of all-axis common parameter No. 23, "Maximum deceleration"

With rotational axes, values in mm are handled in degrees.
If Axis-specific parameter No. 1, "Axis operation type" is set to 1 (Rotational movement axis (angle control)), all values indicated in mm (including parameters, etc.) are converted to angles (in degrees).
These angles (in degrees) indicate angles (in degrees) of a rotating body at the end, as long as the gear ratio parameters (Axis-specific parameter No. 50, 51) are set correctly.

Example) Distance 1mm → 1deg
Speed 1mm/sec → 1deg/sec
Acceleration/deceleration 1G = 9807mm/sec$^2$
→ 9807deg/sec$^2$

- For XSEL-RA/SA/RAX/SAX/RAXD/SAXD, MSEL-PCX/PGX, there is the output operation data. Parameter select for TTA and MSEL) [Refer to 5.5 Position Output Operation Features]

## 3.2 Program

Create a program using the “SEL Language” which is a proprietary language by IAI.

(Note) The number of programs and total number steps vary depending on the controller.

| Controller | Number of programs | Total number of program steps |
|---|---|---|
| XSEL-P/Q/PCT/QCT/PX/QX/ R/S/RX/SX/RXD/SXD | 128 | 9999 |
| XSEL-RA/SA/RAX/SAX/ RAXD/SAXD | 255 | 20000 |
| XSEL-J/K/KE/KT/KET JX/KX/KETX | 64 | 6000 |
| SSEL | 128 | 9999 |
| ASEL/PSEL | 64 | 2000 |
| TT | 64 | 6000 |
| TTA | 255 | 9999 |
| MSEL | 255 | 9999 |

## 3.3 Program Format

### [1] Program Edit

The program is to be input to the program edit window in the PC software.
The created program is to be transferred to the controller to be activated.

SEL language is translated by a step number (1 line) to another for the operation. Thus, it is unnecessary to compile (translate into the computer language).

| No. | B | E | N | Cnd | Cmnd | Operand1 | Operand2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|---|
| 3 | | | | | VEL | 100 | | | |
| 4 | | | | | ACC | 0.3 | | | |
| 5 | | | | | TAG | 1 | | | |
| 6 | | | | | EXSR | 5 | | | |
| 7 | | | | | MOVP | 10 | | | |
| 8 | | | | | MOVP | 11 | | | |
| 9 | | | | | TIMW | 0.3 | | | |
| 10 | | | | | EXSR | 5 | | | |
| 11 | | | | | MOVP | 15 | | | |
| 12 | | | | | EXSR | 6 | | | |
| 13 | | | | | TIMW | 0.2 | | | |
| 14 | | | | | MOVP | 16 | | | |
| 15 | | | | | VEL | 300 | | | |
| 16 | | | | | EXSR | 1 | | | |
| 17 | | | | | MOVP | 20 | | | |

Step No.
Expansion Condition
Input Condition
Input Condition
Command and Declaration
Operand 1
Operand 2
Output

The program edit window is created in a program format (table format) and you are only to input the information for position, command, etc. to the appropriate areas.
In the following, explains about the program format.

(Note) For XSEL-RA/SA/RAX/SAX/RAXD/SAXD controllers, binary and hexadecimal numbers can be input to Operand1 (Operation 1) and Operand2 (Operation 2).

<Expression>
Binary: Add prefix &B
Decimal: No prefix
Hexadecimal: Add prefix &H

<Example>

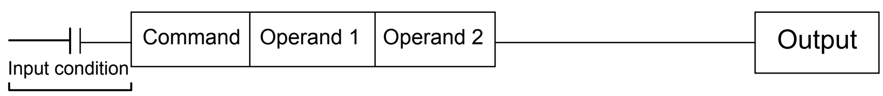
```
AND    200    &B00111000    ; Binary number
AND    200    56            ; Decimal number
AND    200    &H00000038    ; Hexadecimal number
```

<Caution>
The input range of a number is 99999999 to -9999999 of decimal system regardless of the number base.
(In hexadecimal system, 05F5E0FF to FF676981)

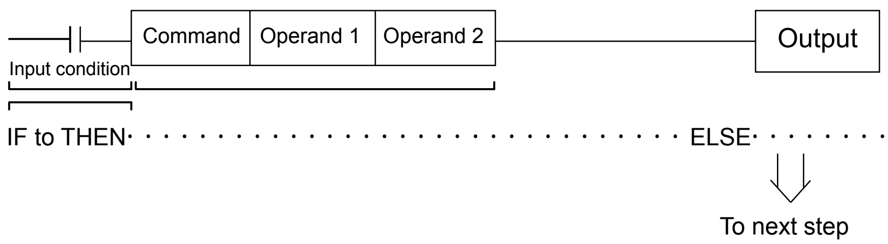
[2] Program Format

| Extension condition (AND, OR) | Input condition (I/O, flag) | Command, declaration | | | Output (Output port, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand1 | Operand2 | Pst |

The above is illustrated as follows in a ladder diagram.

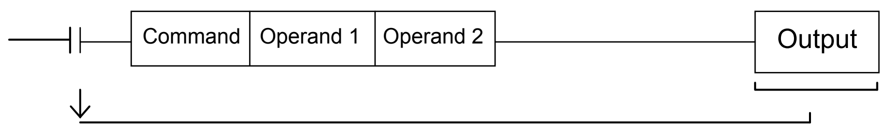



(1) The conditions in front of the command are equivalent to "IF to THEN" for the BASIC language.




1) The command will be executed if the input conditions meet the requirement. If there is an indication for the output, it turns the output port on, and if the input conditions do not meet the requirement, it moves to the next step.
2) If no condition is set, the command is executed unconditionally.
3) To use the condition in reverse (using the so-called contact b logic), add "N" (NOT) to the condition.
4) You can use an input port, output port or flag for the input condition.
5) Operands 1 and 2, and the output, can be specified indirectly.

(2) Operation of the output, specified after the command and operands 1 and 2, is explained below.




1) In the case of an actuator operation control command, etc., the output turns OFF the moment the command execution is started and turns ON when the execution is completed. In the case of a calculation command, etc., the output turns ON when the result becomes a certain value and turns OFF with other values.
2) You can use an output port or flag for the output.

[Application] Extended condition

You can combine extended conditions in a complex manner using the AND gate and OR gate.

(Example)

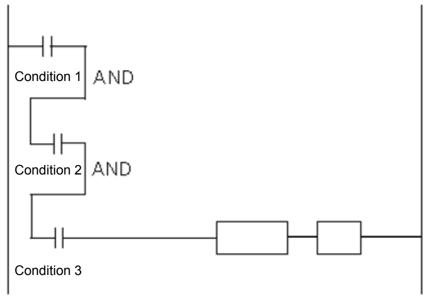
[Extension by AND]

Combination of A (AND) and A (AND)

(SEL language)

| Extension condition | Input condition | Command | | | Output |
|---|---|---|---|---|---|
| | | Command | Operand 1 | Operand 2 | |
| | Condition 1 | | | | |
| A | Condition 2 | | | | |
| A | Condition 3 | Command | Operand 1 | Operand 2 | |
| | | | | | |
| | | | | | |

(Ladder diagram)



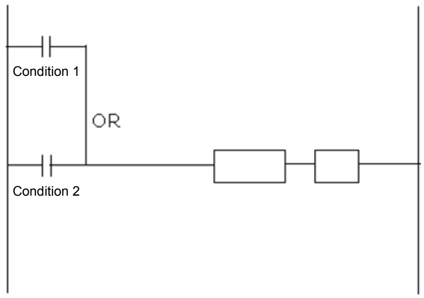

[Extension by OR]

Combination of O (OR) and O (OR)

(SEL language)

| Extension condition | Input condition | Command | | | Output |
|---|---|---|---|---|---|
| | | Command | Operand 1 | Operand 2 | |
| | Condition 1 | | | | |
| ○ | Condition 2 | Command | Operand 1 | Operand 2 | |
| | | | | | |
| | | | | | |
| | | | | | |

(Ladder diagram)



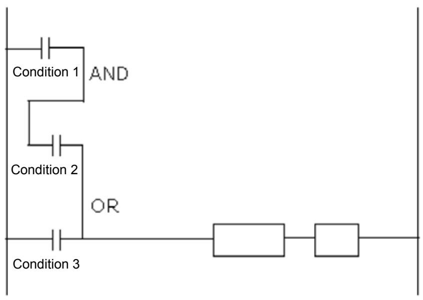

[Extension by AND and OR]

Combination of A (AND) and O (OR)

(SEL language)

| Extension condition | Input condition | Command | | | Output |
|---|---|---|---|---|---|
| | | Command | Operand 1 | Operand 2 | |
| | Condition 1 | | | | |
| A | Condition 2 | | | | |
| ○ | Condition 3 | Command | Operand 1 | Operand 2 | |
| | | | | | |
| | | | | | |

(Ladder diagram)



## 3.4 Relationship of program and position table

In the case of a movement command such as MOVL, set a position number in operand 1. Some commands such as ARCH (arch motion) require a position number to be set in operand 2, as well.
The position corresponding to the position number set in the position table is referenced and the actuator moves to the applicable position.

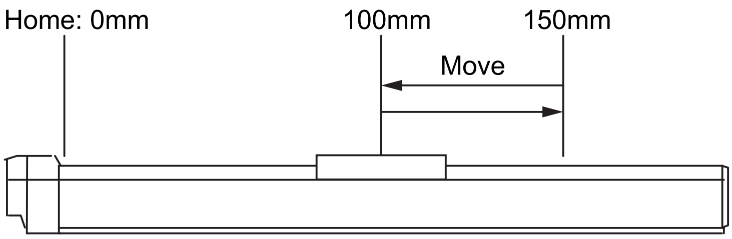
Program format

| No. | B | E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | HOME | 1 | | | 原点復帰 |
| 2 | | | | | VEL | 100 | | | 速度設定100mm/sec |
| 3 | | | | | TAG | 1 | | | |
| 4 | | | | | MOVL | 1 | | | P1移動 |
| 5 | | | | | MOVL | 2 | | | P2移動 |
| 6 | | | | | GOTO | 1 | | | |
| 7 | | | | | | | | | |

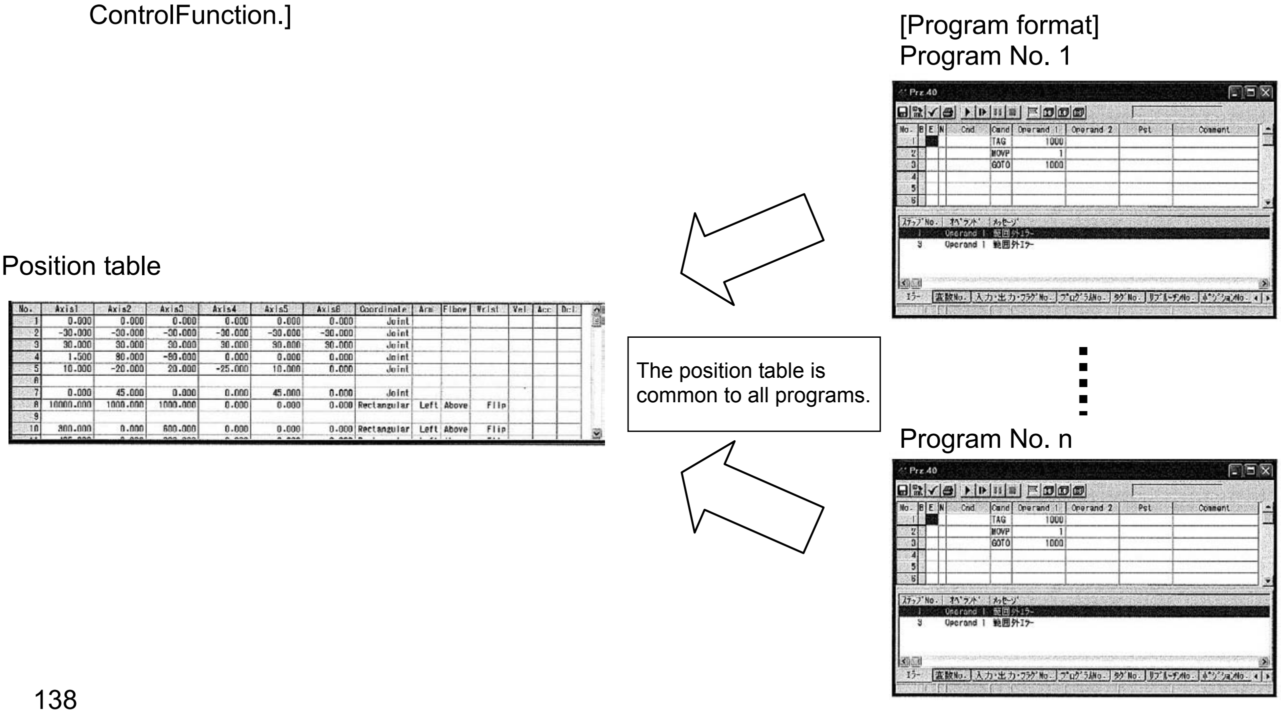
Position table

| No. | Axis1 |
|---|---|
| 1 | 100.000 |
| 2 | 150.000 |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

In the above example, the actuator moves to the positions at 100mm corresponding to position No. 1 and 150mm corresponding to position No. 2.




The position table is a single table that can be referenced from all programs. In the example below, the standard position table is used. A different table is used if the controller has a gateway function.
(Note) For RC Gateway Function and Extension Motion Control Function, a different position table is to be used.
[Refer to the Instruction Manual for XSEL Controller P/Q/PX/QX RC Gateway Function.]
[Refer to the Instruction Manual for XSEL-RA/SA/RAX/SAX/RAXD/SAXD Extension Motion ControlFunction.]



## 3.5 Basic Stage (Program creation and position table creation)

In this section, explains how to create a program for the basic operation patterns.

### 3.5.1 Home Return and Home Return Completion Signal

[1] Description

Output a signal to confirm completion of homing (incremental specification).
With the controller, a home return completion signal can be output using an I/O parameter.
However, the following explains how to output a home return completion signal within a program using a general-purpose output.
Once turned ON, a general-purpose output will remain ON even after the current program ends or other program is started. (It will turn OFF upon emergency stop, etc., but the ON status can be maintained using an I/O parameter (I/O parameter No. 70 and 71).)
(Note) SCARA robots do not require home return operation.

[2] Example of Use

a. Output a home return completion signal.

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
|---|---|---|---|---|---|---|
| | | | HOME | 11 | | |
| | | | BTON | 303 | | |

Execute homing. (1st and 2nd axes)
Output 303 is turned ON when home-return operation is complete

b. Use a home return completion signal to make sure the actuator will not perform homing if it has already been performed.

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
|---|---|---|---|---|---|---|
| | N | 303 | HOME | 11 | | |
| | | | BTON | 303 | | |

If output 303 is OFF (NOT);

Execute homing if output 303 is OFF.
Home-return complete output
(turn Output 303 ON)

c. Use the output field instead of a BTON command.

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
|---|---|---|---|---|---|---|
| | N | 303 | HOME | 11 | | 303 |

Output section

The same process as Example b. is proceeded.

[3] Reference

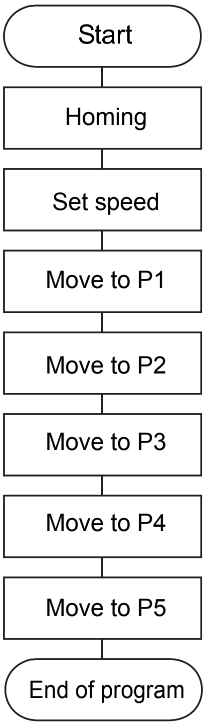
Output port No. 304 can be used as a home return completion output (dedicated output) by setting I/O parameter No. 50 to “2”.

### 3.5.2 Positioning Operation (Moving position)

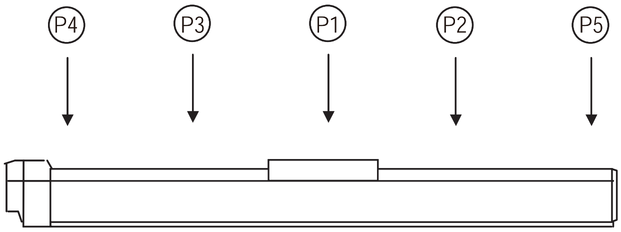
[1] Description
Move the actuator to positions 1 through 5 at a speed of 100mm/sec after homing.

[2] Example of Use

Flowchart




- Homing must be performed and a speed must be set, before the actuator can be operated.
- The actuator moves to the position data coordinates specified by the respective move commands.
- With the absolute specification, homing (HOME command) is not required.




Program (Example)

| No. | B | E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | HOME | 1 | | | 1軸原点復帰 |
| 2 | | | | | VEL | 100 | | | 速度設定100mm/sec |
| 3 | | | | | MOVL | 1 | | | ポジションデータNo.1に移動 |
| 4 | | | | | MOVL | 2 | | | ポジションデータNo.2に移動 |
| 5 | | | | | MOVL | 3 | | | ポジションデータNo.3に移動 |
| 6 | | | | | MOVL | 4 | | | ポジションデータNo.4に移動 |
| 7 | | | | | MOVL | 5 | | | ポジションデータNo.5に移動 |
| 8 | | | | | EXIT | | | | ﾌﾟﾛｸﾞﾗﾑ終了 |
| 9 | | | | | | | | | |

Position data (Example)

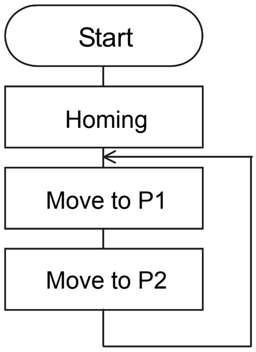
| No. | Axis1 |
|---|---|
| 1 | 100.000 |
| 2 | 150.000 |
| 3 | 50.000 |
| 4 | 0.000 |
| 5 | 200.000 |
| 6 | |
| 7 | |
| 8 | |
| 9 | |

## 3.5.3 Moving Back and Forth between Two Points

[1] Description
Moves back and forth between two points.

[2] Example of Use

Flowchart




- The actuator moves back and forth between P1 and P2 indefinitely.
- Use of only 1 axis is assumed.
- Enter TAG in the first of the steps to be repeated, and enter GOTO in the last of the steps to be repeated.

Program (Example)

| No. | B | E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | HOME | 1 | | | 原点復帰 |
| 2 | | | | | VEL | 100 | | | 速度設定100mm/sec |
| 3 | | | | | TAG | 1 | | | |
| 4 | | | | | MOVL | 1 | | | ポジションデータNo.1に移動 |
| 5 | | | | | MOVL | 2 | | | ポジションデータNo.2に移動 |
| 6 | | | | | GOTO | 1 | | | TAG1 にジャンプ |
| 7 | | | | | | | | | |

Position data (Example)

| No. | Axis1 |
|---|---|
| 1 | 100.000 |
| 2 | 150.000 |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

### 3.5.4 Repeated Operation

[1] Description

Use GOTO and TAG commands to repeat the same operation within the program or to jump to a desired step if a condition is satisfied. A TAG command can be written in a step either before or after a GOTO command.

[2] Example of Use

Example 1

Repeat the same operation. (Move to Position 1 → to position 2 → to position 1 …)

```
          :
          TAG    1
Repeated. MOVP   1   } Steps to be repeated
          MOVP   2
          GOTO   1
```

Example 2

Skip steps.

```
          :
          GOTO   2
Jump.     MOVP   1   } Steps to be ignored
          MOVP   2
          TAG    2
```

## 3.5.5 PATH Operation (Continuous operation among multiple positions)

[1] Description

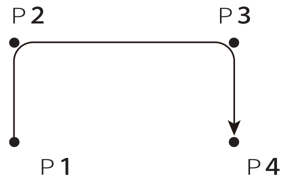
This function moves the robot continuously among 4 arbitrary points. (PATH movement)

[2] Example of Use

The actuator moves along the path shown at right, without stopping at P2 and P3.
Compared with MOVP and MOVL, this command does not require the actuator to position exactly at P2 and P3, and thus the movement tact time can be reduced.
Assume the following command is executed when the actuator is stopped at P1:

PATH 2 4

The actuator will move from P1 to P4 by passing points near P2 and P3.



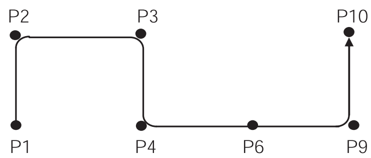

Even if “PATH 2 3” and “PATH 4 4” are input successively, the actuator will still move in the same way as when “PATH 2 4” is input.

If “PATH 4 1” is executed while the actuator is stopped at P4, the actuator will move along the same path in the opposite direction (P4 → P3 → P2 → P1).

It is possible to move through discontinuous positions or move continuously by passing the same position.

| | | | |
|---|---|---|---|
| PATH | 1 | 4 | |
| PATH | 6 | 6 | discontinuous position |
| PATH | 9 | 10 | |

As shown above, specify the number corresponding to the discontinuous position, or No. 6, for both the start position number and end position number in the PATH command. The actuator moves continuously in the sequence of position Nos. P1 → P2 → P3 → P4 → P6 → P9 → P10.




[3] Example of Use

Refer to the page for “PATH” Command in [12] Actuator Control Command for each command language for the caution in use.

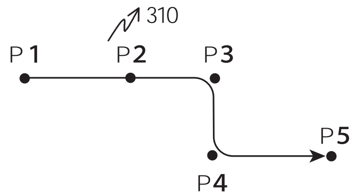
## 3.5.6 External Signal Output during Path Movement

[1] Description
Output signals while the actuator is moving with a PATH command.

[2] Example of Use
Before executing a PATH command, declare a POTP command to specify signal output during movement.
If a given output or global flag is specified in the output field of the PATH command, the output or flag specified in the output field will turn ON as the actuator approaches, via path movement, the position specified in the PATH command.

Example 1

The actuator moves from P1 to P5 along the positions shown at right, without stopping. As the actuator approaches P2, output port 310 turns ON.

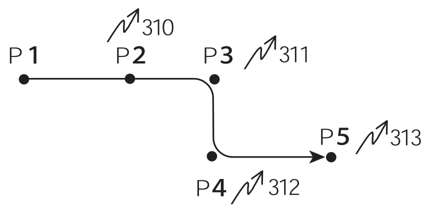



| Cmnd | Operand 1 | Operand 2 | Pst | |
|---|---|---|---|---|
| VEL | 100 | | | |
| POTP | 1 | | | ← A declaration command to specify signal output during path movement. |
| PATH | 1 | 1 | | ← 310 turns ON when the actuator approaches P2 specified in this step. |
| PATH | 2 | 2 | 310 | |
| PATH | 3 | 5 | | |

Example 2

Outputs 310 to 313 can be turned ON sequentially at the respective points of P2 to P5.

| Cmnd | Operand 1 | Operand 2 | Pst | |
|---|---|---|---|---|
| VEL | 100 | | | |
| POTP | 1 | | | ← A declaration command to specify signal output during path movement. |
| PATH | 1 | 1 | | ← 310 to 313 turn ON sequentially at P2 to P5 specified in this step. |
| PATH | 2 | 5 | 310 | |




(Note) This command is able only to output and to turn the flag ON. The output or flag that was turned ON during path operation must be turned OFF (using a BTOF command) after the operation is completed.

[3] Example of Use
Refer to the page for “PATH” Command in [12] Actuator Control Command for each command language for the caution in use.

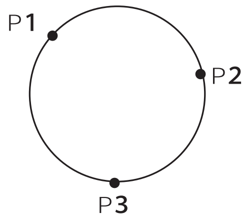
### 3.5.7 Circle/Arc Operation

[1] Description
The actuator moves along a two-dimensional circle or arc.

[2] Example of Use
To specify a circle, specify three points the actuator will pass. To specify an arc, specify the starting point, passing point and end point.

Example 1

Circle



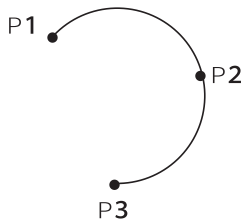

- Specify "CIR2 2 3" after the actuator has moved to P1.
- If "CIR2 2 3" is specified in the figure shown at left, the actuator will move along this circle clockwise.

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
|---|---|---|---|---|---|---|
| | | | VEL | 100 | | |
| | | | MOVP | 1 | | |
| | | | CIR2 | 2 | 3 | |

- To cause the actuator to move counterclockwise, specify "CIR2 3 2".

Example 2

Arc




- Specify "ARC2 2 3" after the actuator has moved to P1.

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
|---|---|---|---|---|---|---|
| | | | VEL | 100 | | |
| | | | MOVP | 1 | | |
| | | | ARC2 | 2 | 3 | |

[3] Reference
1) Some circle and arc commands can be executed not only two-dimensionally (between two actuator axes) but also three dimensionally (among three actuator axes).

CIRS ·········· Move along circle three-dimensionally
ARCS ·········· Move along arc three-dimensionally

2) Refer to the page for "CIR2, ARC2, CIRS, ARCS" Command in [12] Actuator Control Command for each command language for the caution in use.

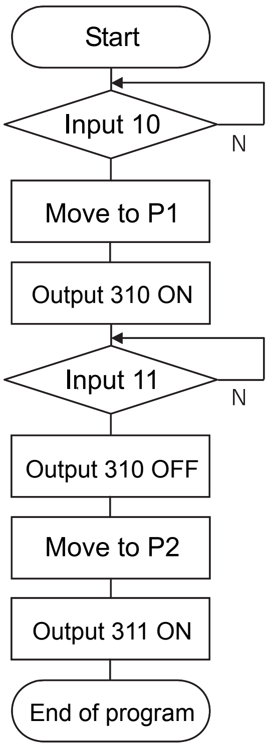
## 3.5.8 Axis Movement by External Signals and Output of Completion Signal to External Device

[1] Description

This is a function to enable the axes movement with an external signal input and to output the complete signal to an external device.

[2] Example of Use

<u>Flowchart</u>




Wait for the input port (external signal) 10 to turn ON, and then move to P1 (Position Data No. 1).
Wait for the input port (external signal) 11 to turn ON, and then move to P2 (Position Data No. 2).
The movement complete signal for P1 is output to the output port 310, and P2 complete signal to port 311.

<u>Program (Example)</u>

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|
| | | | VEL | 100 | | | 速度100mm/sec設定 |
| | | | WTON | 10 | | | 入力10 ON待ち |
| | | | MOVP | 1 | | | P1移動 |
| | | | BTON | 310 | | | 出力310 ON |
| | | | WTON | 11 | | | 入力11 ON待ち |
| | | | BTOF | 310 | | | 出力310 OFF |
| | | | MOVP | 2 | | | P2移動 |
| | | | BTON | 311 | | | 出力311 ON |
| | | | EXIT | | | | ﾌﾟﾛｸﾞﾗﾑ終了 |

### 3.5.9 Changing the Moving Speed

[1] Description
Change the moving speed.

[2] Example of Use
The speed can be set using the following two methods:
a: Use a VEL command within the program
b: Use a speed setting in the position table

Program (Example)

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
|---|---|---|---|---|---|---|
| | | | MOVP | 1 | | P1移動 |
| | | | VEL | 1000 | | 速度1000mm/sec設定 |
| | | | MOVP | 2 | | P2移動 |
| | | | MOVP | 3 | | P3移動 |
| | | | VEL | 50 | | 速度50mm/sec設定 |
| | | | MOVP | 4 | | P4移動 |

Position data (Example)

| No. | Axis1 | Vel | Acc | Dcl |
|---|---|---|---|---|
| 1 | 100.000 | 100 | | |
| 2 | 200.000 | 500 | | |
| 3 | 300.000 | | | |
| 4 | 400.000 | | | |

Each Position (Position Data No. 1 (P1) to No. 4 (P4))

Moving speeds in the above program
Position at 100mm (P1) … The actuator moves at 100mm/sec.
Position at 200mm (P2) … The actuator moves at 500mm/sec.
Position at 300mm (P3) … The actuator moves at 1000mm/sec.
Position at 400mm (P4) … The actuator moves at 50mm/sec.

If a speed is specified in the position data table, this speed takes precedence over the speed specified in the application program, as shown above.

### 3.5.10 Speed Setting Change during PATH (Continuous) Operation

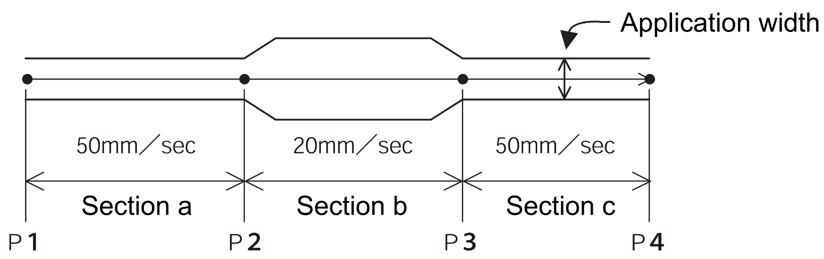
[1] Description

You can change the speed of the actuator without stopping it, by using a PATH command and VEL fields of the position table.

For example, this command is useful in a paint dispensing application where the application volume changes in the middle.

[2] Example of Use

The actuator moves through linear sections a, b and c at 50mm/sec, 20mm/sec and 50mm/sec, respectively, without stopping (PATH operation).




Position data (Example)

| No. | Axis1 | Vel | Acc | Dcl |
|---|---|---|---|---|
| 1 | 0.000 | 50 | | |
| 2 | 100.000 | 50 | | |
| 3 | 200.000 | 20 | | |
| 4 | 300.000 | 50 | | |

Program (Example)

"PATH 1 4" is the only movement command required.

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
|---|---|---|---|---|---|---|
| | | | PATH | 1 | 4 | |

[3] Reference

The speed can also be changed from other program using a CHVL (speed change) command.

### 3.5.11 Variables and Flags [Global/Local]

[1] Description

The internal variables and flags used in the SEL language are classified into local and global types.
The data range used commonly by all programs is called the global range, while the data range used only by each program is called the local range. When multi-tasking programs are run simultaneously, the global range must be used to synchronize the programs and allow cross-referencing of variables among the programs.

[2] Example of Use

Program handshake

Program A

| Cmnd | Operand 1 | |
|---|---|---|
| MOVL | 1 | |
| BTON | 600 | Turn ON Output 600 → |
| WTON | 601 | Waiting for Output 601 to turn ON |
| MOVL | 3 | |

Program B

| Cmnd | Operand 1 | |
|---|---|---|
| WTON | 600 | Waiting for Output 600 to turn ON |
| MOVL | 2 | |
| BTON | 601 | Turn ON Output 601 |

Use of global flags with the above two programs permits handshake between the programs, and the actuator moves per "MOVL 1" in program A, moves per "MOVL 2" in program B, and then move per "MOVL 3" in program A, for example.

**Backup in Battery**

The XSEL controller (Note 1) has a built-in battery for retaining variables and flags used in the programs. The data is retained in the global domain for both variables and flags even if the power is turned OFF.
The variables and flags in the local range are cleared when the program is started (the variables are reset to "0", while the flags turn OFF).

Note 1: XSEL-RA/SA/RAX/SAX/RAXD/SAXD, XSEL-R/S/RX/SX/RXD/SXD, MSEL and TTA do not have a system memory backup battery since they possess the global domain in the non-volatile memory.

The system-memory backup battery is optional for ASEL, PSEL and SSEL controllers.
No system-memory backup battery is available for TT robots.
If the system memory is not backed up with a battery, global areas will be cleared once the power is cut off.

### 3.5.12 How to Use Subroutines

[1] Description

A subroutine is a group of steps that are called and executed several times within a program. Subroutines are used to reduce the number of program steps and make the program easy to read. Up to 99 subroutines can be used in one program. Up to 15 subroutine calls can be nested.

[2] Example of Use

Declare/call subroutines using the following commands:

EXSR: Call a subroutine
BGSR: Declare the start of a subroutine (start of a group of steps)
EDSR: Declare the end of a subroutine (end of a group of steps)

| Cmnd | Operand 1 |
|---|---|
| VEL | 100 |
| MOVL | 1 |
| BTON | 310 |
| WTON | 20 |
| BTOF | 310 |
| MOVL | 2 |
| BTON | 310 |
| WTON | 20 |
| BTOF | 310 |
| MOVL | 3 |
| BTON | 310 |
| WTON | 20 |
| BTOF | 310 |
| | |
| EXIT | |

The same tasks are consolidated into a single location.

| Cmnd | Operand 1 |
|---|---|
| VEL | 100 |
| MOVL | 1 |
| EXSR | 1 |
| MOVL | 2 |
| EXSR | 1 |
| MOVL | 3 |
| EXSR | 1 |
| | |
| EXIT | |
| BGSR | 1 |
| BTON | 310 |
| WTON | 20 |
| BTOF | 310 |
| EDSR | |

Subroutine

[3] Note

Jumping from within a subroutine to a TAG position outside the subroutine using a GOTO command is prohibited.
Allocate a subroutine (BGSR ~ EDSR) to the end of each program.
(While executing a program, if stepping to BGSR Command with an action other than a jump by EXSR Command, the next step to be executed will go back to the top step of the program. Therefore, steps after a subroutine should not basically be executed.)

## 3.5.13 Pausing the Operation

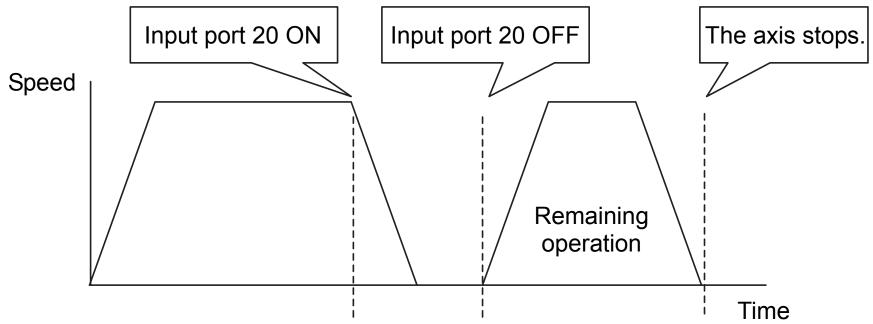
[1] Description

Use a declaration command HOLD to pause the moving axis temporarily via external input.

[2] Example of Use

A pause interruption operation can be executed to a moving axis (to decelerate the axis to a stop) by declaring a HOLD command within the program.
While HOLD is input, the actuator pauses (decelerates to a stop, if currently moving) against all moving commands in the same program.

HOLD 20 A declaration to execute pause if general-purpose input 20 turns ON.




Application

You can specify a global flag, instead of an input port, in Operand 1 of the HOLD command.
Use of a global flag allows the actuator to be paused from other program.
The input signal pattern and stop action can be selected using Operand 2.

0 = Contact a (Decelerates to a stop) ⇒ Same as when Operand 2 is not specified.
1 = Contact b (Decelerates to a stop)
2 = Contact b (Decelerates to a stop, and then servo OFF
⇒ The drive power is not cut off.)

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|
| | | | HOLD | 20 | 2 | | SVOF入力20 B接点 |

[3] Note

If the actuator is paused during homing, it will start the homing sequence from the beginning upon restart.

### 3.5.14 Canceling the Operation

[1] Description

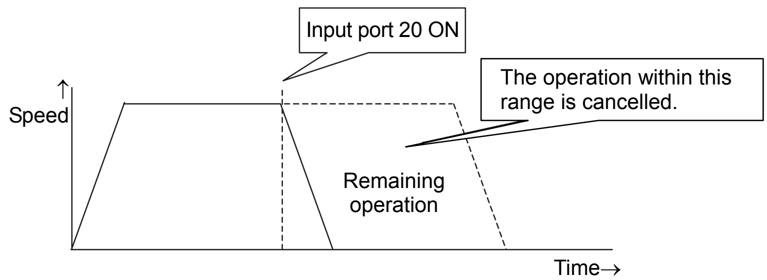
Use a declaration command CANC to decelerate the moving axis to a stop and cancel the remaining operation.

[2] Example of Use

While CANC is input, all movement commands in the same program are cancelled.

```
CANC command
   CANC     20    Cancel the movement commands if input port 20 turns ON (declaration).
        :
   MOVP      1
   MOVP      2
        :
   WTON     21
        :
```

* Declare this command in a step before the movement commands you want to cancel.
* While CANC is input, all operation commands are cancelled sequentially, while tasks other than operation commands (such as I/O processing and calculation processing) are executed sequentially.




Application

A desired input signal pattern can be selected for a CANC command using Operand 2.

0 = Contact a (Decelerates to a stop) ⇒ Same as when Operand 2 is not specified.
1 = Contact b (Decelerates to a stop)

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|
| | | | CANC | 20 | 1 | | ｷｬﾝｾﾙ入力20 B接点 |

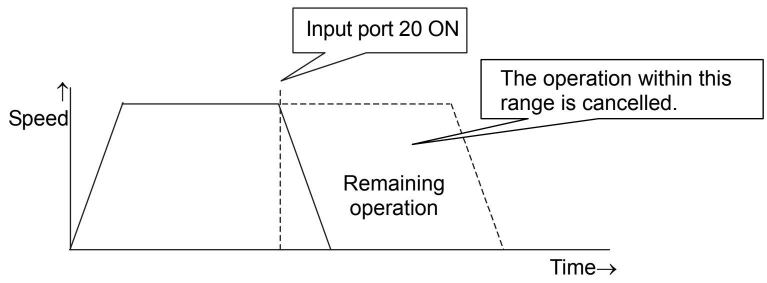
[3] Note

It is recommended that you use a WTON command to create an input waiting step, because otherwise you cannot specify which of the program steps the actuator is currently executing.

## 3.5.15 Aborting from Other Program

[1] Description
Decelerate the moving axis to a stop and cancel the remaining operation. (STOP)

[2] Example of Use
Execute a STOP command from other program to forcibly stop the operation (in the multi-tasking mode). Specify the axis you want to stop using an axis pattern.




Example 1

STOP command

| Main program | | | Stop control program | | |
|---|---|---|---|---|---|
| EXPG | n | The stop program starts. | WTON | 20 | Wait for stop input. |
| : | | | STOP | 11 | Axes 1 and 2 stop. |
| MOVL | 1 | ← | | | |
| MOVL | 2 | | | | |
| : | | | | | |

If "STOP 11" is executed while "MOVL 1" is being executed, "MOVL 1" will be cancelled and the actuator will continue its operation from "MOVL 2".

Example 2

| Main program | | | Stop control program | | |
|---|---|---|---|---|---|
| EXPG | n | The stop program starts. | WTON | 20 | Wait for stop input. |
| : | | | STOP | 10 | Axis 2 stops. |
| MOVP | 1 | ← | | | |
| MOVP | 2 | | | | |
| : | | | | | |

If "STOP 10" is executed while "MOVL 1" is being executed, only the axis 2 part of "MOVL 1" will be cancelled. Both axes 1 and 2 will operate under "MOVL 2".

[3] Note
If a STOP command is executed during a CP operation (interpolation operation) initiated by MOVL, etc., the operations of all axes will be cancelled regardless of the axis pattern specified in the STOP command.

### 3.5.16 Operation by Position Number Specification via External Signals and Output of Completion Signal to External Device

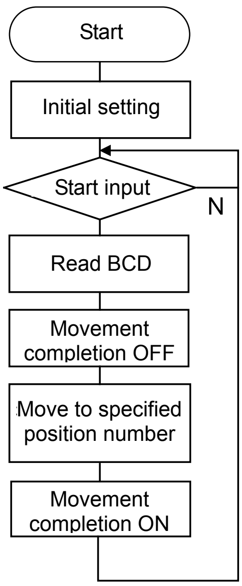
[1] Description

Load externally input BCD codes as position numbers to execute movements.

[2] Example of Use

Use an INB command to load a position number as a BCD code from an input port.
A position number can be specified using a value consisting of up to three digits.

Flowchart




Input assignment[*1]

| Port | Description |
|---|---|
| 1 | Start input |
| 15 | Position specification 1 |
| 16 | Position specification 2 |
| 17 | Position specification 4 |
| 18 | Position specification 8 |
| 19 | Position specification 10 |
| 20 | Position specification 20 |
| 21 | Position specification 40 |
| 22 | Position specification 80 |
| 23 | Position specification 100 |
| 24 | Position specification 200 |
| 25 | Position specification 400 |
| 26 | Position specification 800 |

Output

303 Movement completion

Program (Example)

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|
| | | | VEL | 100 | | | 速度設定 |
| | | | TAG | 1 | | | GOTOの飛び先 |
| | | | WTON | 1 | | | ｽﾀｰﾄ入力待ち |
| | | | INB | 15 | 3 | | ﾎﾟｼﾞｼｮﾝNo．読取り |
| | | | BTOF | 303 | | | 移動完了信号OFF |
| | | | MOVL | *99 | | | ﾎﾟｼﾞｼｮﾝNo．へ移動 |
| | | | BTON | 303 | | | 移動完了信号ON |
| | | | GOTO | 1 | | | TAG1へｼﾞｬﾝﾌﾟ |

*1 Shown above is an example of port assignment for XSEL, ASEL, PSEL and SSEL controllers.
An example for TT robots is shown below.

Input assignment

| Port | Description |
|---|---|
| 28 | Start input |
| 16 | Position specification 1 |
| 17 | Position specification 2 |
| 18 | Position specification 4 |
| 19 | Position specification 8 |
| 20 | Position specification 10 |
| 21 | Position specification 20 |
| 22 | Position specification 40 |
| 23 | Position specification 80 |
| 24 | Position specification 100 |
| 25 | Position specification 200 |
| 26 | Position specification 400 |
| 27 | Position specification 800 |

### 3.5.17 Operation by Coordinate Value Input via External Signals and Output of Completion Signal to External Device

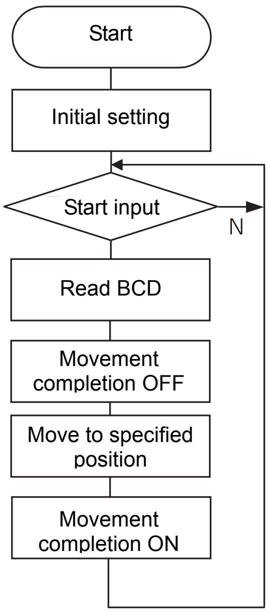
[1] Description

Receive target position data as absolute values from a host device to execute movements.

[2] Example of Use

Use an INB command to load position data as a BCD code from an input port.
Each BCD value should consist of four digits, with the last digit indicating a decimal place.
The moving axis is axis 1.
Example: If a BCD of “1234” is received, the axis will move to the position at 123.4mm.

Flowchart




Input assignment

| Port | Description |
|---|---|
| 1 | Start input |
| 15 | 0.1mm |
| 16 | 0.2mm |
| 17 | 0.4mm |
| 18 | 0.8mm |
| 19 | 1mm |
| 20 | 2mm |
| 21 | 4mm |
| 22 | 8mm |
| 23 | 10mm |
| 24 | 20mm |
| 25 | 40mm |
| 26 | 80mm |
| 27 | 100mm |
| 28 | 200mm |
| 29 | 400mm |
| 30 | 800mm |

Output

303 Movement completion

Program (Example)

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | コメント |
|---|---|---|---|---|---|---|---|
| | | | HOME | 11 | | | 原点復帰 |
| | | | VEL | 100 | | | 速度設定 |
| | | | TAG | 1 | | | GOTOの飛び先 |
| | | | WTON | 1 | | | スタート入力待ち |
| | | | INB | 15 | 4 | | 移動位置読み取り |
| | | | LET | 199 | *99 | | 小数点付けの為実数変数にコピー |
| | | | DIV | 199 | 10 | | 小数点付けの為10で割る |
| | | | PPUT | 1 | 1000 | | ポジションナンバ1000の1軸目にデータ代入 |
| | | | BTOF | 303 | | | 移動完了信号オフ |
| | | | MOVL | 1000 | | | 代入された位置に移動 |
| | | | BTON | 303 | | | 移動完了信号オン |
| | | | GOTO | 1 | | | TAG1にジャンプ |

[3] Note

With TT robots, 16 general-purpose inputs from 016 to 031 can be input. However, if coordinate values from 400mm [maximum TT stroke] to 0.0mm are inputs in 0.1mm increments and a start signal is used to start the movement, as in the example, all of the 16 general-purpose inputs are used. Take note that in this case you can no longer use general-purpose inputs for other purposes.

### 3.5.18 Output of Current Position Coordinate Value to External Device

[1] Description
Read the current actuator coordinate in real time and output the coordinate from an output port as BCD data.

[2] Example of Use
Use a PRDQ command to read the current coordinate value of axis 1.
Output the current coordinate data of axis 1 every 0.2sec as BCD output.
The output range is from 0.00 to 999.99mm.

BCD output assignment

| Output port No. | Description | Output port No. | Description |
|---|---|---|---|
| 324 | 0.01 | 337 | 20 |
| 325 | 0.02 | 338 | 40 |
| 326 | 0.04 | 339 | 80 |
| 327 | 0.08 | 340 | 100 |
| 328 | 0.1 | 341 | 200 |
| 329 | 0.2 | 342 | 400 |
| 330 | 0.4 | 343 | 800 |
| 331 | 0.8 | | |
| 332 | 1 | | |
| 333 | 2 | | |
| 334 | 4 | | |
| 335 | 8 | | |

Unit: mm

Program (Example)

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | コメント |
|---|---|---|---|---|---|---|---|
| | | | TAG | 1 | | | |
| | | | PRDQ | 1 | 101 | | 1軸現在位置を変数101へ |
| | | | MULT | 101 | 100 | | 少数点以下第三位目以下切り捨て |
| | | | LET | 99 | *101 | | 整数変数にコピー |
| | | | OUTB | 324 | 5 | | BCDで5桁分出力 |
| | | | TIMW | 0.2 | | | サンプリングタイム |
| | | | GOTO | 1 | | | |

* The current position coordinate is written to variable 101 according to the PRDQ command.
Since the value that has been read into the variable is in the xxx.xxx format, move the unused digits to below the decimal point so that the result can be output as BCD data.
In this example, the third and subsequent decimal places are not required and thus the value is multiplied by 100 to obtain the data xxxxx.x.
Next, the BCD output data is copied to dedicated variable 99.
The digit below the decimal point is rounded off at this time.
Then, the final value is output to an external device via an OUTB command.
This program is run in the multi-tasking mode as a sub-program.

[3] Note
The unit of output data may have to be changed as deemed appropriate depending on the moving range of the actuator and number of available output ports.
If coordinate values from 0mm to 400mm [maximum TT stroke] are output in 0.01mm increments, as in the example, 19 general-purpose outputs are needed. However, TT robots only have 16 general-purpose outputs of 316 to 331. Accordingly, you must take an appropriate action such as changing the unit of output data to 0.1mm.

## 3.5.19 Conditional Jump

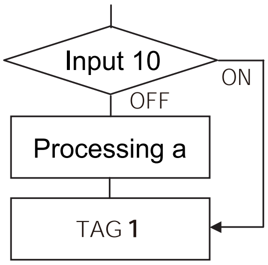
[1] Description

Select the destination to jump to via GOTO using the external input, output and/or internal flag statuses as a condition.
Process is switched over for each input.

[2] Example of Use

Example 1

If input 10 turns ON, the actuator will jump to TAG 1. If it turns OFF, the actuator will proceed to the next processing.



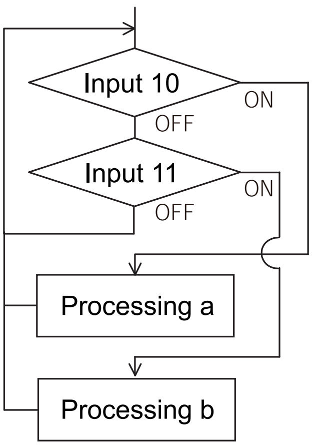

| E | N | Cnd | Cmnd | Operand 1 |
|---|---|---|---|---|
| | | 10 | GOTO | 1 |
| | | | Processing a | |
| | | | TAG | 1 |
| | | | Processing b | |

Execute GOTO 1 if input 10 turns ON.

* If input 10 turns ON, the actuator will skip processing a and perform processing b.
If input 10 turns OFF, the actuator will perform processing a, and then perform processing b.

Example 2

Wait for the input to the two ports 10 and 11, and if Input 10 becomes ON, proceed to the processing a, and proceed to the processing b if Input 11 becomes ON.




| E | N | Cnd | Cmnd | Operand 1 |
|---|---|---|---|---|
| | | | TAG | 1 |
| | | 10 | GOTO | 2 |
| | | 11 | GOTO | 3 |
| | | | GOTO | 1 |
| | | | TAG | 2 |
| | | | Processing a | |
| | | | GOTO | 1 |
| | | | TAG | 3 |
| | | | Processing b | |
| | | | GOTO | 1 |

No input.
Input 10 turns ON.
Input 11 turns ON.

If both inputs 10 and 11 turn ON, the actuator will perform processing a.

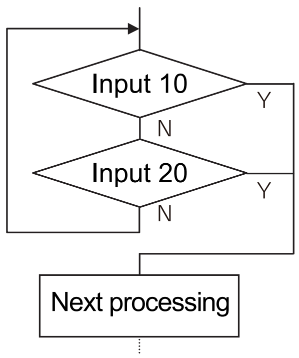
## 3.5.20 How to Pause and Then Resume Program after Output Signal Input

[1] Description

The controller waits for multiple different inputs and performs processing upon reception of any of these inputs.

[2] Example of Use

Inputs 10 and 20 are monitored, and the actuator will proceed to the next step when either input is received (OR logic).




Program a

| E | N | Cnd | Cmnd | Operand 1 |
|---|---|---|---|---|
| | | | TAG | 1 |
| | | 10 | | |
| O | | 20 | GOTO | 2 |
| | | | GOTO | 1 |
| | | | TAG | 2 |

Next processing

Program b

| E | N | Cnd | Cmnd | Operand 1 |
|---|---|---|---|---|
| | | | TAG | 1 |
| | N | 10 | | |
| A | N | 20 | GOTO | 1 |

Next processing

* Both programs a and b perform the same processing.

As shown in the sample, the controller waits for input without using a WTON command.
This method can also be used when multiple input conditions must be combined.

[3] Note

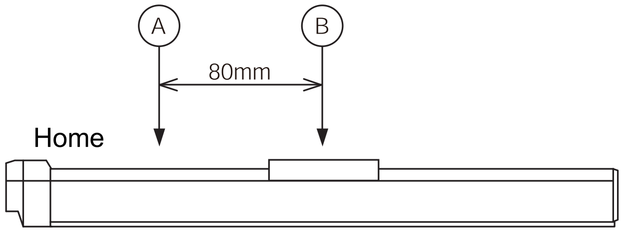
With a WTON command, the program cannot wait for multiple inputs because processing will resume upon receipt of one of the specified inputs.

## 3.5.21 How to Use Offset

[1] Description
With an OFST command, an offset can be specified for position data when you want to shift (offset) all teaching points by several millimeters because the actuator was not installed exactly in the specified position or for other reasons.

[2] Example of Use
Move the actuator from point A to point B, which is offset by 80mm from point A.

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|
| | | | VEL | 100 | | | |
| | | | MOVP | 1 | | | A点へ移動 |
| | | | OFST | 1 | 80 | | 1軸目80mmｵﾌｾｯﾄ |
| | | | MOVP | 1 | | | B点へ移動 |




[3] Note
Once an offset has been set, the offset applies to all movement commands executed thereafter. To cancel the offset, execute an offset command again by specifying 0mm. An offset does not apply to other programs (even in the multi-tasking mode). If a given offset must be applied to all programs, it must be set for all programs individually.

### 3.5.22 How to Repeat Specified Operation Multiple Times

[1] Description
Execute a specific operation n times.

[2] Example of Use
The actuator moves back and forth between P1 and P2 ten times, and then the program ends.
Use a CPEQ command to compare the number of times the movement has been actually repeated, against 10.
It is assumed that homing has been completed.

Program (Example)

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|
| | | | VEL | 100 | | | 速度設定 |
| | | | LET | 1 | 0 | | 変数ｸﾘｱ |
| | | | TAG | 1 | | | |
| | | | MOVP | 1 | | | P1へ移動 |
| | | | MOVP | 2 | | | P2へ移動 |
| | | | ADD | 1 | 1 | | 変数1に1加算 |
| | | | CPEQ | 1 | 10 | 900 | 繰返し回数確認 |
| | N | 900 | GOTO | 1 | | | 10回未満ならTAG1へ |
| | | | EXIT | | | | ﾌﾟﾛｸﾞﾗﾑ終了 |

[3] Reference
The same operation can also be performed using a DWEQ command.

## 3.5.23 Constant Feed Operation [Pitch Feed]

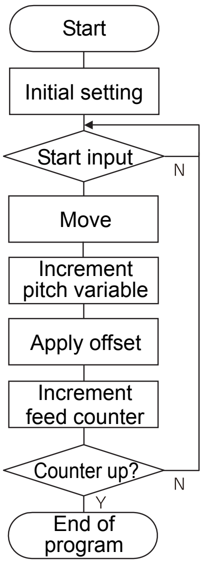
[1] Description

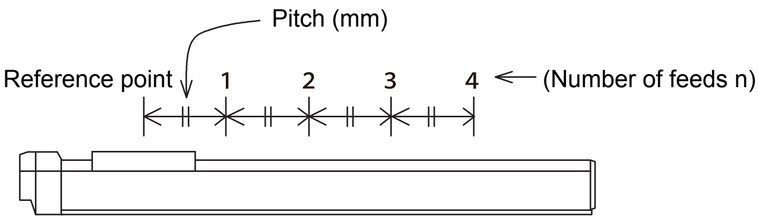
Feed the actuator by a specified pitch n times from a reference point.
The pitch and number of repetitions are specified by variables in advance.

[2] Example of Use

Use an OFST command to perform pitch feed.
The number of times the actuator has been fed is counted by a counter variable.
The X-axis is fed in the positive direction.

Flowchart




Program (Example)

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|
| | | | LET | 1 | 4 | | 送り回数(n=4)代入 |
| | | | LET | 100 | 80 | | ﾋﾟｯﾁ(80mm)代入 |
| | | | LET | 2 | 0 | | 変数ｸﾘｱ(ｶｳﾝﾀ) |
| | | | LET | 101 | 0 | | 変数ｸﾘｱ(ｵﾌｾｯﾄ値) |
| | | | HOME | 1 | | | 原点復帰 |
| | | | VEL | 100 | | | 速度設定 |
| | | | TAG | 1 | | | |
| | | | WTON | 1 | | | ｽﾀｰﾄ入力待ち |
| | | | MOVP | 1 | | | 移動 |
| | | | ADD | 101 | *100 | | ｵﾌｾｯﾄ値にﾋﾟｯﾁ加算 |
| | | | OFST | 1 | *101 | | X軸ｵﾌｾｯﾄ処理 |
| | | | ADD | 2 | 1 | | ｶｳﾝﾀ用変数に + 1 |
| | | | CPGT | 2 | *1 | 900 | 送り終了確認 |
| | N | 900 | GOTO | 1 | | | 未完了なら繰返す |
| | | | EXIT | | | | ﾌﾟﾛｸﾞﾗﾑ終了 |

[3] Note

An OFST command applies to movement commands.
Executing an OFST command alone does not move the axis.

[4] Reference

Pitch feed can also be performed with MVPI and MVLI Commands.

### 3.5.24 How to JOG via External Signal Input

[1] Description

The slider moves forward or backward while an input is ON or OFF.
Instead of an input, an output or global flag can be used as a cue.
The slider will move directly to the next step if the specified input does not satisfy the condition when the command is executed.
Regardless of the input status, the slider will stop upon reaching the soft limit, and the command in the next step will be executed.

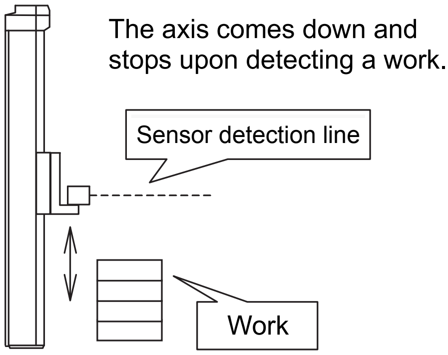
[2] Example of Use

- Explanation of commands

| | | | |
|---|---|---|---|
| JFWN | 1 | 20 | Axis 1 moves forward while input 20 is ON. |
| JFWF | 1 | 21 | Axis 1 moves forward while input 21 is OFF. |
| JBWN | 10 | 22 | Axis 2 moves backward while input 22 is ON. |
| JBWF | 10 | 23 | Axis 2 moves backward while input 23 is OFF. |

Example 1

- Stop the axis when a sensor input is received.




| | | | |
|---|---|---|---|
| : | | | |
| VEL | 50 | | Specify a low speed. |
| JFWF | 1 | 20 | Move until a sensor input (20) is received. |
| EXIT | | | The program ends. |

Example 2

- Cause the actuator to jog just like in teaching pendant operation (2 axes are operated).

<u>Program (Example)</u>

| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
|---|---|---|---|---|---|---|
| | | | TAG | 1 | | |
| | | | JFWN | 1 | 20 | |
| | | | JBWN | 1 | 21 | |
| | | | JFWN | 10 | 22 | |
| | | | JBWN | 10 | 23 | |
| | N | 24 | GOTO | 1 | | |
| | | | EXIT | | | |

[3] Note

HOLD, STOP and CANC commands remain valid while the actuators are jogging.

## 3.5.25 Switching Programs

[1] Description
Use EXPG/ABPG commands to switch programs using a program.

[2] Example of Use

Example 1

Start program 2 once the processing of program 1 is completed, and then end program 1.

| Program 1 | Program 2 |
|---|---|
| : | : |
| EXPG 2 | : |
| EXIT | |

Example 2

Start a program via an external signal, and then end the other program.

| Program 1 | Program 2 |
|---|---|
| ABPG 2 | ABPG 1 |
| : | : |

If program 2 is started while program 1 is running, program 1 will be aborted.
If program 1 is started while program 2 is running, program 2 will be aborted.

Application

If a program number is specified in operand 2, the programs from the one corresponding to the program number in operand 1 to the other corresponding to the program number in operand 2 can be started (EXPG) or ended (ABPG) simultaneously.

[3] Note

- Up to 16 programs (maximum of 8 programs in the case of ASEL/PSEL/SSEL controllers) can be run at the same time. To use other programs when the controller is already running 16 programs, switch programs by closing a program or programs that are not required.
- If an ABPG command was executed to end a program while the program was executing a movement command, the actuator immediately decelerates to a stop.

## 3.5.26 Aborting a Program

[1] Description

Abort a program currently running.
Execute an ABPG command (command to abort other program) from other program in the multi-tasking mode.

[2] Example of Use

Main program (Prg. 1)

```
EXPG    n    The abort control program starts.
WTON   10
MOVP    1
BTON  303
       :
       :
```

Abort control program (Prg. n)

```
WTON   20    Wait for an abort input.
ABPG    1    Prg. 1 is aborted.
EXIT         The program ends.
```

[3] Note

If the running program was executing any movement command, the applicable axis immediately decelerates to a stop and then the program ends.

### 3.5.27 Way to Prevent Duplicated Startup by Program

[1] Description

How to prevent other programs from starting redundantly using virtual I/O port N710☐ (Program No.☐ running) is explained. If a given program is not running as determined by the checking of corresponding virtual I/O port N710☐ (Program No.☐ running), that program is started.

[2] Example of Use

```
   PRG1 (for task management)
   TAG1
N7102   EXPG      2   PRG2 stopped
                            PRG2 running
N7103   EXPG      3   PRG3 stopped
                            PRG3 running
N7104   EXPG      4   PRG4 stopped
                            PRG4 running
        TIMW   0.02   Task open
        GOTO      1
```

```
PRG2 (auto)
ABPG   3   4
   .
   .
```

```
PRG3 (step)
ABPG   2
ABPG   4
   .
   .
```

```
PRG4 (JOG)
ABPG   2   3
   .
   .
```

## 3.5.28 How to Cause Rotational Axis [Multi-rotation Specification] to Rotate Multiple Times

Regarding the axis operation types and rotation axis modes
(1) Axis-specific parameter No. 1, “Axis operation type”

| No. | Parameter name | Default value | Input range | Unit |
|---|---|---|---|---|
| 1 | Axis operation type | Varies depending on the actuator. | 0 to 1 | None |

● Explanation
This parameter defines the type of the actuator used. (Set this to 1.)
● Setting values
0: Linear movement axis   Actuator other than rotational axis
1: Rotational movement axis   Rotational axis (RS-30/60, RCS2-RT*/RTC*)

(2) Axis-specific parameter No. 66, “Rotational axis mode selection”

| No. | Parameter name | Default value | Input range | Unit |
|---|---|---|---|---|
| 66 | Rotational axis mode selection | 0 | 0 to 5 | None |

● Explanation
This parameter selects a desired rotational axis mode. (Set this to 1.)
Related parameter: Axis-specific No. 7, “Soft limit+”
● Setting values
0: Normal
1: Index mode
* When the index mode is selected, the soft limit is fixed to 359.999mm internally. Short-cut control is enabled while the index mode is selected.
2 to 5: Reserved by the system

⚠Caution: Absolute-specification actuators do not support the following settings:
- If this parameter is set to 0 (Linear movement axis), the infinite stroke mode cannot be set with parameter No. 68.
- If this parameter is set to 1 (Rotational movement axis), short-cut control cannot be selected in parameter No. 67.

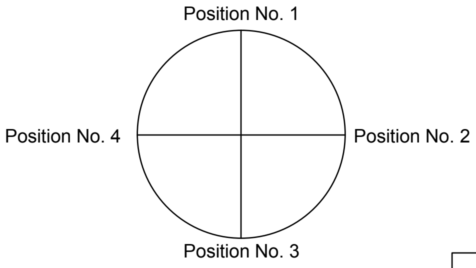
(3) Axis-specific parameter No. 67, “Short-cut control selection for rotational movement axis”

| No. | Parameter name | Default value | Input range | Unit |
|---|---|---|---|---|
| 67 | Short-cut control selection for rotational movement axis | 0 | 0 to 1 | None |

- Set this parameter to 1 when the rotation of the rotary axis is required to be in one way.
  Multi-rotation operation can be performed by setting this parameter to 1 (Short-cut control selected) and repeating a movement command in the same rotating direction.
  What is short-cut control?
  A type of operation in which the actuator moves to the closest point in the next move.
  It may shortcut if a far point is indicated. Pay attention to the indication value if the multi-rotation operation is preferred.

- Setting values
  0: Not selected
  1: Selected

Example: Move the actuator through position No. 2 → 3 → 4 → 1 successively based on position No. 1 being the reference point.




Program

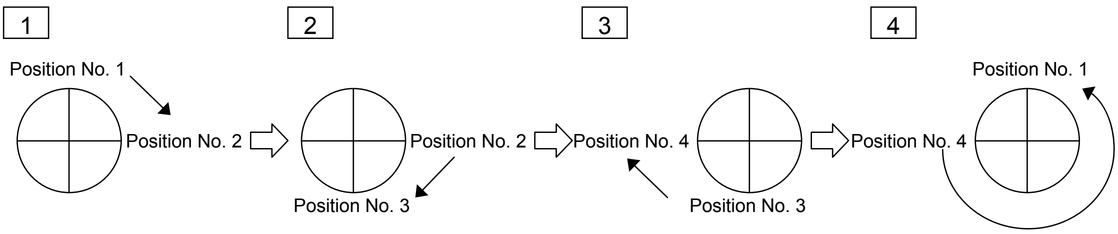
| Step No. | Command | Operation |
|---|---|---|
| 1 | MOVP | 2 |
| 2 | MOVP | 3 |
| 3 | MOVP | 4 |
| 4 | MOVP | 1 |

Position

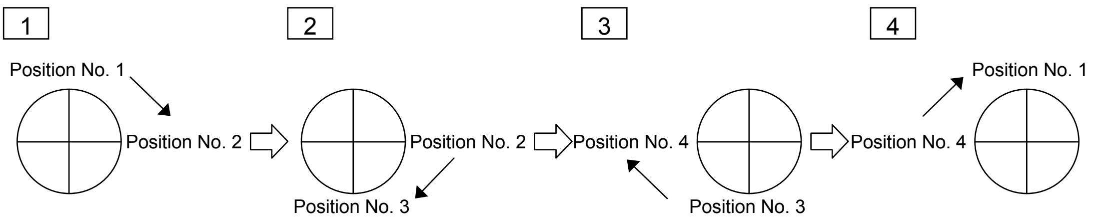
| Position No. | Position data |
|---|---|
| 1 | 0 |
| 2 | 90 |
| 3 | 100 |
| 4 | 270 |

1° of position data corresponds to 1mm.

Short-cut control not selected




Short-cut control selected




* By selecting short-cut control, you can cause the actuator to rotate in a specific direction.

## 3.6 For Advanced Operations (program edit)

### 3.6.1 Handling of Axis Numbers and Patterns

You can specify each axis using an axis number or multiple axes using an axis pattern.

[1] Axis number and indication of axis
Axes are indicated as follows so that multiple axes can be expressed.
To specify only one of multiple axes, specify it by the applicable axis number.

Single axis/rectangular axis

| Axis | Axis number |
|---|---|
| Axis 1 | 1 |
| Axis 2 | 2 |
| Axis 3 | 3 |
| Axis 4 | 4 |
| Axis 5 | 5 |
| Axis 6 | 6 |
| Axis 7 | 7 |
| Axis 8 | 8 |

SSEL, ASEL and PSEL controllers support only up to two axes. TT robots support only up to three axes.

In addition to following the above rule, you can also express axis numbers using symbols.

SCARA robot

| Axis | Axis number |
|---|---|
| X-axis | 1 |
| Y-axis | 2 |
| Z-axis | 3 |
| R-axis | 4 |

(Note) The movements of arms 1 and 2 of a SCARA robot are interlocked. It is not that arm 1 always represents the X-axis and arm 2, Y-axis.
Consider that the X-axis (axis No. 1) moves in the direction of X coordinates, while the Y-axis (axis No. 2) moves in the direction of Y coordinates.
Note that only when an AXST command is issued, the X-axis represents the arm 1 axis, while the Y-axis represents the arm 2 axis.

In addition to following the above rule, you can also express axis numbers using symbols.

[2] Axis pattern

[Single axis/rectangular axis]
Use “1” or “0” to indicate which axis(es) you want to use.

(Higher) (Lower)

| Axis | Axis 8 | Axis 7 | Axis 6 | Axis 5 | Axis 4 | Axis 3 | Axis 2 | Axis 1 |
|---|---|---|---|---|---|---|---|---|
| Use | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Do not use | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

[Example] Use axes 1 and 2.

Axis 2
↓
You should state 000011… (The 0’s at the beginning are not necessary. Take them away and use 11.)
↑
Axis 1

[Example] Use axes 1 and 4.

Axis 4
↓
You should state 1001… (In this case, the 0’s are needed to indicate the position of axis 4.)
↑
Axis 1

Specifying an axis pattern indirectly using a variable
Consider an axis pattern as a binary expression and assign a decimal equivalent of it to a variable.

[Example] Axis pattern where home return is performed only for axis 3

```
HOME    100
```

This pattern is specified indirectly as follows:
100 (binary) → 4 (decimal)
Accordingly:

```
LET     6     4
HOME    *6
```

If multiple axes must be specified at the same time, use an axis pattern.

- Commands where an axis pattern is used to specify axes
  OFST, GRP, SVON, SVOF, HOME, JFWN, JFWF, JBWN, JBWF, STOP, PTST, PRED CHVL, PBND, WZNA, WZNO, WZFA, WZFO, PAXS, NBND, PTRQ, MOVD, MVDI, NTCH, RAXS, XAXS, ECMD(250)

[SCARA robot]
Use “1” or “0” to indicate which axis(es) you want to use.

| | (Higher) | | | (Lower) |
|---|---|---|---|---|
| Axis | R-axis | Z-axis | Y-axis | X-axis |
| Use | 1 | 1 | 1 | 1 |
| Do not use | 0 | 0 | 0 | 0 |

(Note) The movements of arms 1 and 2 of a SCARA robot are interlocked. It is not that arm 1 always represents the X-axis and arm 2, Y-axis.
Consider that the X-axis (axis No. 1) moves in the direction of X coordinates, while the Y-axis (axis No. 2) moves in the direction of Y coordinates.

[Example] Use the X-axis and Y-axis.

Y-axis
↓
You should state 0011… (The 0’s at the beginning are not necessary. Take them away and use 11.)
↑
Axis 1

[Example] Use the X-axis and R-axis.

R-axis
↓
You should state 1001… (In this case, the 0’s are needed to indicate the position of axis R.)
↑
X-axis

Specifying an axis pattern indirectly using a variable
Consider an axis pattern as a binary expression and assign a decimal equivalent of it to a variable.

If multiple axes must be specified at the same time, use an axis pattern.

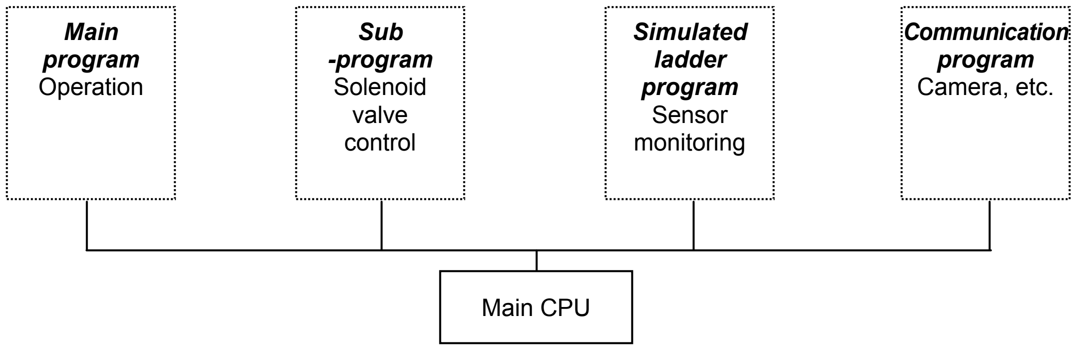
- Commands where an axis pattern is used to specify axes
  OFST, GRP, PTST, PRED, PBND
  (Note) In the case of SVON, SVOF and STOP, all axes are specified regardless of the axis pattern.

## 3.6.2 Setting of Multi-tasking and Task Level

[1] Multi-tasking

Controllers come standard with the multi-tasking function that allows multiple programs to be run simultaneously, such as moving actuators in one program and turning ON/OFF solenoid valves in another programs.
"Multi-tasking" literally means performing multiple tasks. The main CPU processes each program step by step. If multiple programs are run that contain actuator commands, timer commands, input waiting commands, etc., however, the main CPU uses an idle time while waiting for completion of each commanded task to process different programs.
(If a given program has no idle time, the system forcibly switches to the next program after 1msec based on "time slicing" action.)
XSEL controllers adopt high-speed CPUs, so multi-tasking is also performed at high speed.
Note that this function also supports simulated ladder circuits, which means that as long as your equipment is small enough you can build it as a sequencer.

[Example of multi-tasking (running multiple programs)]

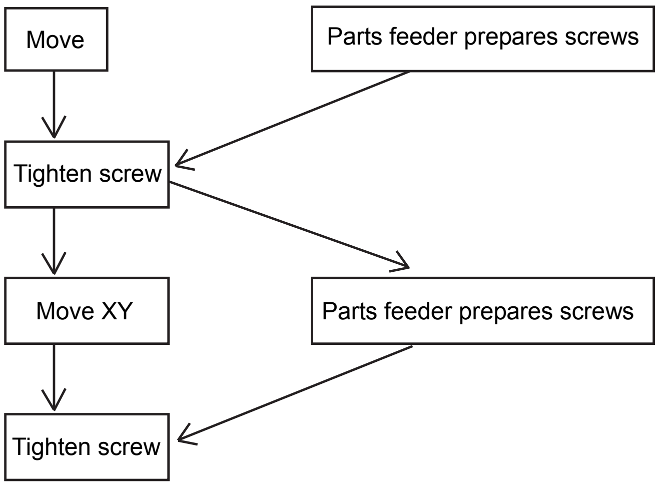



[2] Task level
If you want to perform a given task (program) preferentially over other tasks (programs), you can do so with a CHPR command by setting the parameter to "1: HIGH". If the parameter is set to "0: NORMAL", no priority is set.
You can also set task levels for simulated ladder programs. [Refer to Section 3.6.3]

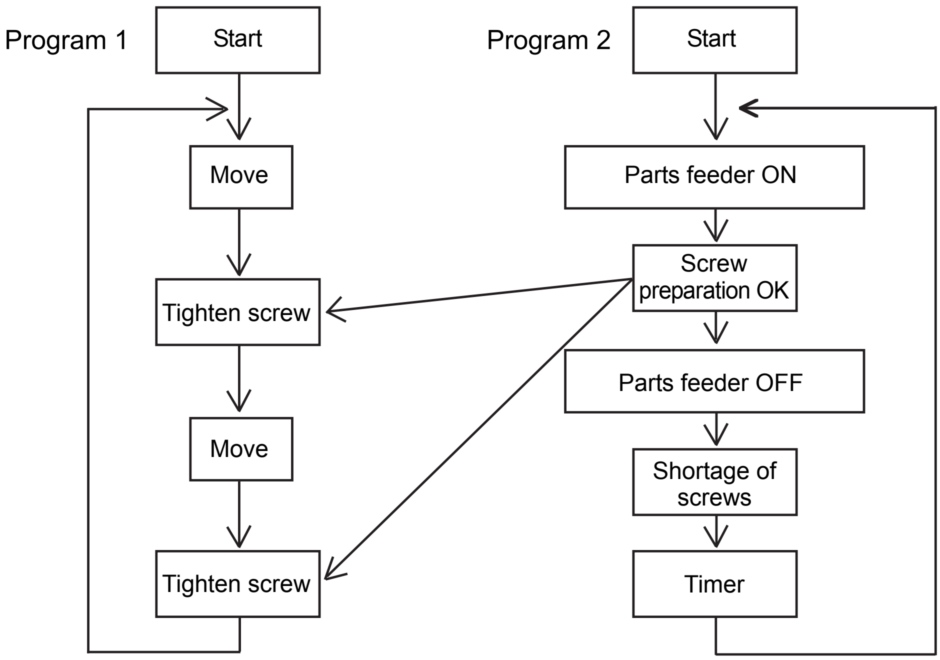
[3] Multi-Tasking

Take a screw-tightening robot, for example.
In general, a screw-tightening robot consists of axis 1 and axis 2 actuators and a screw-tightening machine (up/down air cylinder, etc.).

Operation Flow




Although the flow chart is simple, the movement of axis 1 and axis 2 actuators and the operation of the parts feeder must take place simultaneously. This requires "multi-tasking" operation.



[4] Program Switching

Various methods are available to switch between programs, depending on the purpose of programs. The representative methods are explained below.

```
                     ┌─ External start
Program switching ───┤
                     └─ Program ───┬─ Single-tasking -----------EXIT command
                                   └─ Multi-tasking -------------EXPG command
```

There are mainly two ways. One is to conduct with external startup and the other with application program.

(1) External start method ······· Refer to the Instruction Manual for each controller.

(2) Program method

○ Single-tasking

By executing EXIT Command (program finish) at the end of each program, finish the program and put back to the condition when the power is turned off. The home position, however, is remained, thus the next program can be executed with external start input by specifying another program number.

○ Multi-tasking

By creating a program for control and executing EXPG Command (startup of another program) in the program, multiple programs run in parallel one after another.

### 3.6.3 Pseudo-Ladder Task

A pseudo-ladder task function can be used depending on the command and extension condition.
The input format is shown below. Note that this function must be used by expert engineers with a full knowledge of PLC software design.

[1] Basic Frame

| Extension condition | N | Input condition | Command | Operand 1 | Operand 2 | Output | |
|---|---|---|---|---|---|---|---|
| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | |
| LD | | 7001 | CHPR | 1 | | | |
| | | | TPCD | 1 | | | |
| | | | TAG | 1 | | | |
| | | | | | | | Ladder statement field |
| LD | | 7001 | TSLP | 1 to 100 | | | |
| | | | | | | | Ladder statement field |
| | | | | | | | |
| LD | | 7001 | TSLP | 1 to 100 | | | |
| LD | | 7001 | GOTO | 1 | | | |
| LD | | 7001 | EXIT | | | | |

*      * Virtual input 7001: "Normally ON" contact

[2] Ladder Statement Field

1) Extension conditions

LD ·······LOAD
A ·······AND
O ·······OR
AB ·······AND BLOCK
OB ·······OR BLOCK

All of the above extension conditions can be used in non-ladder tasks.

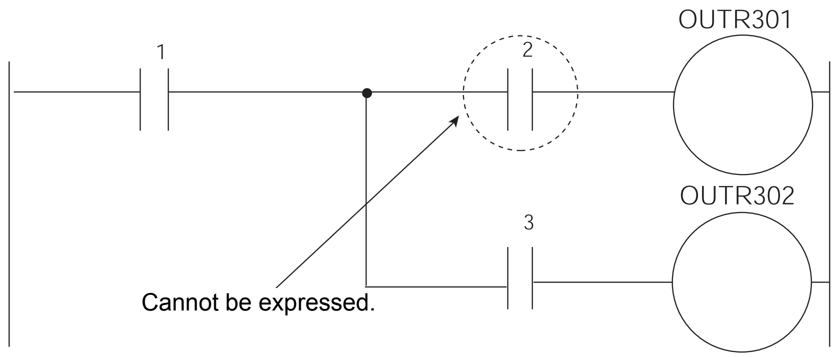
2) Ladder commands

OUTR ·········· Ladder output relay (Operand 1 = Output, flag number)
TIMR ·········· Ladder timer relay
(Operand 1 = Local flag number, Operand 2 = Timer setting (sec))

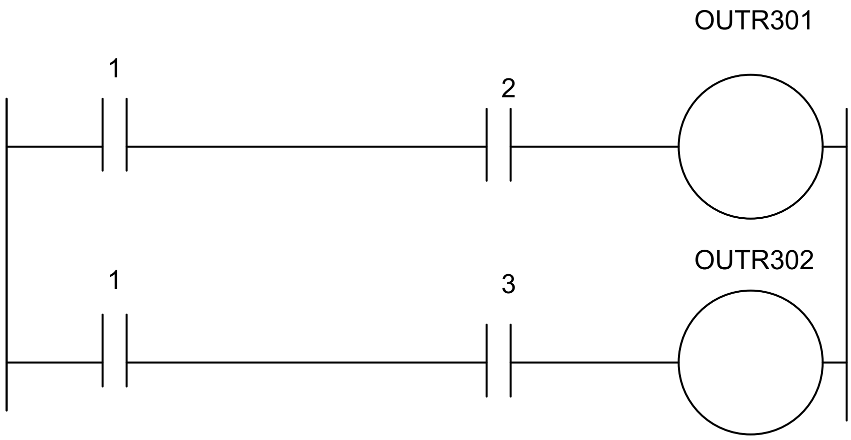
[3] Points to Note

- This system only processes software ladders using an interpreter. Therefore, the processing time is much longer than that of a dedicated commercial sequencer.
  (This system is not suitable for large-scale ladder processing.)
- If an extension condition is not specified for steps in which an input condition is specified, the steps will be treated as LD (LOAD).
- Always specify a “normally ON” contact for those steps that must be processed without fail, such as CHPR, TSLP and GOTO. (LD 7001)
  Virtual input 7001: “Normally ON” contact

- Ladder processing is based on software ladders using an interpreter, you cannot branch an output “1” to produce an input “2” or “3” as shown in the input circuit below.

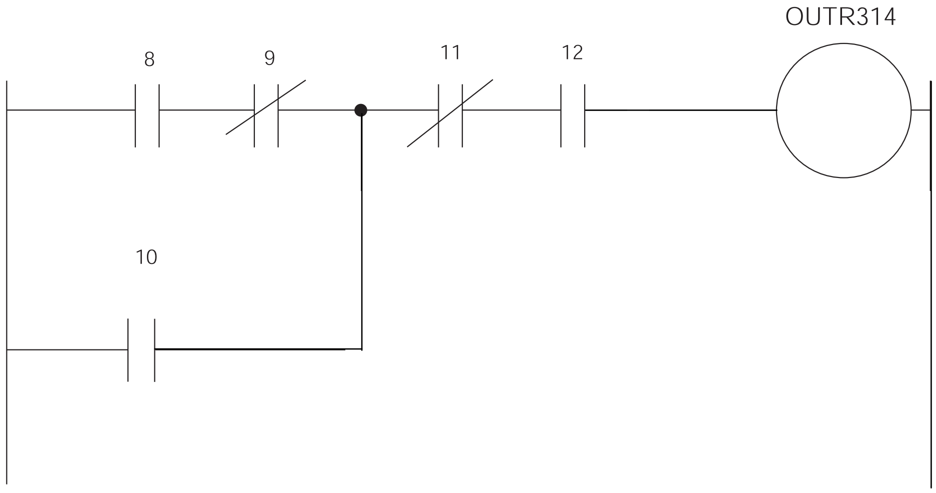



To perform this operation, you can write a ladder as follows, for example.
However, this is conditional upon the output “1” not changing during the output processing at OUTR301 in line 1. Make sure the output “1” does not change due to other programs.



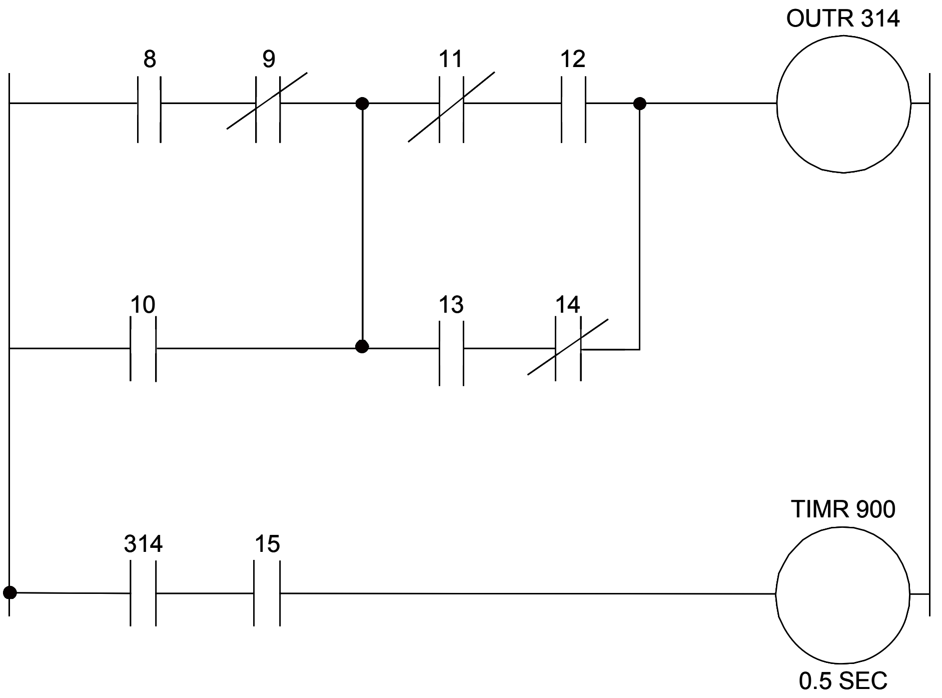

| Extension condition | N | Input condition | Command | Operand 1 | Operand 2 | Operand 3 |
|---|---|---|---|---|---|---|
| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Operand 3 |
| | | | | | | |
| LD | | 1 | | | | |
| A | | 2 | OUTR | 301 | | |
| LD | | 1 | | | | |
| A | | 3 | OUTR | 302 | | |
| | | | | | | |

[4] Program Example




| Extension condition | N | Input condition | Command | Operand 1 | Operand 2 | Output |
|---|---|---|---|---|---|---|
| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| LD | | 7001 | CHPR | 1 | | |
| | | | TPCD | 1 | | |
| | | | TAG | 1 | | |
| | | | | | | |
| LD | | 8 | | | | |
| A | N | 9 | | | | |
| O | | 10 | | | | |
| LD | N | 11 | | | | |
| A | | 12 | | | | |
| AB | | | OUTR | 314 | | |
| | | | | | | |
| LD | | 7001 | TSLP | 3 | | |
| LD | | 7001 | GOTO | 1 | | |
| LD | | 7001 | EXIT | | | |

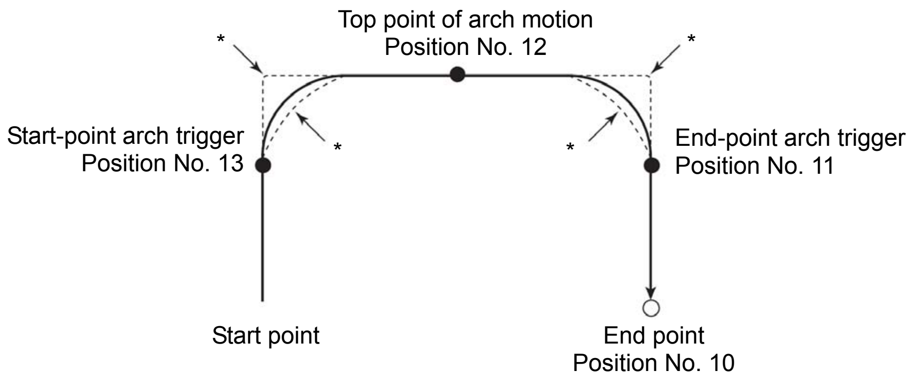
An example where 13, 14, 15 and timer TIMER900 are added further is given below.




| Extension condition | N | Input condition | Command | Operand 1 | Operand 2 | Output |
|---|---|---|---|---|---|---|
| E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| LD | | 7001 | CHPR | 1 | | |
| | | | TPCD | 1 | | |
| | | | TAG | 1 | | |
| | | | | | | |
| LD | | 8 | | | | |
| A | N | 9 | | | | |
| O | | 10 | | | | |
| LD | N | 11 | | | | |
| A | | 12 | | | | |
| LD | | 13 | | | | |
| A | N | 14 | | | | |
| OB | | | | | | |
| AB | | | OUTR | 314 | | |
| LD | | 314 | | | | |
| A | | 15 | OUTR | 900 | 0.5 | |
| LD | | 7001 | TSLP | 3 | | |
| LD | | 7001 | GOTO | 1 | | |
| LD | | 7001 | EXIT | | | |

### 3.6.4 How to Use Arch Motion

Move from the current position to end point via arch motion.




Example of program

```
ACHZ   3                                                          -------- Declare arch motion Z-axis
ATRG  13 (start-point arch trigger position)  11 (end-point arch trigger position)  -------- Set arch trigger

ARCH  10 (position of end point)              12 (position of top point of arch motion) ------  Arch motion
```

- Use an ACHZ command to specify the arch motion Z-axis. In the case of a SCARA robot, you only need to specify the Z-axis (axis 3) to perform arch motion. (ACHZ 3)
- Use an ATRG command to specify the arch motion trigger.
  After rising up to the start arch trigger from the current position by ARCH Command, a movement in the direction other than Z-axis starts with the arch motion.The actuator passes the top point being the Z point specified in operand 2 and completes the movement in a direction other than that of arch motion Z-axis, after which it passes near the end-point arch trigger and reaches the positions of the specified point.

Note

When operation is resumed after a pause, the transition from rising movement to horizontal movement and transition from horizontal movement to rising movement may follow the paths indicated by * (dotted lines) in the figure. Exercise caution to prevent contact.

- The arch motion Z-axis coordinate at the end point corresponds to the sum of the arch-motion Z-axis component of position data specified in operand 1, if any, and the arch-motion Z-axis offset. If the position data has no arch-motion Z component, the arch motion Z-axis coordinate corresponds to the sum of the arch motion Z-axis coordinate at the start point and the arch motion Z-axis offset. (Normally an offset is added to all positions such as the arch trigger and Z point.)
- If the start-point arch trigger is set below the start point or end-point arch trigger is set below the end point, an error occurs. (Note: The upward and downward directions have nothing to do with + and - of coordinates.)
- The rising direction of the arch motion Z-axis is the direction of moving from the end point to Z point (while the downward direction is the opposite of that direction), and has nothing to do with the magnitude correlation of coordinate values. Accordingly, be sure to check the actual operating direction when using this command.
- As for the data of end-position arch trigger, also start/end the operation at a point above the applicable arch trigger for any effective axis data other than data of the arch motion Z-axis, if available.
- If a composite arch trigger is set and any effective axis data is available other than data of the effective axis at the end point or arch motion Z-axis, the applicable axis also operates. In this case, also start/end the operation at a point above the applicable arch trigger.

### 3.6.5 How to Use Palletizing Function

The SEL language provides palletizing commands that support palletizing operation. These commands allow simple specification of various palletizing settings and enable arch motion ideal for palletizing. You can also call a subroutine at the palletizing destination to perform palletizing operation.

[1] How to Use

Use palletizing commands in the following steps:

(1) Palletizing setting
Set palletizing positions, arch motion, etc., using palletizing setting commands.

(2) Palletizing calculation
Specify palletizing positions using palletizing calculation commands.

(3) Palletizing movement
Execute motion using palletizing movement commands.

[2] Palletizing Setting

Use the palletizing setting commands to set items necessary for palletizing operation. The setting items include the following:

[Palletizing number setting ................ Command: BGPA]

At the beginning of a palletizing setting, determine a palletizing number using a BGPA command to declare the start of palletizing setting.
At the end, declare the end of palletizing setting using an EDPA command.

| | | |
|---|---|---|
| BGPA | 1 | Declare the start of setting for palletizing No. 1. |
| | | Set palletizing in these steps. |
| EDPA | | Declare the end of palletizing setting at the end. |

A maximum of 10 sets (palletizing No. 1 to 10) of palletizing setting can be specified for each program.

[Palletizing pattern ................ Command: PAPN]

Select a pattern indicating the palletizing order.
The two patterns illustrated below are available.
The encircled numbers indicate the order of palletizing and are called “palletizing position numbers”.

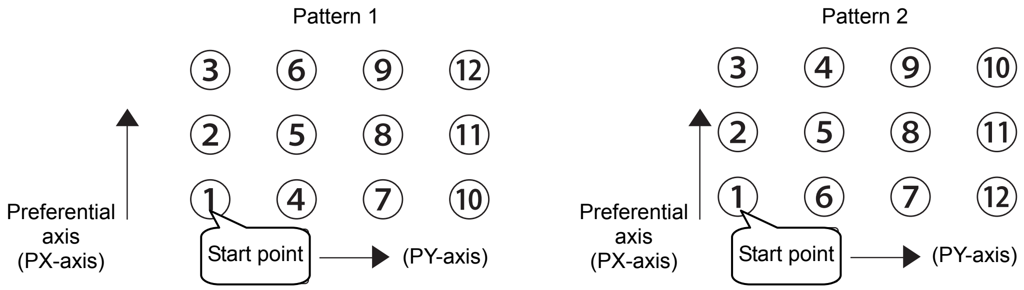



Fig. 1

| | | | |
|---|---|---|---|
| PAPN | 2 | | When pattern 2 is selected<br>(Setting is not necessary if pattern 1 is selected.) |

The row from 1 to 3 to be placed first is called the “preferential axis (PX-axis)”, while the other direction comprising the palletizing plane is called the “PY-axis”.

[Palletizing counts ................ Command: PAPI]

Set the palletizing counts.

| | | | |
|---|---|---|---|
| PAPI | 3 | 4 | Count for preferential axis (PX-axis): 3, Count for PY-axis: 4 |

[Palletizing position setting]

Palletizing position setting is performed mainly by method A or B, as explained below. Set the palletizing positions for each palletizing setting based on method A or B.

| | Setting method | Commands |
|---|---|---|
| A | [3-point or 4-points teaching method]<br>Set three position-data points or four position-data points specifying the palletizing positions. | PAPS |
| B | [Method to set palletizing positions in parallel with the actuators (in parallel with an axis on the work coordinate system in the case of a SCARA robot)]<br>Set from the palletizing axes, palletizing reference point and palletizing pitches. | PASE, PAST<br>PAPT |

A. 3-point teaching method

To set the palletizing positions by 3-point teaching, store desired positions in position data fields as three continuous position data and then specify the first position number using a PAPS command.
This method allows you to set the PX-axis and PY-axis as three-dimensional axes not parallel with the actuators and not crossing with each other.
In the example shown below, position data [1], [3] and [10] are stored in three continuous position data fields.

When three points are taught from position No. 11

Position No. 11 [1] : Start point (First palletizing position)
Position No. 12 [3] : Palletizing position corresponding to the end point in the PX-axis direction
Position No. 13 [10]: Palletizing position corresponding to the end point in the PY-axis direction
(Position No. 14 [12]: End point (For 4-point teaching))
The encircled numbers indicate palletizing position numbers (palletizing order).

Use a PAPS command to specify the position number corresponding to the start point.

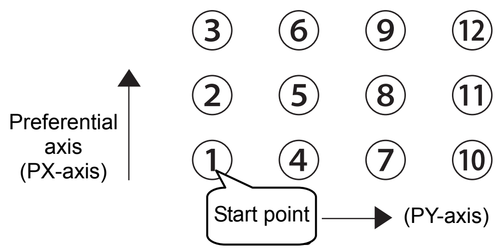



Fig. 1

PAPS 11

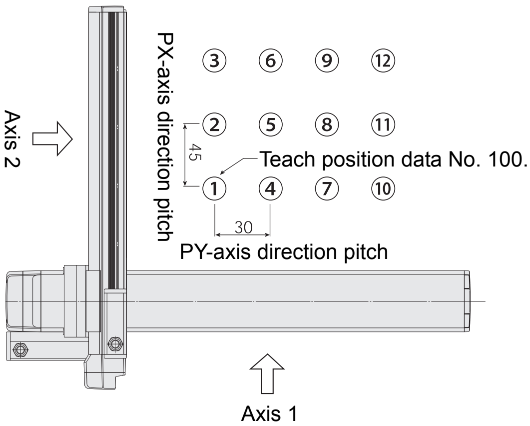
The pitches are calculated automatically from the count set for each axis.
In 3-point teaching, you can specify position data for two axes or three axes. If position data is specified for three axes, the palletizing plane becomes a three-dimensional plane.

Leave the R-axis data box blank for the position data indicated with PAPS Command for SCARA Robot. (Or make it invalid with GRP Command)
Set the R-axis coordinates for palletizing position with PEXT Command.

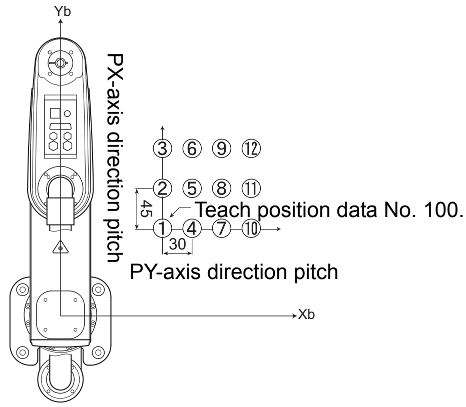
B. Method to set palletizing positions in parallel with the actuators

| | |
|---|---|
| Palletizing reference point: | Store the position data of the start point (palletizing position No. 1) in a position data field and specify the applicable position number using a PAST command, as shown below. For SCARA Robot, set the R-axis coordinates for palletizing position with PEXT Command. |
| Palletizing pitches: | Use a PAPT command to specify the pitches in the PX-axis and PY-axis directions. |
| Palletizing axes: | Use a PASE command to specify the two axes, one representing the PX-axis direction and the other representing the PY-axis direction, to be used in palletizing. |

(Axis number of an actuator in perpendicular to the axis in priority number in an actuator (work coordinate system axis when SCARA Robot) in parallel to the priority axis (PX Axis))




Rectangular Axes




SCARA Robot

| | | | |
|---|---|---|---|
| PAST | 100 | | Teach position data No. 100 as the start point. |
| PAPT | 45 | 30 | The PX-axis direction pitch is 45mm and the PY-axis direction pitch is 30mm. |
| PASE | 2 | 1 | Set axis 2 as the priority axis (PX-axis) and axis 1 as the rectangular axis and the priority axis. |

(Note) When the above palletizing axes, palletizing pitches and palletizing reference point are used, the PX-axis and PY-axis must be parallel with the actuators and crossing with each other. In the example in the figure for SCARA Robot, shows the case of the work coordinate system No. 0 (Base Coordinate System).

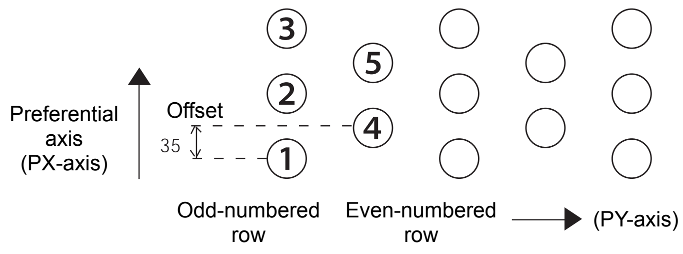
Select either method A or B for each palletizing setting.

[Zigzag setting ................ Command: PSLI]

Use a PSLI command to set a zigzag layout as shown below.

Zigzag offset: Offset amount in the preferential-axis direction, which will be applied when even-numbered rows are placed.
"Even-numbered rows" refer to the rows occurring at the even numbers based on the row placed first representing the first row.

Zigzag count: Number in the even-numbered rows. Two in the diagram below.




PSLI 35 2

[Arch motion setting]

(a) Arch motion Z-axis No. ......................... Applicable command: ACHZ

- In the case of a SCARA robot, you only need to specify the Z-axis (axis 3) to perform arch motion.

  ACHZ 3

(b) Arch motion Z-axis offset ...................... Applicable command: OFAZ

(c) Composite arch motion ......................... Applicable command: AEXT

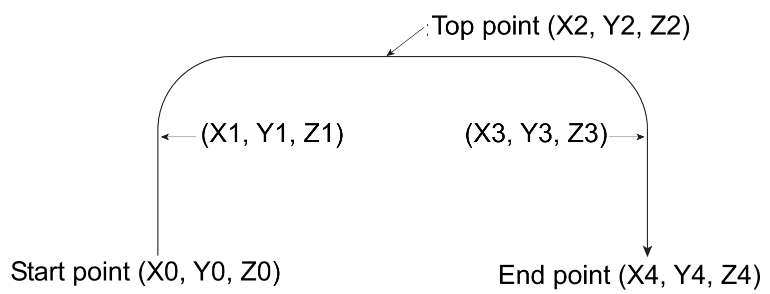
Composite arch motion data refers to position data used when you want to cause any axis other than the effective axis at the end point or arch motion Z-axis to perform an additional operation (such as when setting a rotational angle).
Note, however, that any composite axis operation starts and ends at a position above the applicable arch trigger.
Set this composite arch motion setting command by specifying a position number under which composite arch motion data is set.

(d) Arch trigger .......................................... Applicable command: ATRG

The following arch trigger settings are available for arch motion.
Set this arch trigger setting command by specifying a position number under which arch trigger coordinate data is stored.

(d-1) Start-point arch trigger
Specify the position to be reached after the arch motion is started from the start point and the actuator moves in the arch motion Z-axis coordinate direction, after which the actuator will start moving in the direction of other axis. Start-point arch trigger = Z1

(d-2) End-point arch trigger
Specify the position to be reached in the arch-motion Z-axis coordinate direction during the arch motion down movement, after which the actuator will end moving in the direction of other axis. End-point arch trigger = Z3



[Palletizing arch motion setting]

(a) Axis number corresponding to palletizing Z direction...... Applicable command: PCHZ

(b) Palletizing Z-axis offset .................................................. Applicable command: OFPZ

(c) Composite palletizing ...................................................... Applicable command: PEXT

Composite palletizing data refers to position data used when you want to cause any axis other than the PX- or PY- (or PZ-) axis to perform an additional operation in a palletizing movement command (such as when setting a rotational angle).

Note, however, that any composite axis operation starts and ends at a position above the applicable palletizing arch trigger.

Set this composite palletizing setting command by specifying a position number under which composite palletizing data is stored.

(d) Palletizing arch trigger .................................................... Applicable command: PTRG

If the end point is the palletizing point, you need palletizing arch triggers just like arch triggers.

Set this palletizing arch trigger setting command by setting a position number under which palletizing arch trigger coordinate data is stored.

(d-1) Palletizing start-point arch trigger

(d-2) Palletizing end-point arch trigger

[3] Palletizing Calculation

The items that can be operated or obtained using palletizing calculation commands are shown below:

[Palletizing position number Commands ...... PSET, PINC, PDEC, PTNG]

Number showing the ordinal number of a palletizing point. (In Fig. 1 for [2] given in the explanation of palletizing pattern, the encircled numbers are palletizing position numbers.)

| |
|---|
| Always set this command before executing a palletizing movement command (ARCH excepted) --- PSET |

For example, executing a palletizing movement command by setting 1 as the palletizing position number will move the axes to the start point. Executing a palletizing movement command by setting 2 as the palletizing position number will move the axes to the point immediately next to the start point in the PX-axis direction.

[Palletizing angle Command ........PARG]

It is an angle (θ in the figure below) of the palletizing preferential axis (PX-axis) against the work coordinate system axis.

θ indicates an angle calculated by ignoring the coordinate in the palletizing Z-axis direction.

In the figure below, θ will become a positive value if axis 1 is used as the reference for angle calculation.

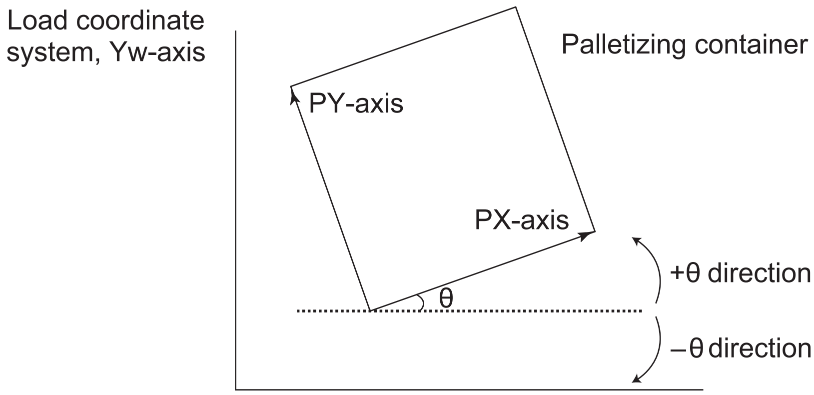



Fig. 4

If the composite axis is a rotating axis, you can obtain the palletizing angle and add it as an offset to the operation of the composite axis in order to correct the composite axis against any position error of the palletizing container.

With XSEL commands, executing a “get palletizing angle” command following a palletizing setting via 3-point teaching will automatically obtain the palletizing angle.

If 3-point teaching is set three-dimensionally, you must specify the palletizing Z-axis.

[Palletizing calculation data Command .....PAPG]

When a palletizing position number is set, this data refers to the position coordinate data of the palletizing point corresponding to that palletizing position number.

Note, however, that this position coordinate data does not reflect any normal offset or palletizing Z-axis offset.

[4] Palletizing Movement

Palletizing movement commands include commands used to move the actuator to palletizing points and other that uses position data to specify the end point.

[Movement commands to palletizing point ....... PMVP, PMVL, PACH]

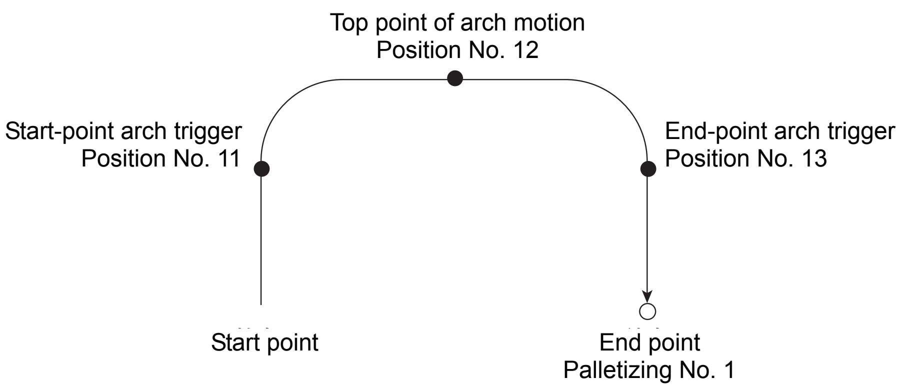
Calculate the position coordinate of a two-dimensionally or three-dimensionally positioned palletizing point and use this coordinate as the end point to move the actuator. (The actuator moves to the palletizing point corresponding to the palletizing position number specified in the command when executed)

You need two actuator axes to constitute a two-dimensional plane. If you need a vertical axis (PZ-axis), you must specify one more axis.

PMVP: Move from the current position to a palletizing point via PTP.

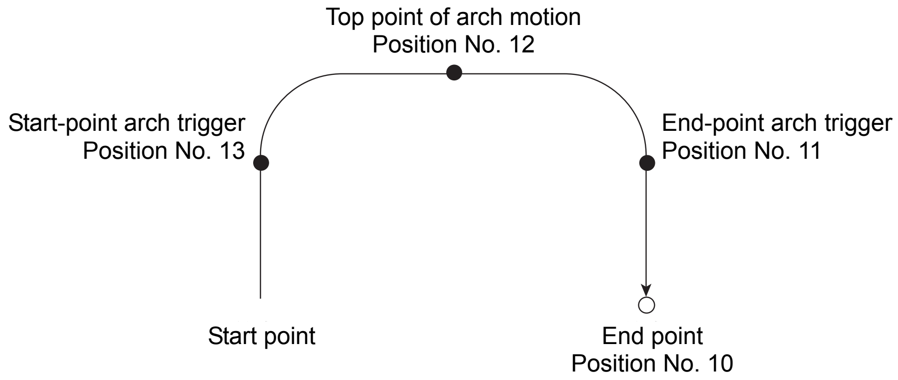
PMVL: Move from the current position to a palletizing point via interpolation.

PACH: Move from the current position to palletizing position via arch motion. You must set palletizing arch motion based on palletizing setting.




```
PCHZ    3
PTRG    11    13
  :
PACH    1     12
```

[Movement command that uses position data as end point...ARCH]
Arch motion is performed to the end point specified by position data.
If the movement is linear in parallel with the actuator, arch motion operation can be possible by specifying only two axes including the applicable axis and PZ-axis.
Arch motion must be set.



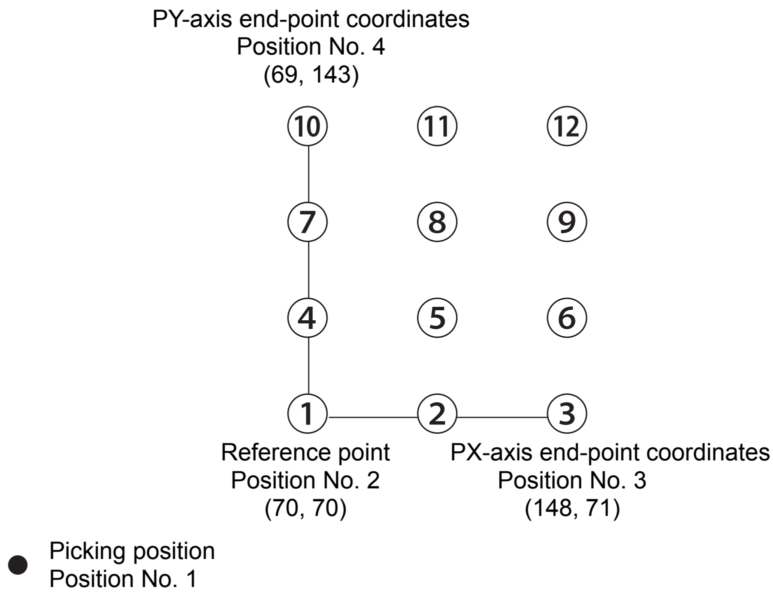

```
ACHZ    3
ATRG    13    11
  ·
  ·
  ·
ARCH    10    12
```

[5] Program Examples

[Simple program example (two-axis specification) using PAPS (set by 3-point teaching)]
The example below specifies movement only and does not cover picking operation.

| Step | E | N | Cnd | Cmnd | Operand1 | Operand2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | BGPA | 1 | | | Start setting palletizing No. 1 |
| 2 | | | | PAPI | 3 | 4 | | Number of palletizing points 3 × 4 |
| 3 | | | | PAPS | 2 | | | Set 3-point teaching |
| 4 | | | | EDPA | | | | End setting palletizing No. 1 |
| 5 | | | | | | | | |
| 6 | | | | VEL | 200 | | | Speed 200mm/sec |
| 7 | | | | MOVL | 1 | | | Move to pick position |
| 8 | | | | PSET | 1 | 1 | | Set palletizing position number to 1 |
| 9 | | | | TAG | 1 | | | |
| 10 | | | | PMVL | 1 | | | Move to palletizing position via interpolation |
| 11 | | | | MOVL | 1 | | | Move to pick position via interpolation |
| 12 | | | | PINC | 1 | | 600 | Increment palletizing position number by 1 |
| 13 | | | 600 | GOTO | 1 | | | Move to beginning of loop if PINC was successful |
| 14 | | | | EXIT | | | | End |

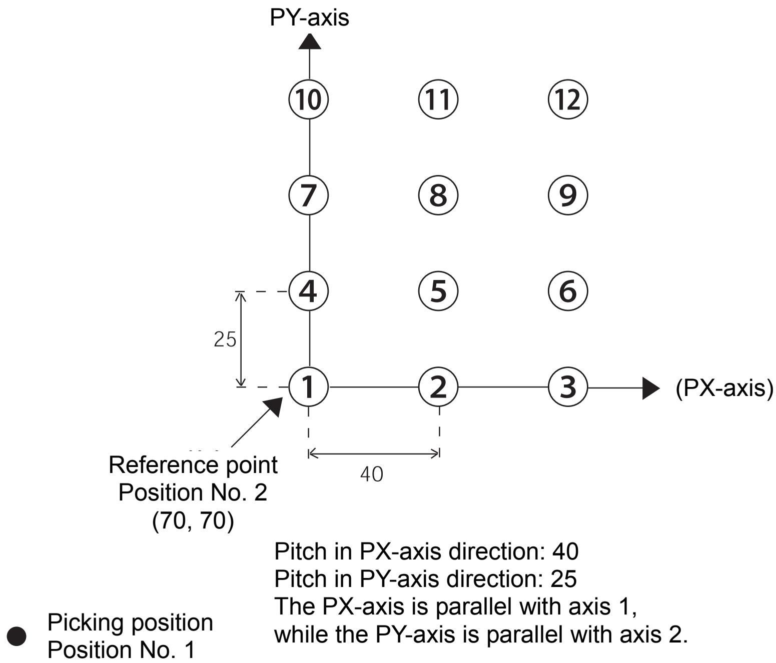
| No. | Axis 1 | Axis 2 | Vel | Acc | Dcl | Remarks |
|---|---|---|---|---|---|---|
| 1 | 10.000 | 10.000 | | | | Pick position |
| 2 | 70.000 | 70.000 | | | | Position data of reference point |
| 3 | 148.000 | 71.000 | | | | Position data PX-axis end point |
| 4 | 69.000 | 143.000 | | | | Position data PY-axis end point |



[Simple program example (two-axis specification) using PAPS, PAPT and PAST]
The example below specifies movement only and does not cover picking operation.

| Step | E | N | Cnd | Cmnd | Operand1 | Operand2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | BGPA | 1 | | | Start setting palletizing No. 1 |
| 2 | | | | PAPI | 3 | 4 | | Number of palletizing points 3 × 4 |
| 3 | | | | PASE | 1 | 2 | | PX-axis = Axis 1, PY-axis = Axis 2 |
| 4 | | | | PAPS | 40 | 25 | | Pitch X = 40, Y = 25 |
| 5 | | | | PAST | 2 | | | Position No. 2 = Reference point |
| 6 | | | | EDPA | | | | End setting palletizing No. 1 |
| 7 | | | | | | | | |
| 8 | | | | VEL | 200 | | | Speed 200mm/sec |
| 9 | | | | MOVL | 1 | | | Move to pick position |
| 10 | | | | PSET | 1 | 1 | | Set palletizing position number to 1 |
| 11 | | | | TAG | 1 | | | |
| 12 | | | | PMVL | 1 | | | Move to palletizing position via interpolation |
| 13 | | | | MOVL | 1 | | | Move to pick position via interpolation |
| 14 | | | | PINC | 1 | | 600 | Increment palletizing position number by 1 |
| 15 | | | 600 | GOTO | 1 | | | Move to beginning of loop if PINC was successful |
| 16 | | | | EXIT | | | | End |

| No. | Axis 1 | Axis 2 | Vel | Acc | Dcl | Remarks |
|---|---|---|---|---|---|---|
| 1 | 10.000 | 10.000 | | | | Pick position |
| 2 | 70.000 | 70.000 | | | | Position data of reference point |




Pitch in PX-axis direction: 40
Pitch in PY-axis direction: 25
The PX-axis is parallel with axis 1,
while the PY-axis is parallel with axis 2.

[Program example using PAPS (set by 3-point teaching)]
The example below specifies movement only and does not cover picking operation.

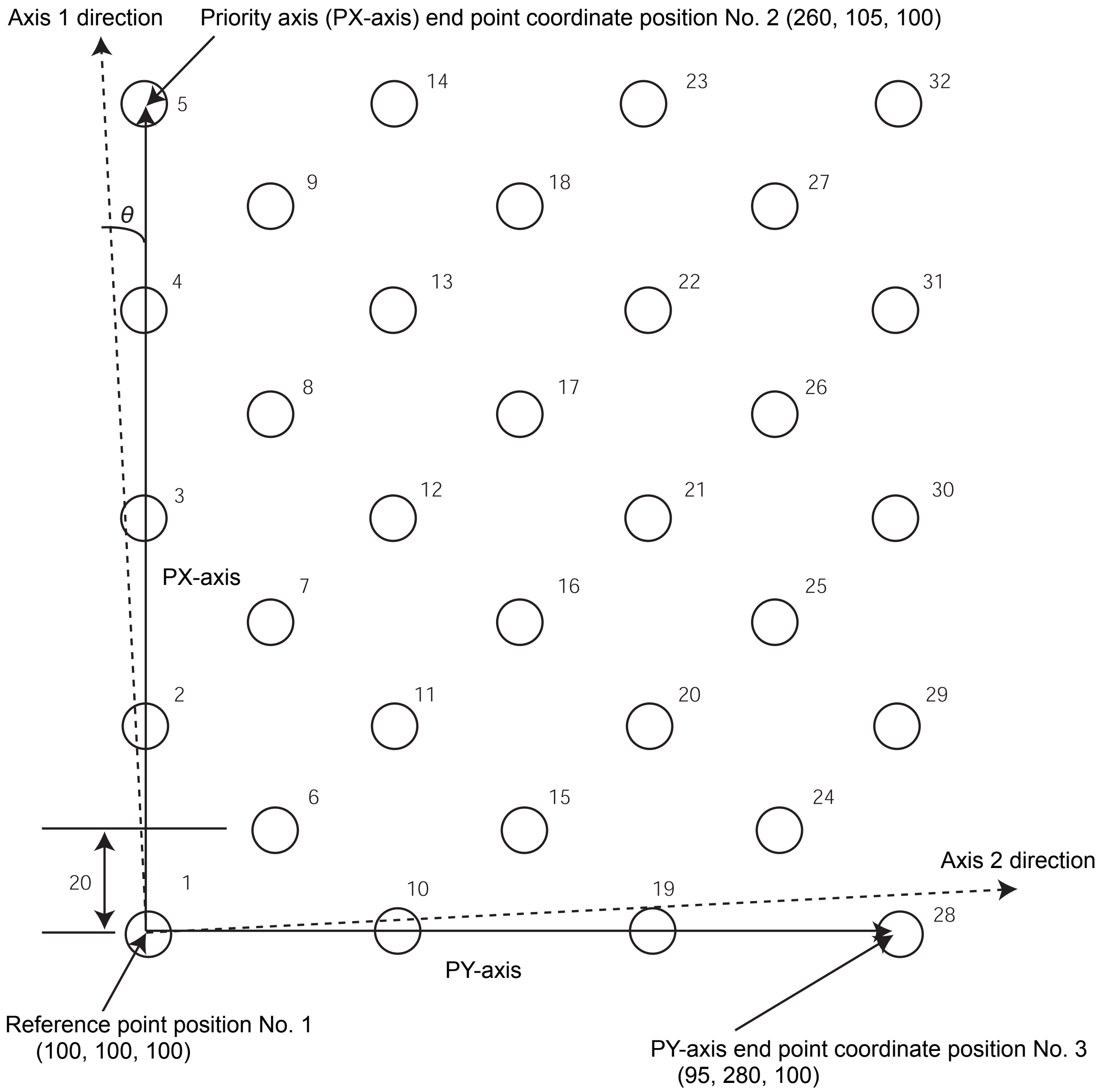
| Step | E | N | Cnd | Cmnd | Operand1 | Operand2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | BGPA | 1 | | | Start setting palletizing No. 1 |
| 2 | | | | | | | | |
| 3 | | | | PAPI | 5 | 7 | | Number of palletizing points 5 × 7 |
| 4 | | | | PAPN | 1 | | | Palletizing pattern 1 |
| 5 | | | | PAPS | 1 | | | Set by 3-point teaching |
| 6 | | | | | | | | Use data of position No. 1 |
| 7 | | | | PSLI | 20 | 4 | | Zigzag offset = 20mm |
| 8 | | | | PCHZ | 3 | | | Palletizing Z-axis = Axis 3 |
| 9 | | | | PTRG | 4 | 4 | | Set palletizing arch trigger |
| 10 | | | | | | | | Use data of position No. 4 |
| 11 | | | | OFPZ | 100 | | | PZ-axis offset = 100mm |
| 12 | | | | PEXT | 6 | | | Set composite palletizing |
| 13 | | | | | | | | Use data of position No. 6 |
| 14 | | | | EDPA | | | | |
| 15 | | | | | | | | |
| 16 | | | | PARG | 1 | 1 | | Get palletizing angle |
| 17 | | | | | | | | The data is stored in variable 199. |
| 18 | | | | PPUT | 4 | 6 | | Store angle data in variable 199 under axis 4 in position No. 6 |
| 19 | | | | | | | | |
| 20 | */////////////////////////////////////////////////////// | | | | | | | |
| 21 | | | | | | | | |
| 22 | | | | ATRG | 4 | 4 | | Set arch trigger |
| 23 | | | | | | | | Use data of position No. 4 |
| 24 | | | | ACHZ | 3 | | | Set arch motion Z-axis |
| 25 | | | | | | | | |
| 26 | | | | ACC | 0.3 | | | Acceleration |
| 27 | | | | DCL | 0.3 | | | Deceleration |
| 28 | | | | VLMX | | | | |
| 29 | | | | | | | | |
| 30 | | | | PSET | 1 | 1 | | Set palletizing position number to 1 |

Continues to the next page

| Step | E | N | Cnd | Cmnd | Operand1 | Operand2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 31 | | | | MOVP | 8 | | | Move to pick position |
| 32 | | | | | | | | |
| 33 | | | | TAG | 1 | | | Beginning of loop process |
| 34 | | | | PACH | 1 | 9 | | Palletizing arch motion |
| 35 | | | | | | | | Z point specified under position No. 9 |
| 36 | | | | ARCH | 8 | 9 | | Arch motion |
| 37 | | | | | | | | Z point specified under position No. 9 |
| 38 | | | | PINC | 1 | | 600 | Increment palletizing position number by 1 |
| 39 | | | 600 | GOTO | 1 | | | Move to beginning of loop if PINC was successful |
| 40 | | | | | | | | |
| 41 | | | | EXIT | | | | End task |
| 42 | | | | | | | | |
| 43 | | | | | | | | |
| 44 | | | | | | | | |
| 45 | | | | | | | | |

| No. | Axis 1 | Axis 2 | Axis 3 | Axis 4 | Remarks |
|---|---|---|---|---|---|
| 1 | 100.000 | 100.000 | 100.000 | *.*** | Position data of reference point |
| 2 | 260.000 | 105.000 | 100.000 | *.*** | Position data PX-axis end point |
| 3 | 95.000 | 280.000 | 100.000 | *.*** | Position data PY-axis end point |
| 4 | *.*** | *.*** | 50.000 | *.*** | Position data for arch trigger |
| 5 | *.*** | *.*** | *.*** | *.*** | (Not used) |
| 6 | *.*** | *.*** | *.*** | –1.79 | Position data for composite palletizing |
| 7 | *.*** | *.*** | *.*** | *.*** | (Not used) |
| 8 | 0.000 | 0.000 | 100.000 | 0.000 | Position data of pick position |
| 9 | *.*** | *.*** | 0.000 | *.*** | Z position data |
| 10 | | | | | |

Schematic diagram of placement point positions according to the program defined earlier




- The number at the top right of each circle indicates the palletizing position number.
- Number of points in PX-axis direction = 5, Number of points PY-axis direction = 7
- Zigzag offsets: 20
- Number of zigzags: 4
- Pallet angle error θ: -1.79°

[Example of program using PASE, PAPT and PAST]
The following program consists of movements only and does not support pick operation.

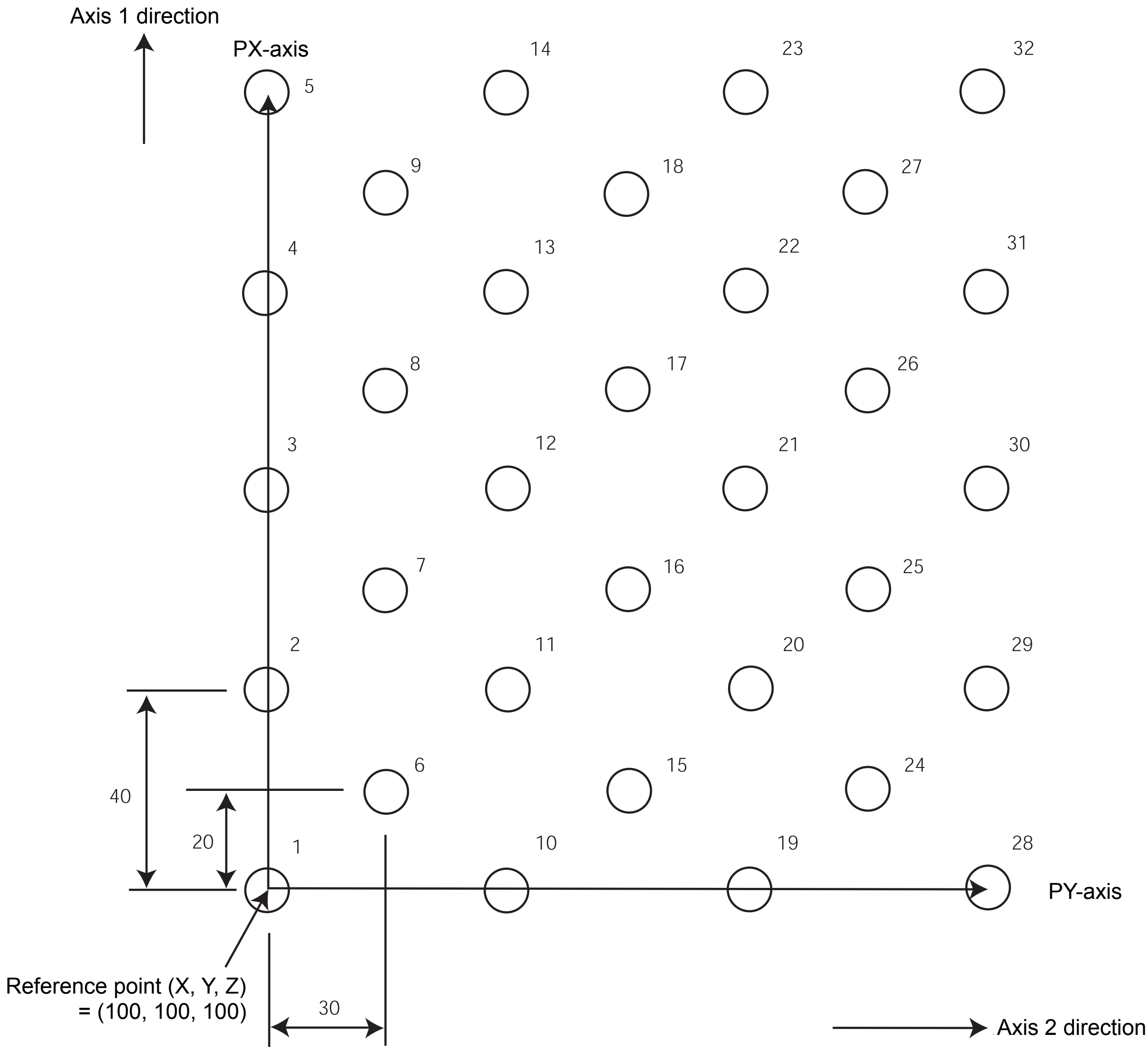
| Step | E | N | Cnd | Cmnd | Operand1 | Operand2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | BGPA | 1 | | | Start setting palletizing No. 1 |
| 2 | | | | | | | | |
| 3 | | | | PAPI | 5 | 7 | | Number of palletizing points 5 × 7 |
| 4 | | | | PAPN | 1 | | | Palletizing pattern 1 |
| 5 | | | | PASE | 1 | 2 | | PX-axis = Axis 1, PY-axis = Axis 2 |
| 6 | | | | PAPT | 40 | 30 | | Pitch (X = 40, Y = 30mm) |
| 7 | | | | PAST | 1 | | | Set reference position data |
| 8 | | | | | | | | Use data of position No. 1 |
| 9 | | | | PSLI | 20 | 4 | | Zigzag offset = 20mm |
| 10 | | | | | | | | Number of zigzags = 4 |
| 11 | | | | PCHZ | 3 | | | Palletizing Z-axis = Axis 3 |
| 12 | | | | PTRG | 4 | 4 | | Set palletizing arch trigger |
| 13 | | | | | | | | Use data of position No. 4 |
| 14 | | | | OFPZ | 100 | | | PZ-axis offset = 100mm |
| 15 | | | | | | | | |
| 16 | | | | EDPA | | | | |
| 17 | | | | | | | | |
| 18 | */////////////////////////////////////////////////////////// | | | | | | | |
| 19 | | | | ATRG | 4 | 4 | | Set arch trigger |
| 20 | | | | | | | | Use data of position No. 4 |
| 21 | | | | ACHZ | 3 | | | Set arch motion Z-axis |
| 22 | | | | | | | | |
| 23 | | | | ACC | 0.3 | | | Acceleration |
| 24 | | | | DCL | 0.3 | | | Deceleration |
| 25 | | | | VLMX | | | | |
| 26 | | | | | | | | |
| 27 | | | | PSET | 1 | 1 | | Set palletizing position number |
| 28 | | | | MOVP | 8 | | | Move to pick position |
| 29 | */////////////////////////////////////////////////////////// | | | | | | | |
| 30 | | | | | | | | |

Continues to the next page

| Step | E | N | Cnd | Cmnd | Operand1 | Operand2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 31 | | | | TAG | 1 | | | Beginning of loop process |
| 32 | | | | PACH | 1 | 9 | | Palletizing arch motion |
| 33 | | | | | | | | Z point specified under position No. 9 |
| 34 | | | | ARCH | 8 | 9 | | Arch motion |
| 35 | | | | | | | | Z point specified under position No. 9 |
| 36 | | | | PINC | 1 | | 600 | Increment palletizing position number by 1 |
| 37 | | | 600 | GOTO | 1 | | | Move to beginning of loop if PINC was successful |
| 38 | | | | | | | | |
| 39 | | | | EXIT | | | | End task |
| 40 | | | | | | | | |

| No. | Axis 1 | Axis 2 | Axis 3 | Axis 4 | Remarks |
|---|---|---|---|---|---|
| 1 | 100.000 | 100.000 | 100.000 | *.*** | Position data of reference point |
| 2 | *.*** | *.*** | *.*** | *.*** | (Not used) |
| 3 | *.*** | *.*** | *.*** | *.*** | (Not used) |
| 4 | *.*** | *.*** | 50.000 | *.*** | Position data for arch trigger |
| 5 | *.*** | *.*** | *.*** | *.*** | (Not used) |
| 6 | *.*** | *.*** | *.*** | *.*** | (Not used) |
| 7 | *.*** | *.*** | *.*** | *.*** | (Not used) |
| 8 | 0.000 | 0.000 | 100.000 | 0.000 | Position data of pick position |
| 9 | *.*** | *.*** | 0.000 | *.*** | Z position data |
| 10 | | | | | |

Schematic diagram of placement point positions according to the program defined earlier




- The number at the top right of each circle indicates the palletizing position number.
- Number of points in PX-axis direction = 5, Number of points PY-axis direction = 7
- Pitch in PX-axis direction: 40
- Pitch in PY-axis direction: 30
- Zigzag offsets: 20
- Number of zigzags: 4

[6] SCARA Robot Program Examples

Notice: It is necessary that the payload for workpiece and chuck is certainly set in the program with WGHT Command for SCARA Robot below.
The acceleration/deceleration can be optimized in relation to the set payload.
In case the setting is not established, the maximum payload should be set, which makes operation slow.

- IXA SCARA Robot
- IX SCARA Robot excludes IX-NNN (NNC) 1205, 1505 and 1805
- IXP PowerCON SCARA Robot

[Refer to “1. Preparation in Advance 1.4.6 SCARA Robot [4] PTP Acceleration/Deceleration Optimization Function”]

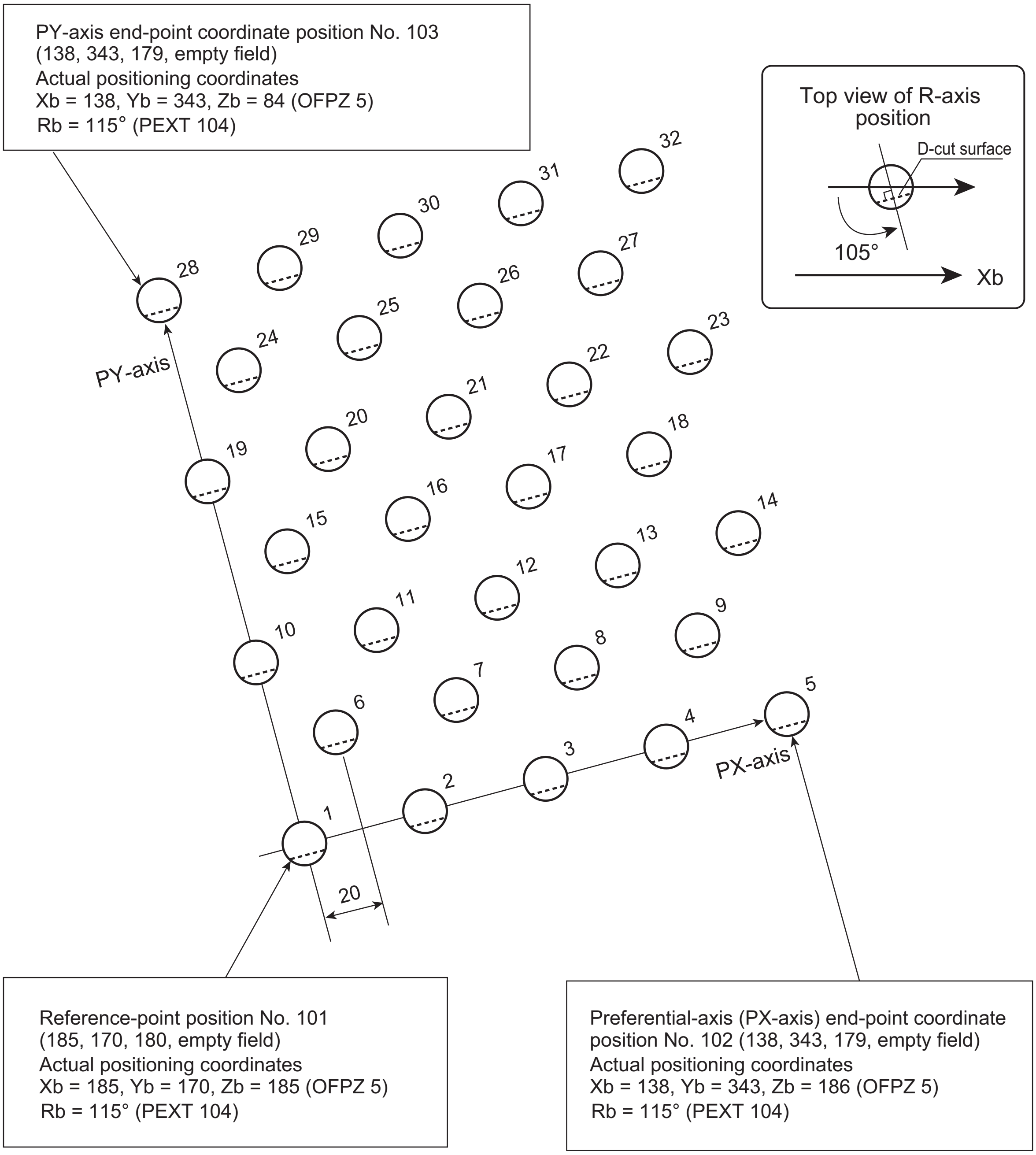
[Program example using PAPS (set by 3-point teaching)]
The example below specifies movement only and does not cover picking operation.

| Step | E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | VELS | 80 | | | PTP travel speed: 80% |
| 2 | | | | ACCS | 50 | | | PTP travel acceleration: 50% |
| 3 | | | | DCLS | 50 | | | PTP travel deceleration: 50% |
| 4 | | | | VEL | 100 | | | CP travel speed: 100 mm/sec |
| 5 | | | | ACC | 0.3 | | | CP travel acceleration: 0.3 G |
| 6 | | | | DCL | 0.3 | | | CP travel deceleration: 0.3 G |
| 7 | | | | WGHT | 1000 | | | Load mass1000g setting |
| 8 | | | | SLWK | 0 | | | Select load coordinate system No. 0. |
| 9 | | | | SLTL | 0 | | | Select tool coordinate system No. 0. |
| 10 | | | | | | | | |
| 11 | | | | BGPA | 1 | | | Start setting palletizing No. 1. |
| 12 | | | | PAPI | 5 | 7 | | Palletizing counts: 5 x 7 |
| 13 | | | | PAPS | 101 | | | Set by 3-point teaching. |
| 14 | | | | PEXT | 104 | | | Set palletizing R-axis coordinate. |
| 15 | | | | PSLI | 20 | 4 | | Zigzag offset = 20 mm |
| 16 | | | | PAPN | 1 | | | Palletizing pattern 1 |
| 17 | | | | PCHZ | 3 | | | Palletizing Z-axis = Axis 3 |
| 18 | | | | PTRG | 105 | 105 | | Set palletizing arch triggers. |
| 19 | | | | OFPZ | 5 | | | PZ-axis offset = 5 mm |
| 20 | | | | EDPA | | | | |
| 21 | | | | | | | | |
| 22 | | | | ATRG | 105 | 105 | | Set arch triggers. |
| 23 | | | | ACHZ | 3 | | | Arch-motion Z-axis = Axis 3 |
| 24 | | | | | | | | |
| 25 | | | | PTPL | | | | Perform positioning in PTP mode using left arm. |
| 26 | | | | MOVP | 110 | | | Move to picking position in PTP mode. |
| 27 | | | | PSET | 1 | 1 | | Set palletizing position number to 1. |
| 28 | | | | TAG | 1 | | | Beginning of loop processing |
| 29 | | | | PACH | 1 | 106 | | Palletizing arch motion |
| 30 | | | | ARCH | 110 | 106 | | Arch motion |
| 31 | | | | PINC | 1 | | 600 | Increment palletizing position number by 1. |
| 32 | | | 600 | GOTO | 1 | | | Beginning of loop when PINC is successful. |
| 33 | | | | MOVL | 109 | | | Move to standby position in CP mode. |
| 34 | | | | EXIT | | | | End |

Position data (Stroke with arm length 500)

| No. | Axis1 | Axis2 | Axis3 | Axis4 | |
|---|---|---|---|---|---|
| 101 | 185.000 | 170.000 | 180.000 | | Reference-point position |
| 102 | 340.000 | 211.000 | 181.000 | | PX-axis end point |
| 103 | 138.000 | 343.000 | 179.000 | | PY-axis end point |
| 104 | | | | 105.000 | Palletizing R-axis position |
| 105 | | | 100.000 | | Arch/palletizing trigger position |
| 106 | | | 80.000 | | Highest position (Z point) |
| 107 | | | | | |
| 108 | | | | | |
| 109 | 0.000 | 160.000 | 0.000 | 0.000 | Standby position |
| 110 | -200.000 | 330.000 | 180.000 | 0.000 | Pickup position |

Schematic diagram of palletizing positions based on the above program




The number shown at top right of each cycle indicates the corresponding palletizing position number.

Count in PX-axis direction = 5, count in PY-axis direction = 7

Zigzag offset: 20, zigzag count: 4

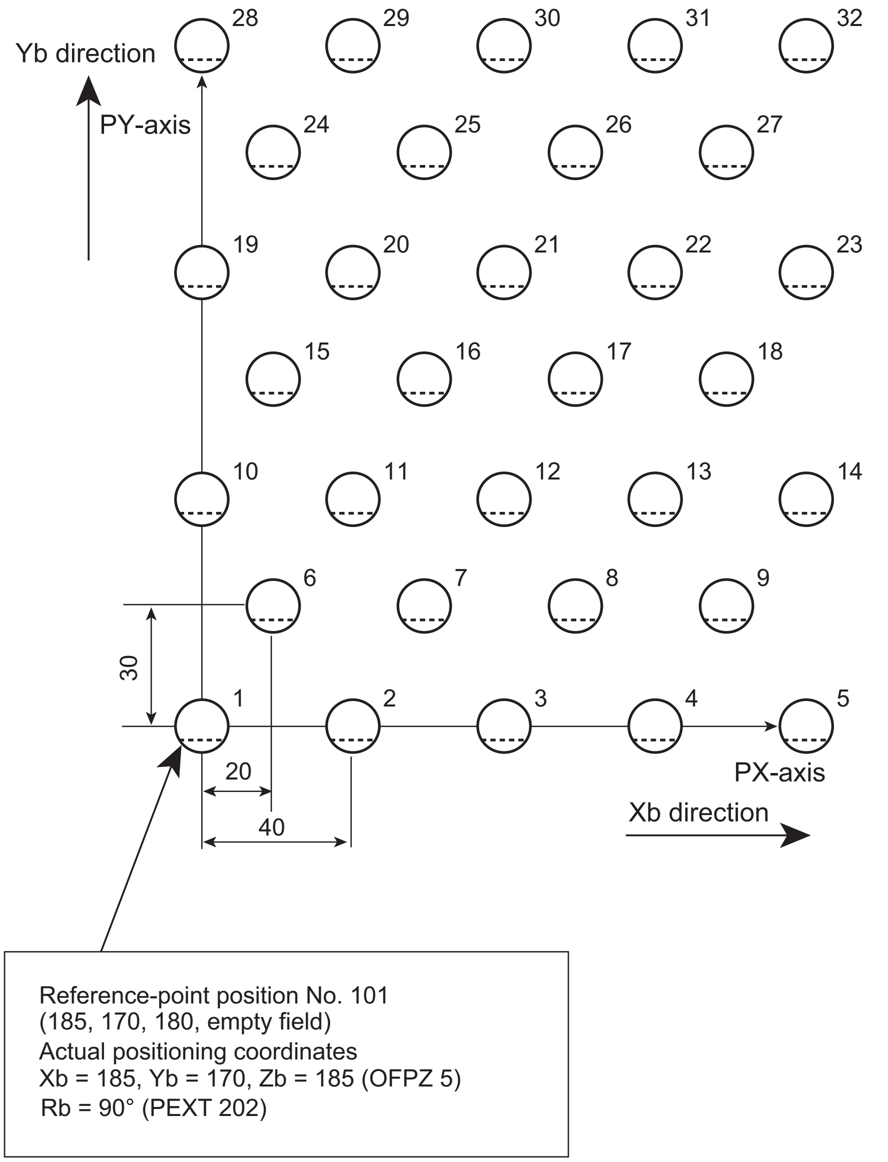
[Program example using PASE, PAPT and PAST]
The example below specifies movement only and does not cover picking operation.

| Step | E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | VELS | 80 | | | PTP travel speed: 80% |
| 2 | | | | ACCS | 50 | | | PTP travel acceleration: 50% |
| 3 | | | | DCLS | 50 | | | PTP travel deceleration: 50% |
| 4 | | | | VEL | 100 | | | CP travel speed: 100 mm/sec |
| 5 | | | | ACC | 0.3 | | | CP travel acceleration: 0.3 G |
| 6 | | | | DCL | 0.3 | | | CP travel deceleration: 0.3 G |
| 7 | | | | WGHT | 1000 | | | Load mass1000g setting |
| 8 | | | | SLWK | 0 | | | Select load coordinate system No. 0. |
| 9 | | | | SLTL | 0 | | | Select tool coordinate system No. 0. |
| 10 | | | | | | | | |
| 11 | | | | BGPA | 1 | | | Start setting palletizing No. 1. |
| 12 | | | | PAST | 201 | | | Set reference-point data. |
| 13 | | | | PASE | 1 | 2 | | PX-axis = X-axis, PY-axis = Y-axis |
| 14 | | | | PAPT | 40 | 30 | | Pitch PX: 40, PY: 30 |
| 15 | | | | PAPI | 5 | 7 | | Palletizing counts: 5 x 7 |
| 16 | | | | PSLI | 20 | 4 | | Zigzag offset = 20 mm, count = 4 |
| 17 | | | | PEXT | 202 | | | Set palletizing R-axis coordinate. |
| 18 | | | | PCHZ | 3 | | | Palletizing Z-axis = Axis 3 |
| 19 | | | | PTRG | 203 | 203 | | Set palletizing arch triggers. |
| 20 | | | | OFPZ | 5 | | | PZ-axis offset = 5 mm |
| 21 | | | | EDPA | | | | |
| 22 | | | | | | | | |
| 23 | | | | ATRG | 203 | 203 | | Set arch triggers. |
| 24 | | | | ACHZ | 3 | | | Arch-motion Z-axis = Axis 3 |
| 25 | | | | | | | | |
| 26 | | | | PTPL | | | | Perform positioning in PTP mode using left arm. |
| 27 | | | | MOVP | 208 | | | Move to picking position in PTP mode. |
| 28 | | | | PSET | 1 | 1 | | Set palletizing position number to 1. |
| 29 | | | | TAG | 1 | | | Beginning of loop processing |
| 30 | | | | PACH | 1 | 204 | | Palletizing arch motion |
| 31 | | | | ARCH | 208 | 204 | | Arch motion |
| 32 | | | | PINC | 1 | | 600 | Increment palletizing position number by 1. |
| 33 | | | 600 | GOTO | 1 | | | Beginning of loop when PINC is successful. |
| 34 | | | | MOVL | 207 | | | Move to standby position in CP mode. |
| 35 | | | | EXIT | | | | End |

Position data (Stroke with arm length 500)

| No. | Axis1 | Axis2 | Axis3 | Axis4 | Description |
|---|---|---|---|---|---|
| 201 | 185.000 | 170.000 | 180.000 | | Reference-point position |
| 202 | | | | 90.000 | Palletizing R-axis position |
| 203 | | | 100.000 | | Arch/palletizing trigger position |
| 204 | | | 80.000 | | Highest position (Z point) |
| 205 | | | | | |
| 206 | | | | | |
| 207 | 0.000 | 160.000 | 0.000 | 0.000 | Standby position |
| 208 | -200.000 | 330.000 | 180.000 | 0.000 | Pickup position |

Schematic diagram of palletizing positions based on the above program
(The PX and PY-axes are parallel with Xb and Yb (base coordinates), respectively.)




The number shown at top right of each cycle indicates the corresponding palletizing position number.
Count in PX-axis direction = 5, count in PY-axis direction = 7
Pitch in PX-axis direction: 40
Pitch in PY-axis direction: 30
Zigzag offset: 20, zigzag count: 4

### 3.6.6 Handling of WAIT Timers

WAIT timers are provided to wait for certain events to occur.
Use a TIMW command to specify waiting.
WAIT timers can be actuated in each program.

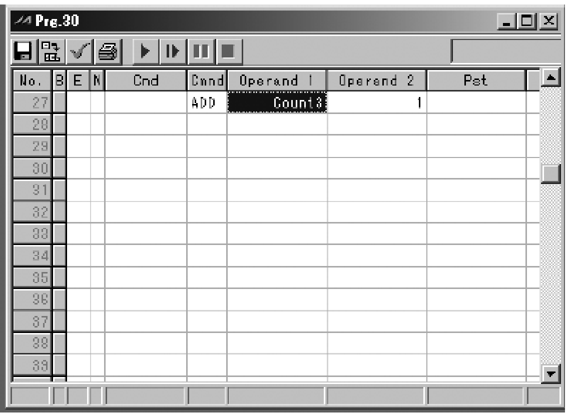
### 3.6.7 Handling of Shot Pulse Timers

Shot pulse timers provide a function to turn ON/OFF an I/O flag for a specified time.
You can use a BTPN command to turn ON an I/O or flag for a specified time.
Similarly, you can use a BTPF command to turn OFF an I/O or flag for a specified time.
The maximum number of shot pulse timers that can be actuated simultaneously in one program is 16 as a total of BTPN and BTPF commands.
Note, however, that there are no limitations to how many times these timers can be used in one program.

### 3.6.8 Handling of Number of Symbol Definitions

With XSEL controllers, you can create a program with ease by using symbols representing variable numbers, flag numbers, etc.
In the example below, variable No. 203 is defined the symbol “Count3” in the symbol edit screen.
The defined symbol can be used in programs, and each statement of “Count3” in programs indicates variable No. 203.


Symbol edit screen




Program edit screen

For information on how to edit symbols, refer to “Editing Symbols” in the XSEL Teaching Pendant Instruction Manual or “Symbol Edit Window” in the XSEL PC Software Instruction Manual.

[1] Scope of support
The following items support use of symbols:
Variable number, flag number, tag number, subroutine number, program number, position number, input port number, output port number, input/output ports number, axis number, constant

[2] Convention of symbol statement
1) Up to nine single-byte alphanumeric or underscore characters, starting from an alphabet. (Note: Up to eight single-byte characters in the case of character string literals)
   * Symbols can also start with an underscore if you are using PC Software Ver.1.1.0.5 or later and Teaching Pendant TB-02(D) : first edition or later, TB-01(D) : first edition or later, SEL-T (D) of Ver.1.04 or later.
   * Among the ASCII codes 21h to 7Eh, those single-byte characters that can be entered from the keyboard can be entered as the second and subsequent characters in a symbol, if you are using PC Software Ver.1.1.0.5 or later.
   * Note that same ASCII codes may be expressed differently if the font used on the PC is different from that used on the teaching pendant (the same also applies to character string literals).
     5Ch ····· PC software: Backslash ＼ (overseas specification, etc.)
     Teaching pendant: Yen symbol ¥
     7Eh ····· PC software: ~
     Teaching pendant: Right arrow →
2) Defining symbols of the same name within the same function is prohibited. (Defining local symbols of the same name in different programs is permitted.)
3) Defining symbols of the same name within the flag number group, input port number group or output port number group, input/output ports number group, is prohibited. (Defining local symbols of the same name in different programs is permitted.)
4) Defining symbols of the same name within the integer variable number group or real variable number group is prohibited.
(Defining local symbols of the same name in different programs is permitted.)
5) Defining symbols of the same name within the integer constant group or real constant group is prohibited.

### 3.6.9 Serial Communication

[1] String processing commands

Strings are character strings. Strings used by the controllers covered by this manual include global strings and local strings.
Global strings can be read or written commonly from any program.
Local strings are valid only within each program and cannot be used in other programs.
Global strings and local strings are differentiated by the range to which their number belongs.

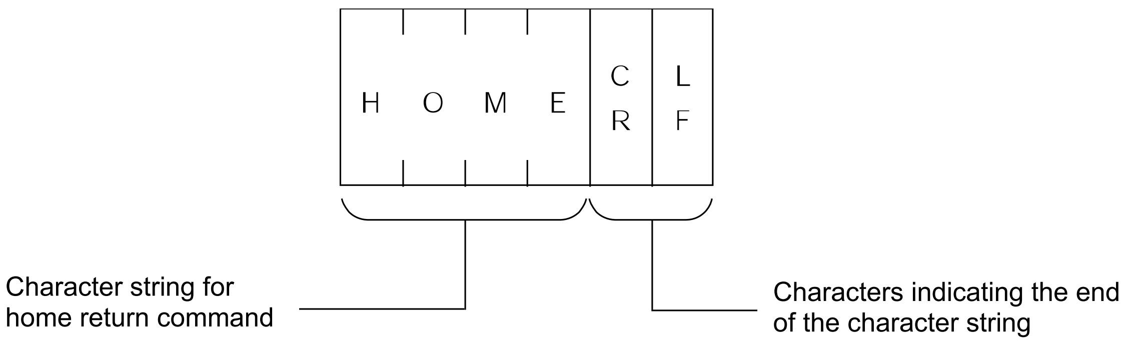
| | |
|---|---|
| Global areas | 300 to 999 (700) |
| Local areas | 1 to 299 (299) |

The communication with the external devices requires to be conducted with the serial communication using character lines, thus a use of the string is required.

[2] Explanation of transmission format

Communication performed by the systems covered by this manual is basically exchange of character strings.
Which character strings should be used for which operations is determined beforehand, so that the receiving side can recognize each character string and perform the corresponding operation.
A combination of these strings and characters indicating the end of one character string is called "transmission format", and the user can determine a desired transmission format freely.
For example, assume a character string consisting of four characters "HOME" which is used as a home return command.
It is determined the character to finish the character line should be either "CR" or "LF" on PC.
Therefore, it is necessary to follow this rule.

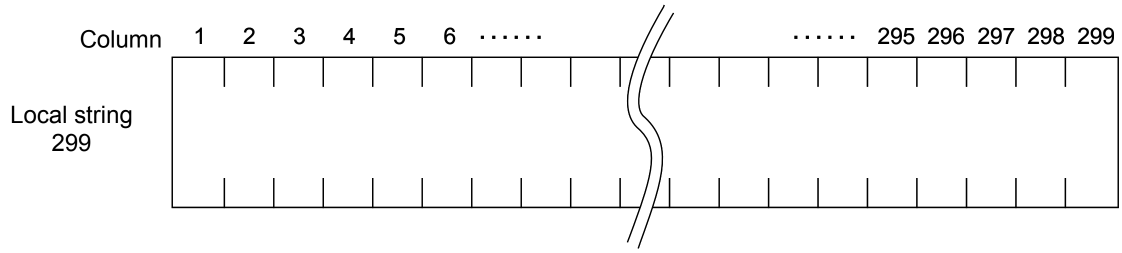
Example of transmission format

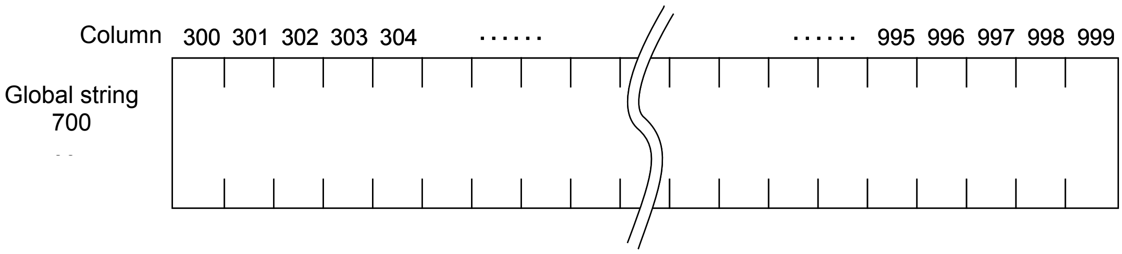


[3] Explanation of string

Strings sent according to the format explained above are stored in boxes designed to contain character strings, so that they can be used freely in the program.

Two types of strings are available: global strings that can be read or written in all programs, and local strings that can be red or written only in each program. Both strings are differentiated by their column numbers.

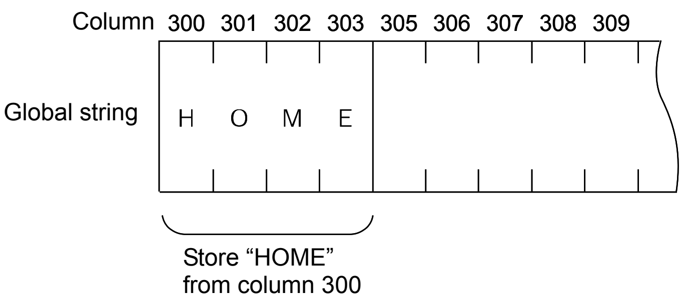



One character is stored in each of the fields of these strings.

The position of a given field comprising a string is indicated by “column XX” and which column to store can be set freely for each command.

For instance, if a character line “HOME”, which indicates the home-return command, is received, and the character line is desired to be used in several programs, you should save the data to Column 300 in the global string.



[4] Determination of transmission format

In this example of application program, three types of transmission formats are required, or namely transmission formats for home return command, movement command and movement completion. These formats are determined as follows. Note that these are only examples and the user can freely determine each format.

A. Home return command format

This format is used to issue a home return command from the PC to the controller.

| | | | | | |
|---|---|---|---|---|---|
| H | O | M | E | CR | LF |

B. Movement command format

This format is used to issue an axis movement command from the PC to the controller.

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M | O | V | E | Speed | | | Axis 1 position | | | | | | | Axis 2 position | | | | | | | CR | LF |
| | | | | 9 | 9 | 9 | 9 | 9 | 9 | . | 9 | 9 | 9 | 9 | 9 | 9 | . | 9 | 9 | 9 | | |

C. Movement completion format

This format is sent from the controller to the PC when the home return or movement is completed.

| | | | |
|---|---|---|---|
| O | K | CR | LF |

[5] Processing procedure

The processing procedure you should follow to program this application example is explained.

A. Set “LF” as characters (terminator characters) indicating the end of a string.
B. Open channel 1 of the RS232 unit to use this channel 1.
C. Program so that any data, if sent through channel 1, is received into columns starting from column 1 for local strings.
D. Program so that if the received data is “MOVE”, the applicable speed data is converted to a binary value and the converted binary value is set in variable 10, while the applicable position data is converted to a binary value and the obtained binary value is set in position No. 1, after which the actuator moves and when the movement is completed, “OK” is sent.

[6] Application program

| STEP | No. | N | OP-CODE | OPRND1 | OPRND2 | POST | Comment |
|---|---|---|---|---|---|---|---|
| 1 | | | SCHA | 10 | | | Set LF as terminator characters |
| 2 | | | OPEN | 1 | | | Open SIO channel 1 |
| 3 | | | TAG | 1 | | | |
| 4 | | | READ | 1 | 1 | | Read into columns starting from SIO 1 column 1 |
| 5 | | | | | | | |
| 6 | | | ISEQ | 1 | ‘HOME’ | | If Home return command |
| 7 | | | HOME | 11 | | | Home return |
| 8 | | | EXSR | 1 | | | Send OK |
| 9 | | | EDIF | | | | |
| 10 | | | | | | | |
| 11 | | | ISEQ | 1 | ‘MOVE’ | | If movement command: |
| 12 | | | SLEN | 3 | | | Reading period with three digits |
| 13 | | | VAL | 10 | 5 | | Set speed in variable 10 |
| 14 | | | VEL | *10 | | | Set speed |
| 15 | | | | | | | |
| 16 | | | PCLR | 1 | 1 | | Clear position 1 |
| 17 | | | SLEN | 3.3 | | | |
| 18 | | | VAL | 199 | 8 | | Set axis 1 position in variable 199 |
| 19 | | | PPUT | 1 | 1 | | Set axis 1 data |
| 20 | | | | | | | |
| 21 | | | VAL | 199 | 15 | | Set axis 2 position in variable 199 |
| 22 | | | PPUT | 2 | 1 | | Set axis 2 data |
| 23 | | | MOVL | 1 | | | Move |
| 24 | | | EXSR | 1 | | | Send OK |
| 25 | | | EDIF | | | | |
| 26 | | | | | | | |
| 27 | | | GOTO | 1 | | | |
| 28 | | | | | | | ----------- |
| 29 | | | BGSR | 1 | | | OK send subroutine |
| 30 | | | SCPY | 1 | ‘OK’ | | Set OK |
| 31 | | | SPUT | 3 | 13 | | Set CR |
| 32 | | | SPUT | 4 | 10 | | Set LF |
| 33 | | | WRIT | 1 | 1 | | Send |
| 34 | | | EDSR | | | | |

[7] Number of SIO Channels for each Controller

The channel numbers of SIO channels used in RS232C serial communication are as shown below.
Use OPEN and CLOS commands to specify SIO channel numbers that are used to open and close the RS232C serial communication line.
How many SIO channel numbers are available varies depending on the controller.

| Controller | SIO channel number |
|---|---|
| XSEL-P/Q/PCT/QCT/PX/QX/R/ S/RX/SX/RXD/SXD | 1 to 2 |
| XSEL-RA/SA/RAX/SAX/RAXD/ SAXD | 1 |
| XSEL-J/JX (1, 2-axis specification) TT | 1*1 |
| XSEL-J/JX (3, 4-axis specification) XSEL-K/KE/KT/KET, KX/KETX, TTA, MSEL | 1*1 *2 |
| SSEL, ASEL, PSEL | 0*1 |

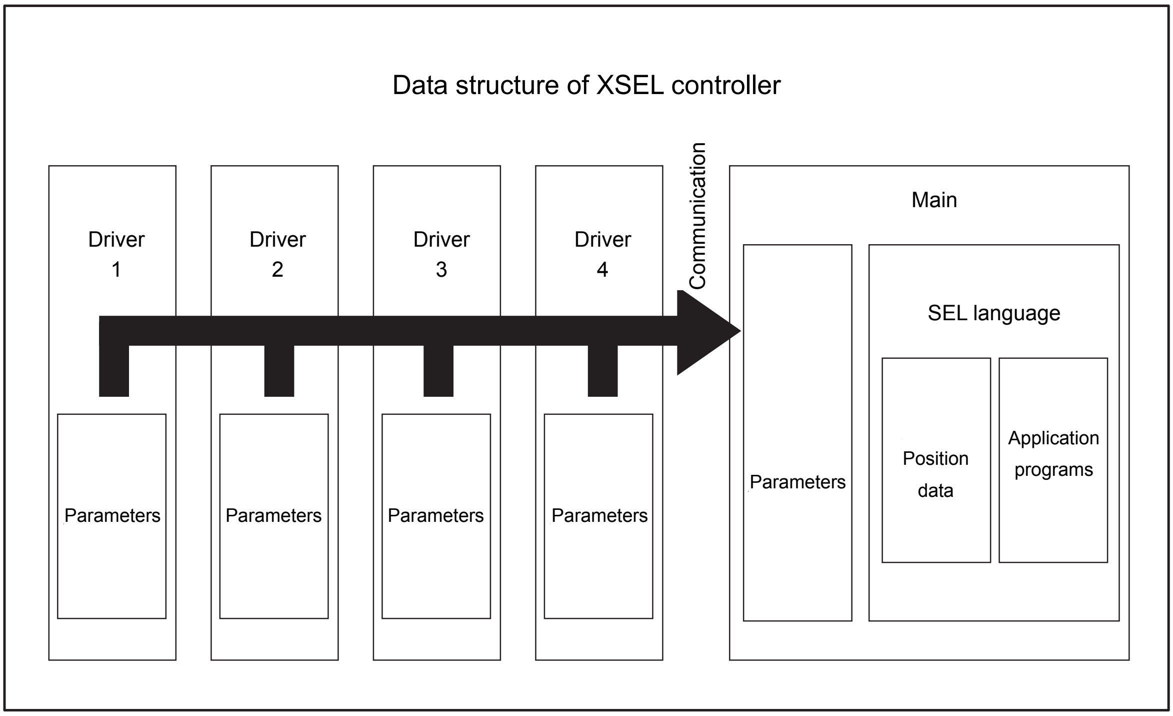
*1 This channel is used as the teaching-pendant connector port.

*2 If an expansion SIO board is installed, No. 2 and subsequent channels can be used.

## 3.7 Controller Data Structure and Saving of Data

### 3.7.1 XSEL-J/K/KE/KT/KET, JX/KX/KETX

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.




The customer must create position data and application programs.
Certain parameters can be changed according to the customer's system.

[2] Saving of data

XSEL controllers have areas saved by the backup battery and areas saved by the flash memory.

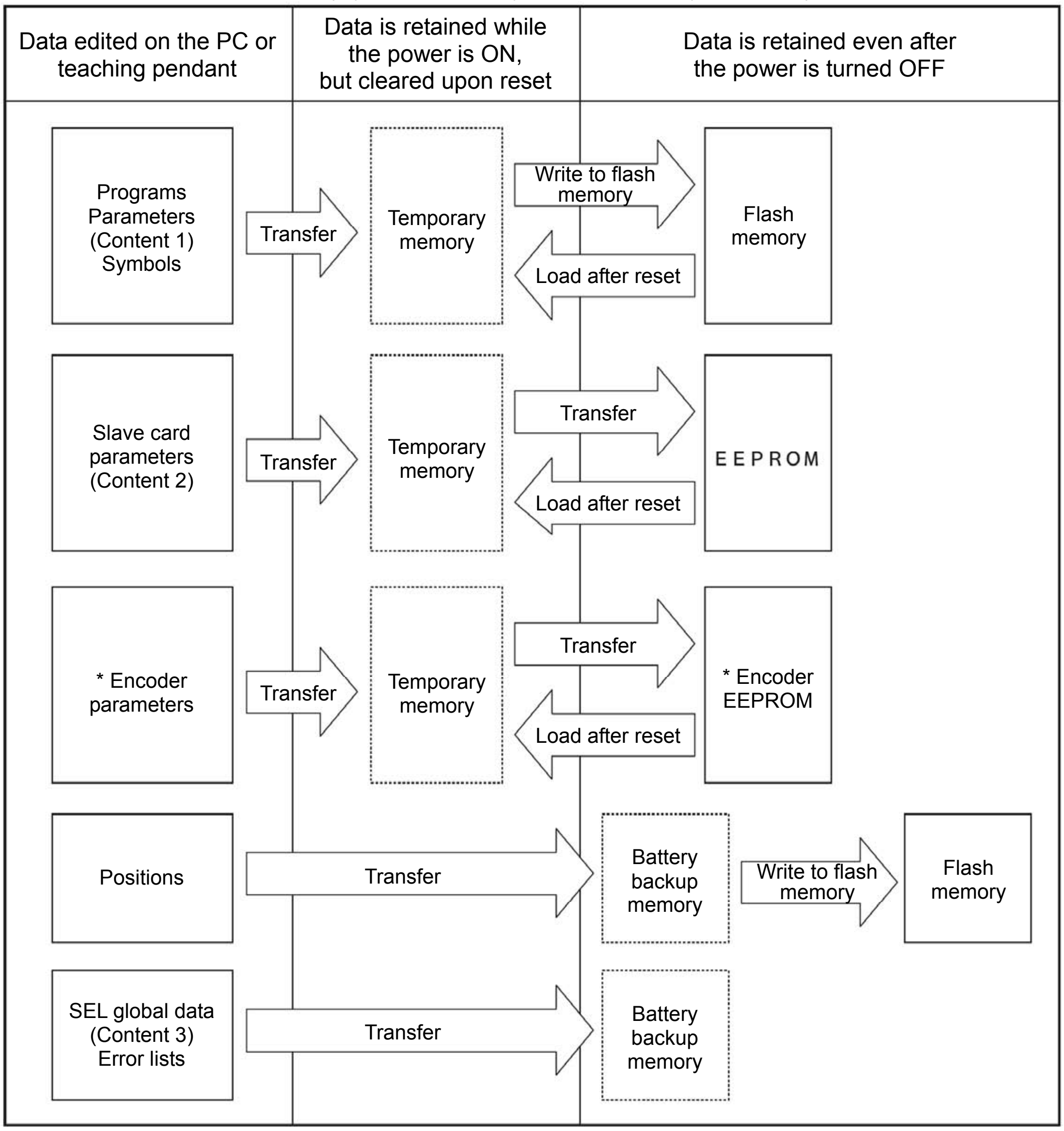
Also note that even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.

So that your important data is saved without fail, write the data to the flash memory.



[System-memory backup battery is used]

Other parameter No. 20 = 2 (System-memory backup battery installed)




* Encoder parameters are stored not in the controller, but in the EEPROM of the actuator's encoder. Accordingly, they are loaded to the controller when the power is turned on or software is reset.

Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in each temporary memory (dotted box) (excluding parameters).

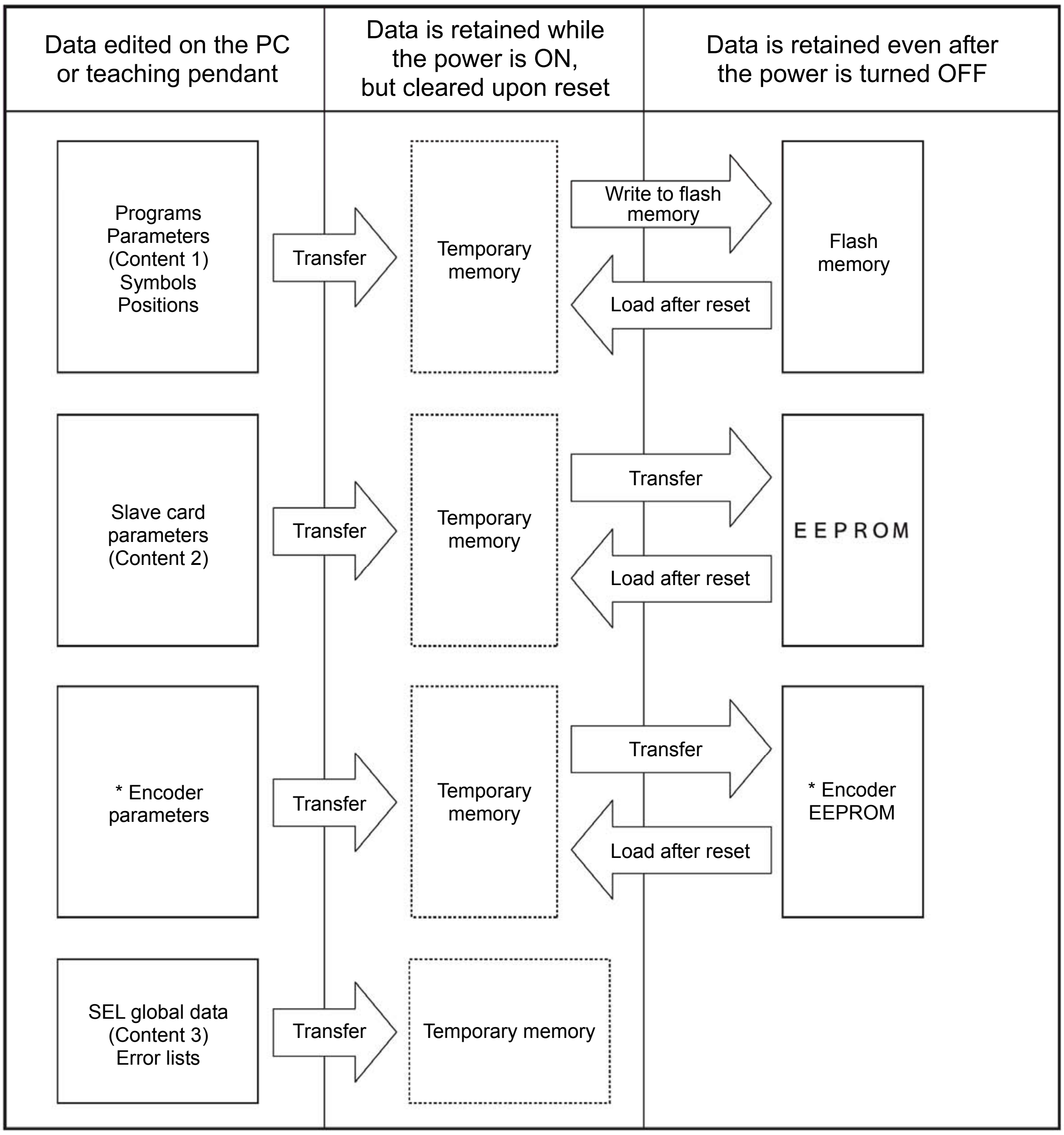
Content 1: Parameters other than those included in Content 2 below and encoder parameters

Content 2: Driver card and I/O slot card (power card) parameters

Content 3: Flags, variables and strings

[System-memory backup battery is not used]

Other parameter No. 20 = 0 (System-memory backup battery not installed)




Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in each temporary memory (dotted box) (excluding parameters).

Note: SEL global data cannot be retained unless the backup battery is installed.

[3] Notes

⚠ **Caution**

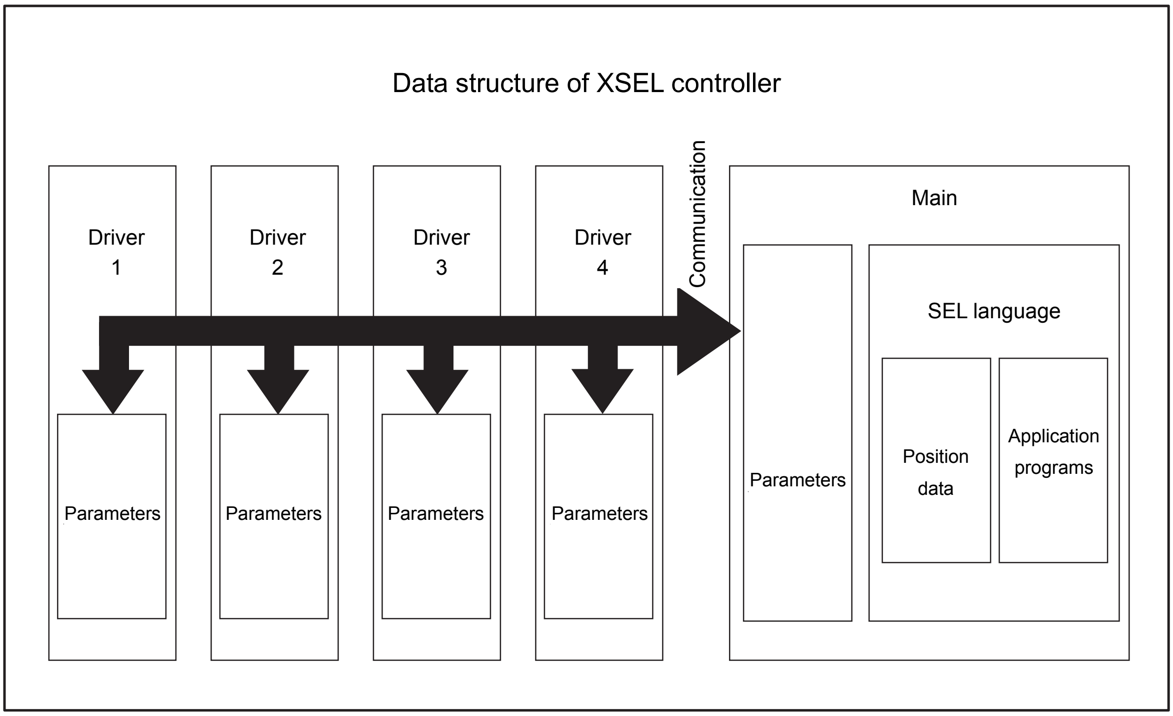
- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned on or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

### 3.7.2 XSEL-P/Q/PCT/QCT, PX/QX

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.

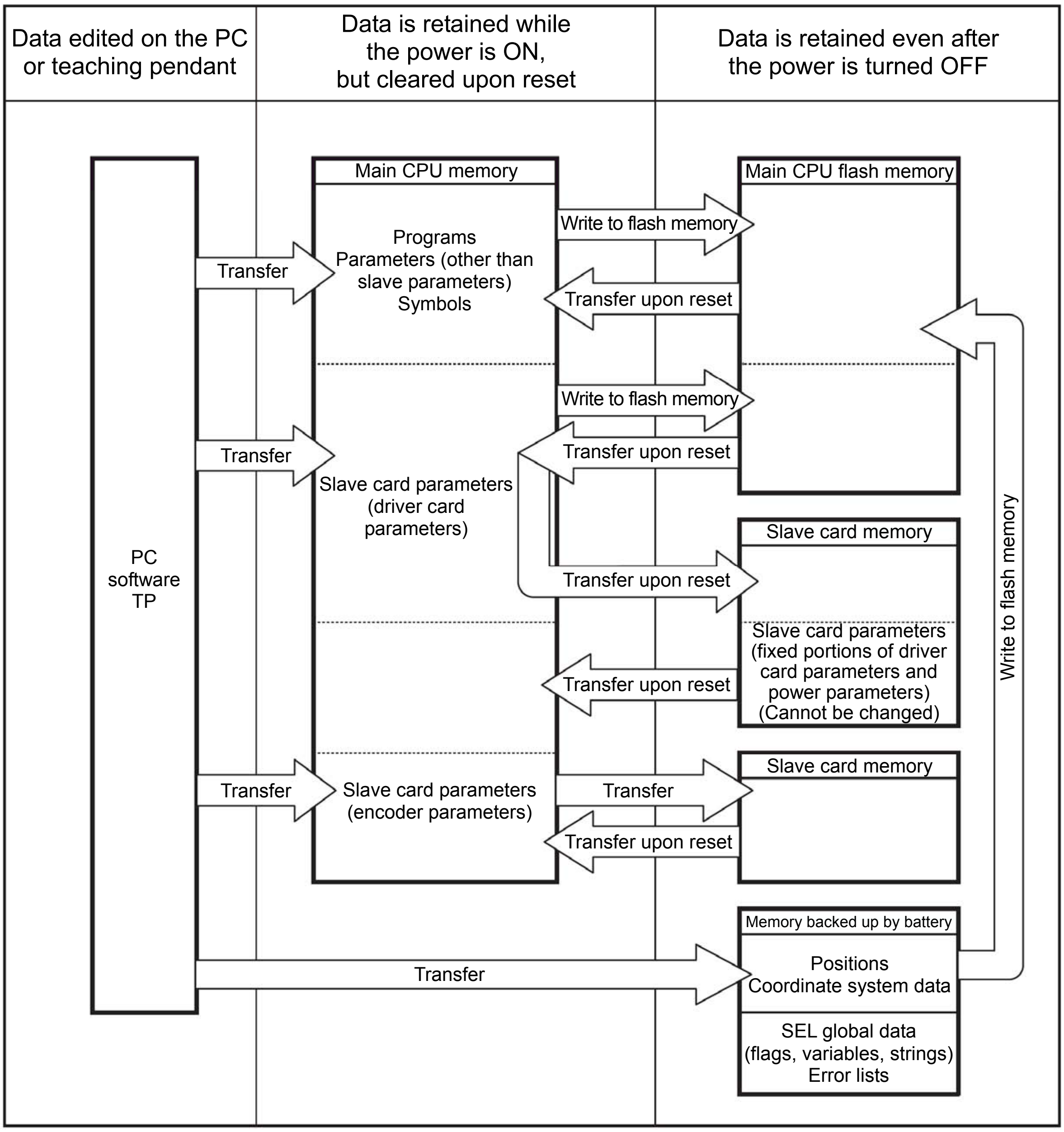



The customer must create position data and application programs. Certain parameters can be changed according to the customer's system.

[2] Saving of data
XSEL controllers have areas saved by the backup battery and areas saved by the flash memory.
Also note that even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.
So that your important data is saved without fail, write the data to the flash memory.



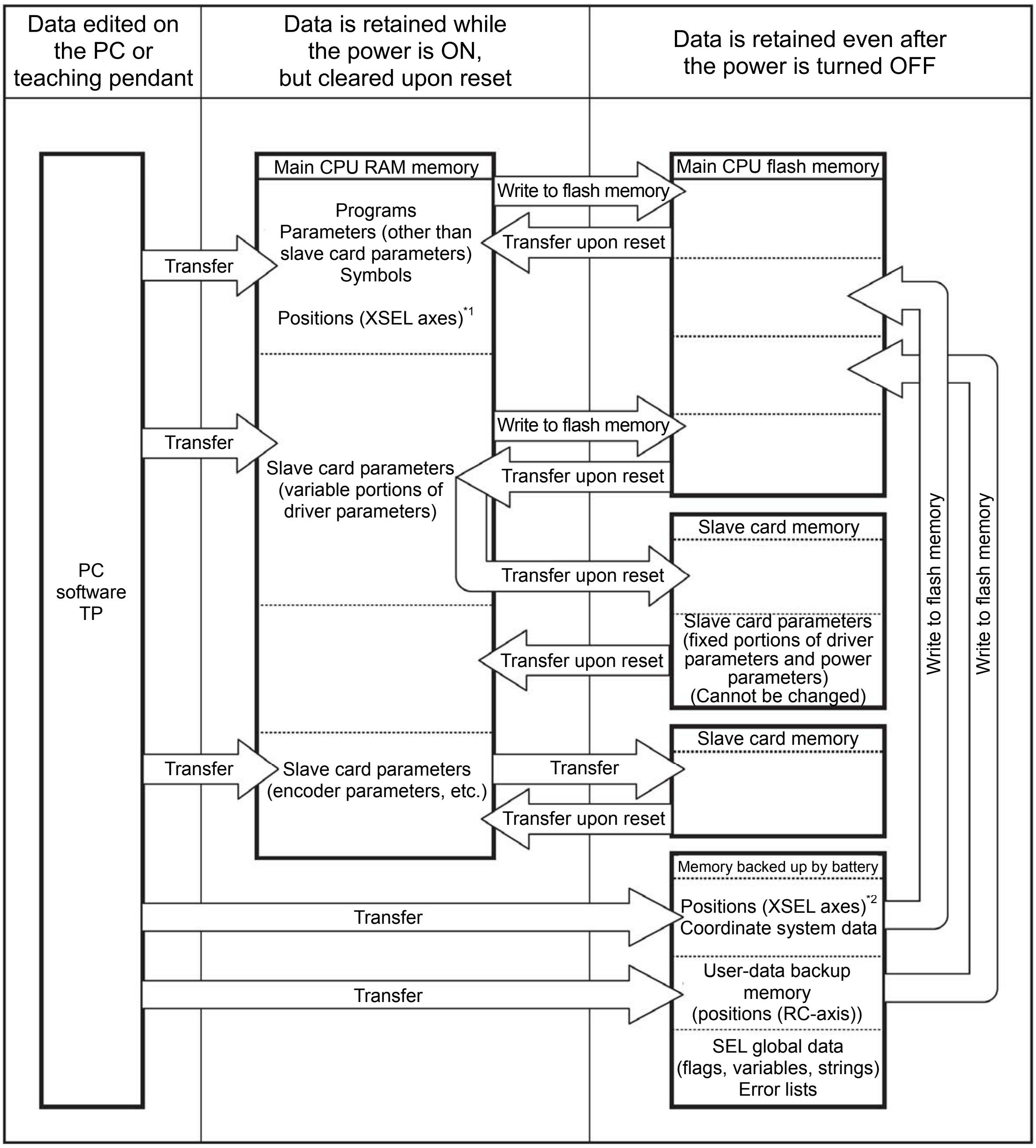
[System-memory backup battery is used]
1) XSEL-P/Q/PCT/QCT, PX/QX
(gateway function + 5V supply switch not available, memory capacity 16M)
Other parameter No. 20 = 2 (System-memory backup battery installed)




Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).

2) XSEL-P/Q/PCT/QCT, PX/QX
(gateway function + 5V supply switch available, memory capacity 32M)

Other parameter No. 20 = 2 (System-memory backup battery installed)



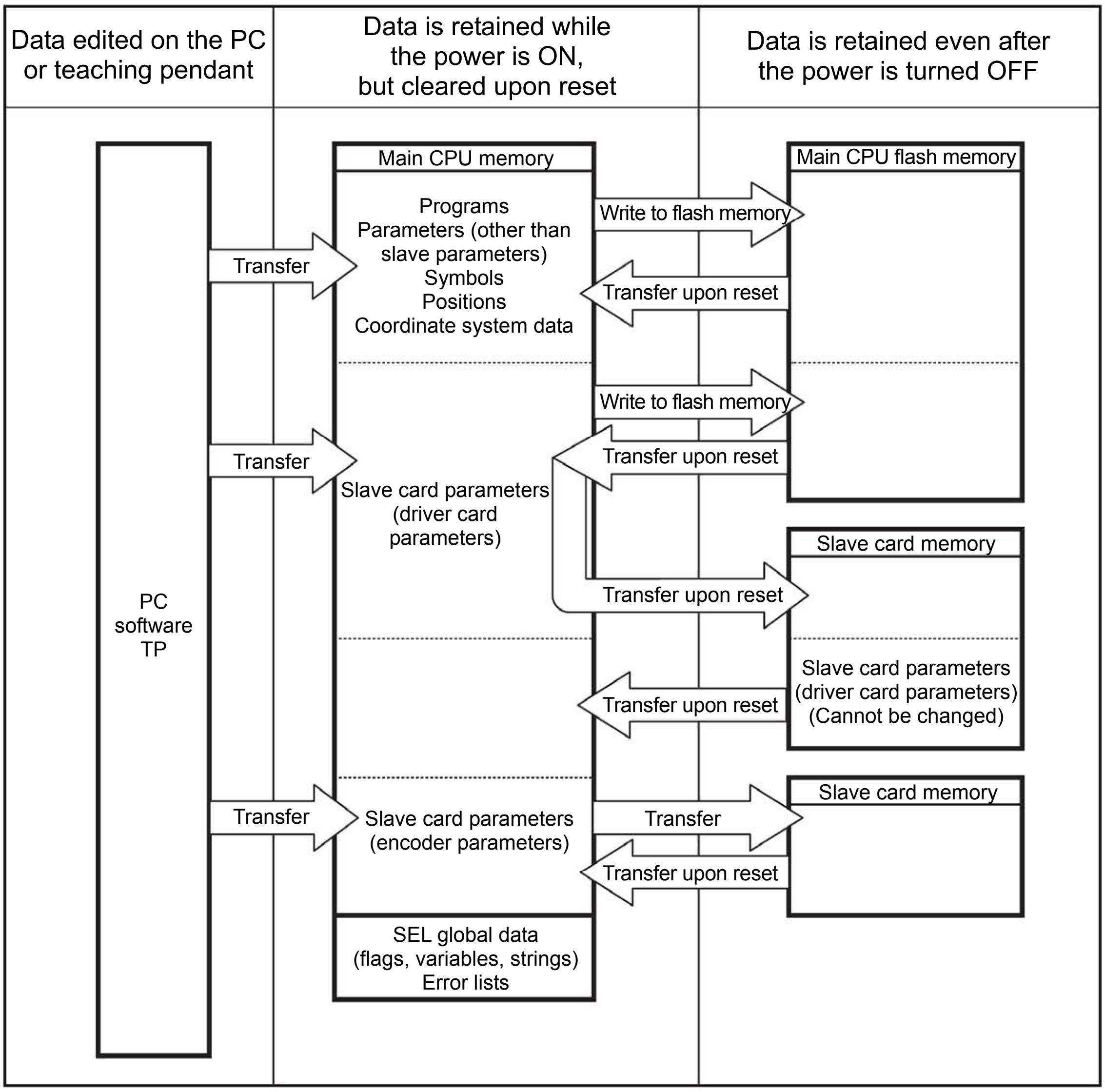

Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).

*1 XSEL-P/Q/PCT/QCT and PX/QX controllers support No. 10001 to 20000.
*2 XSEL-P/Q/PCT/QCT and PX/QX controllers support No. 1 to 10000.

[System-memory backup battery is not used]

1) XSEL-P/Q/PCT/QCT, PX/QX
(gateway function + 5V supply switch not available, memory capacity 16M)

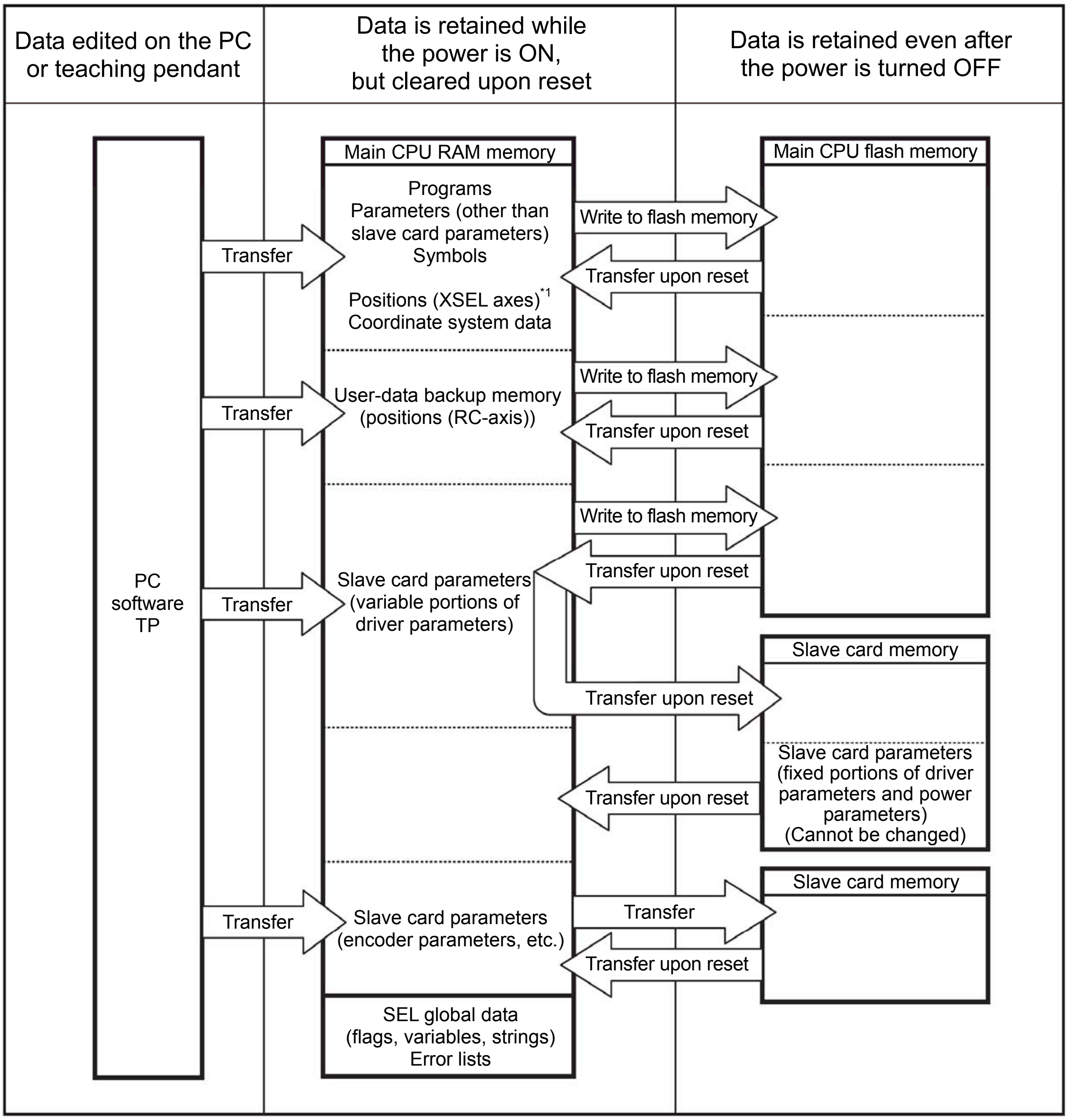
Other parameter No. 20 = 0 (System-memory backup battery not installed)




Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in the main CPU memory (excluding parameters).
Note: SEL global data cannot be retained unless the backup battery is installed.

2) XSEL-P/Q/PCT/QCT, PX/QX
(gateway function + 5V supply switch available, memory capacity 32M)

Other parameter No. 20 = 0 (System-memory backup battery installed)




Since programs, parameters, symbols, positions and user-data backup memory are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in the main CPU memory (excluding parameters).
Note: SEL global data cannot be retained unless the backup battery is installed.

*1 XSEL-P/Q/PCT/QCT and PX/QX controllers support No. 1 to 20000.

[3] Notes

⚠ **Caution**

- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

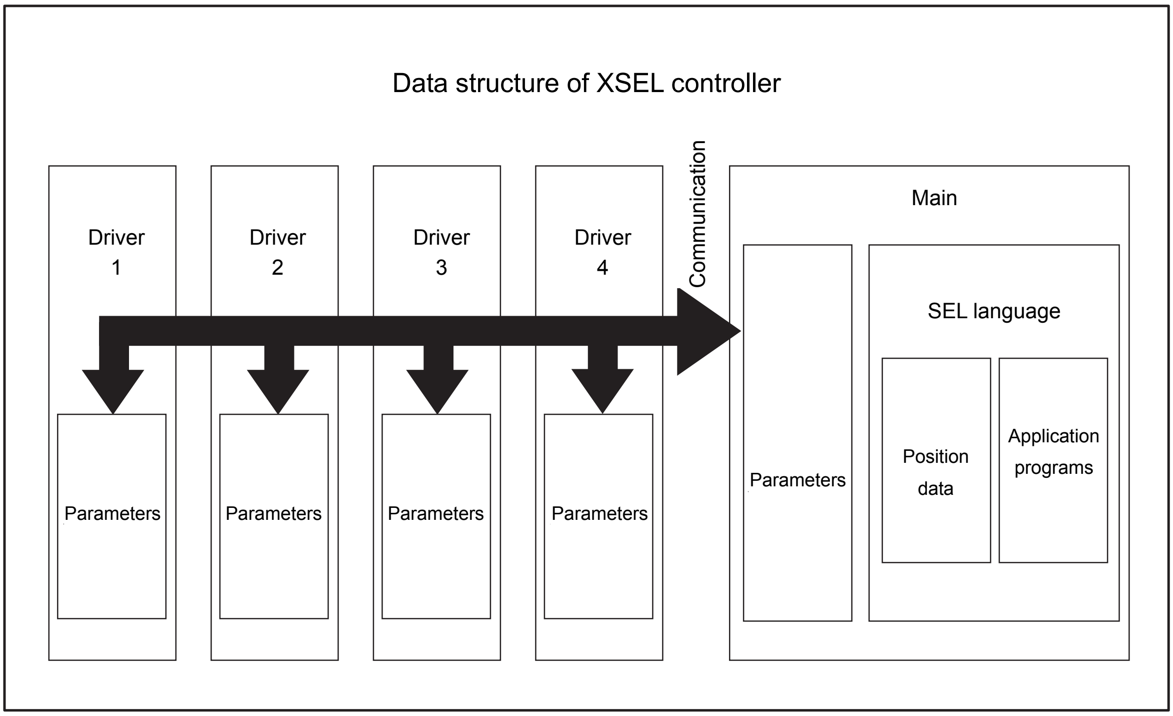
- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

- **Notes on increased number of positions**
  On controllers with increased memory capacity (with gateway function), the number of position data points has increased to 20000.
  Accordingly, take note of the following points:
  * If the memory backup battery is used (other parameter No. 20 = 2), position data is saved in the memory backup battery for position No. 1 to 10000, and in the main CPU flash ROM for position No. 10001 to 20000. Accordingly, turning OFF the power or resetting the software without writing the position data to the flash ROM will cause the data of position No. 10001 to 20000 to be cleared and the data previously written to the flash ROM will be loaded the next time the controller is started. To retain your data, therefore, make sure you write it to the flash ROM. If the memory backup battery is not used (other parameter No. 20 = 2), all position data of No. 1 to 20000 is saved in the main CPU flash ROM. In this case, again, write your data to the flash ROM to make sure the data is retained.

### 3.7.3 XSEL-R/S/RX/SX/RXD/SXD

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.



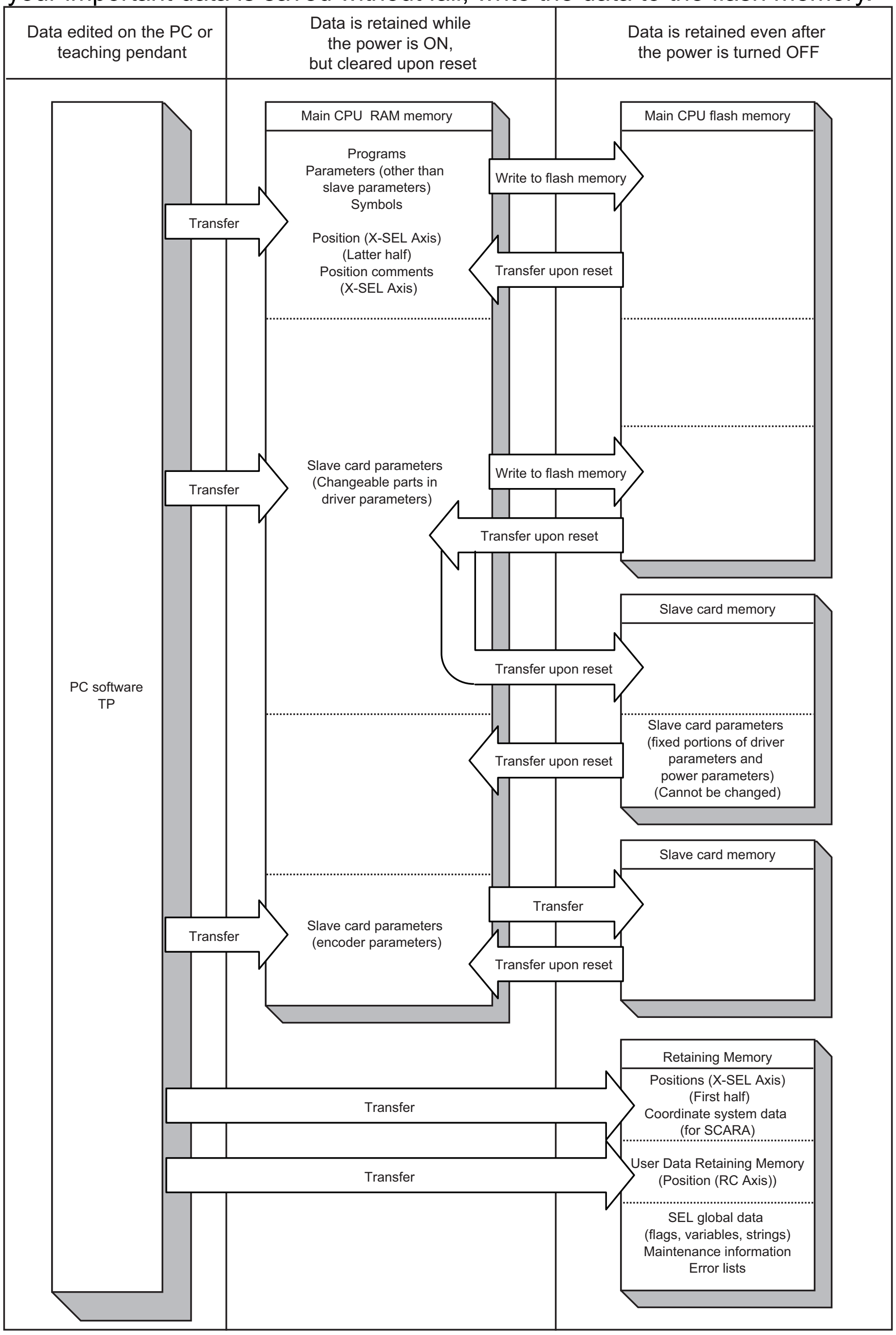

The customer must create position data and application programs. Certain parameters can be changed according to the customer's system.

[2] Saving of data

In XSEL controller, there is a storage domain with saving memory and a storage domain with flash memory.

Also note that even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.

So that your important data is saved without fail, write the data to the flash memory.




Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.

The controller always operates according to the data in each temporary memory (excluding parameters).

Note: The first half of the position data is stored in the saving memory while the second half in flash memory. The comment for each position data can be used in Positions No. 1 to 10000, and it is saved in the flash memory.

[3] Notes

⚠ **Caution**

- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

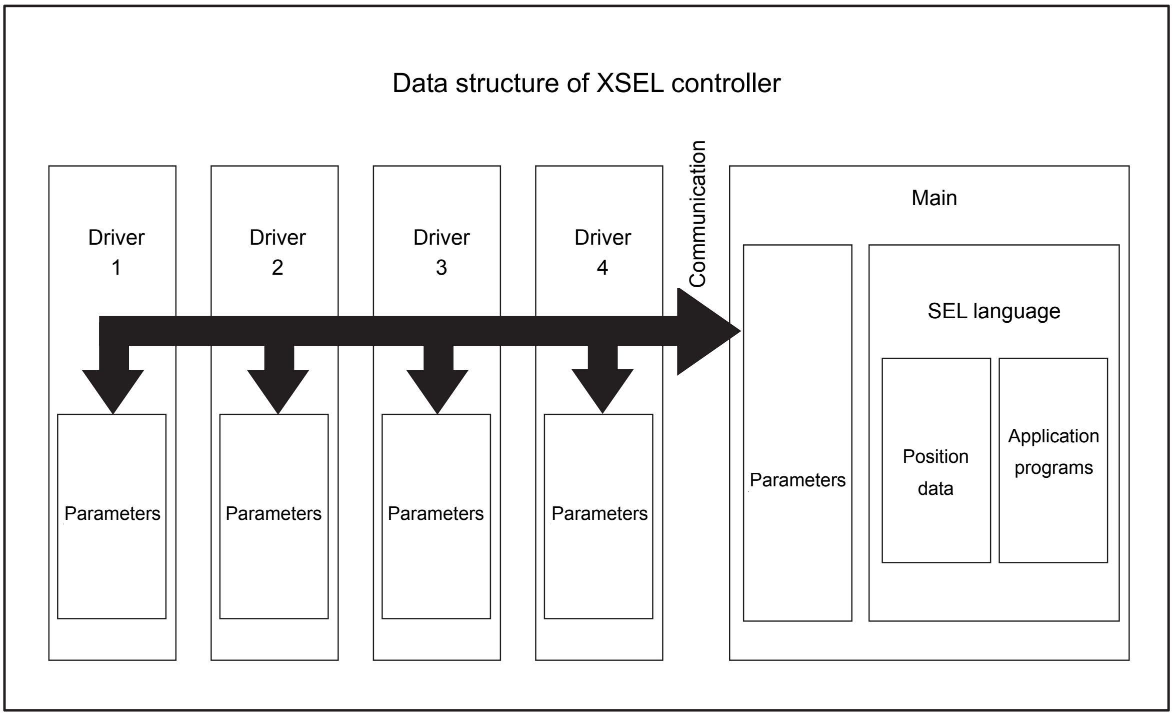
- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

- **Regarding Position Data Save**
  The storage domain for the position data is saving memory for the position (first half) and flash ROM of the main CPU for the position (second half). All the position data comment is to be stored in the flash ROM of the main CPU. Therefore, if the power is turned OFF or the software reset is conducted before writing to the flash ROM, the position (second half) and the position comment data are deleted, and the data that was previously written to the flash ROM is read out the next time the system is turned on. Do not fail to conduct the flash ROM writing when data saving is required.

- **About Initializing of Memory**
  Because the position data, maintenance information data and SEL global data will not be initialized (error data remains) even after an error is detected, make sure not to use the data without canceling it. To cancel an error, initialize the memory of the data which an error has been detected.
  For the position data (second half), do not fail to conduct the flash ROM writing at the same time after initializing.
  (Reference) How to Initialize Memory
  - Position Data: Select [Memory Initialization] → [Position Data] Menu in the PC software
  - Coordinate System Data: Select [Memory Initialization] → [Coordinate System Definition Data] Menu in the PC software
  - User Retaining Memory: Select [Memory Initialization] → [User Retaining Memory] Menu in the PC software
  - SEL Global Data: Select [Memory Initialization] → [Global Variables/Flags] Menu in the PC software
  - Maintenance Information Data: Select [Memory Initialization] → [Maintenance Information] in the PC software and select [Information Initialization]
    * Initialization available when Error No. 4A4, 4A5 or 4A6 has occurred

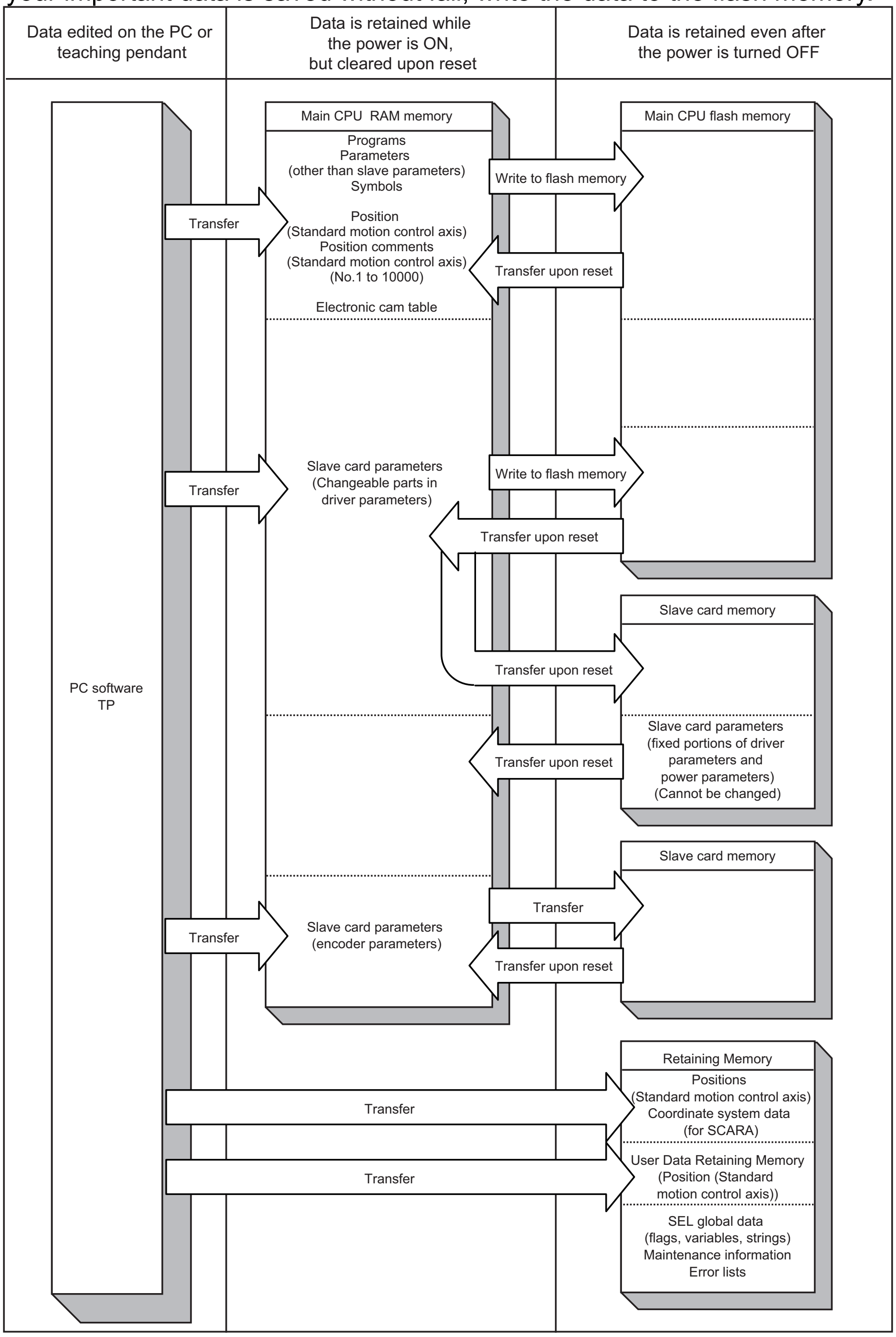
### 3.7.4 XSEL-RA/SA/RAX/SAX/RAXD/SAXD

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.




The customer must create position data and application programs. Certain parameters can be changed according to the customer's system.

[2] Saving of data
In XSEL controller, there is a storage domain with saving memory and a storage domain with flash memory.
Also note that even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.
So that your important data is saved without fail, write the data to the flash memory.




Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).
Note: The first half of the position data is stored in the battery-less backup memory while the second half in flash memory. The comment for each position data can be used in Positions No. 1 to 10000, and it is saved in the flash memory.

[3] Notes

⚠ **Caution**

- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

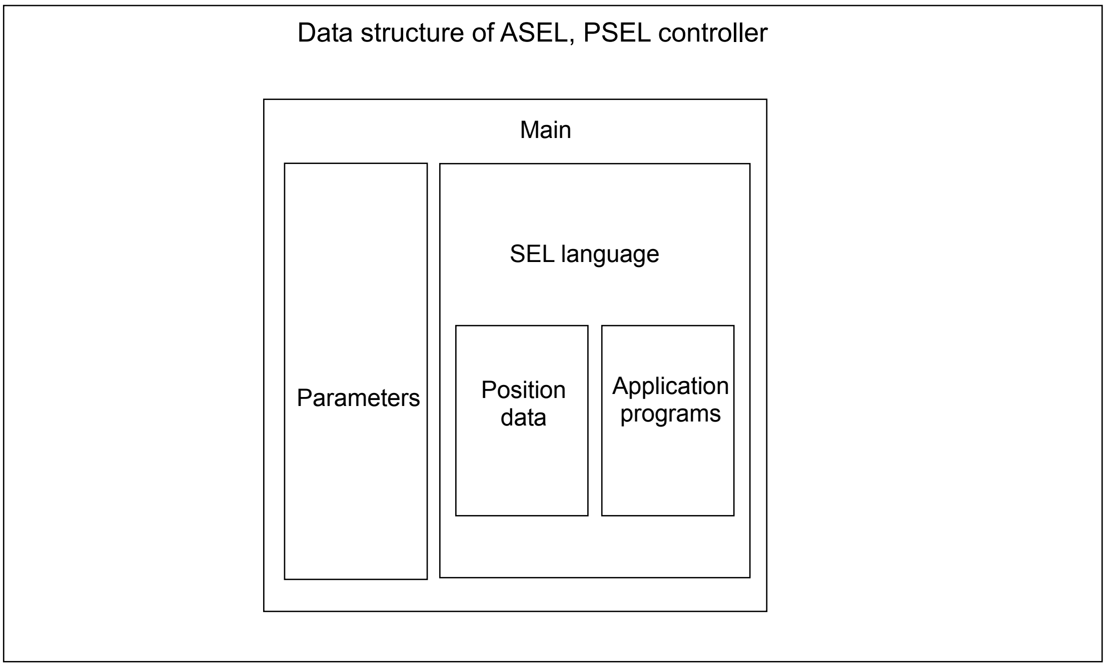
- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

- **Regarding Position Data Save**
  The storage domain for the position data is battery-less backup memory for the position (first half) and flash ROM of the main CPU for the position (second half). All the position data comment is to be stored in the flash ROM of the main CPU.
  Therefore, if the power is turned OFF or the software reset is conducted before writing to the flash ROM, the position (second half) and the position comment data are deleted, and the data that was previously written to the flash ROM is read out the next time the system is turned on. Do not fail to conduct the flash ROM writing when data saving is required.

- **About Initializing of Memory**
  Because the position data, maintenance information data and SEL global data will not be initialized (error data remains) even after an error is detected, make sure not to use the data without canceling it. To cancel an error, initialize the memory of the data which an error has been detected.
  For the position data (second half), do not fail to conduct the flash ROM writing at the same time after initializing.
  (Reference) How to Initialize Memory
  - Position Data: Select [Memory Initialization] → [Position Data] Menu in the PC software
  - Coordinate System Data: Select [Memory Initialization] → [Coordinate System Definition Data] Menu in the PC software
  - User Retaining Memory: Select [Memory Initialization] → [User Retaining Memory] Menu in the PC software
  - SEL Global Data: Select [Memory Initialization] → [Global Variables/Flags] Menu in the PC software
  - Maintenance Information Data: Select [Memory Initialization] → [Maintenance Information] in the PC software and select [Information Initialization]
    * Initialization available when Error No. 4A4, 4A5 or 4A6 has occurred

### 3.7.5 ASEL, PSEL

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.



[2] Saving of data

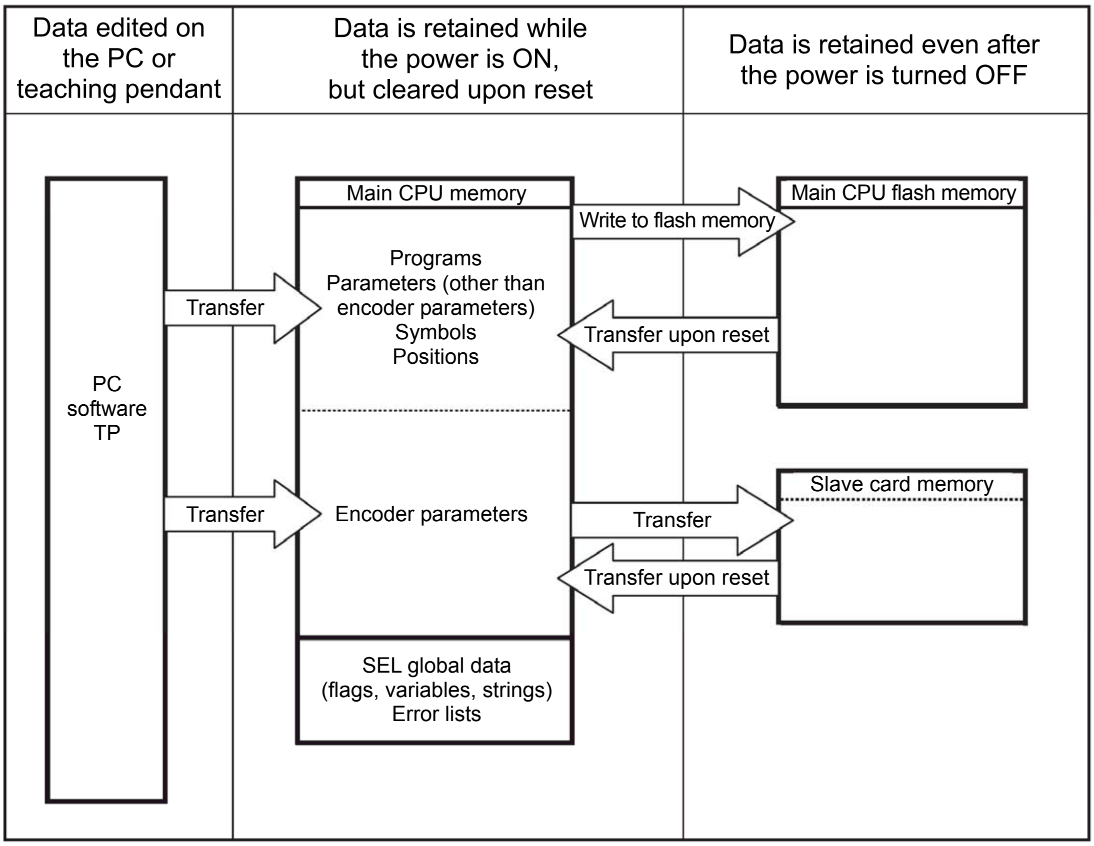
On ASEL and PSEL controllers, data is saved as shown below.

Even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.

To save the data without fail, be sure to write the data you want to save to the flash ROM.

[System-memory backup battery is not used]

Other parameter No. 20 = 0 (System-memory backup battery not installed)

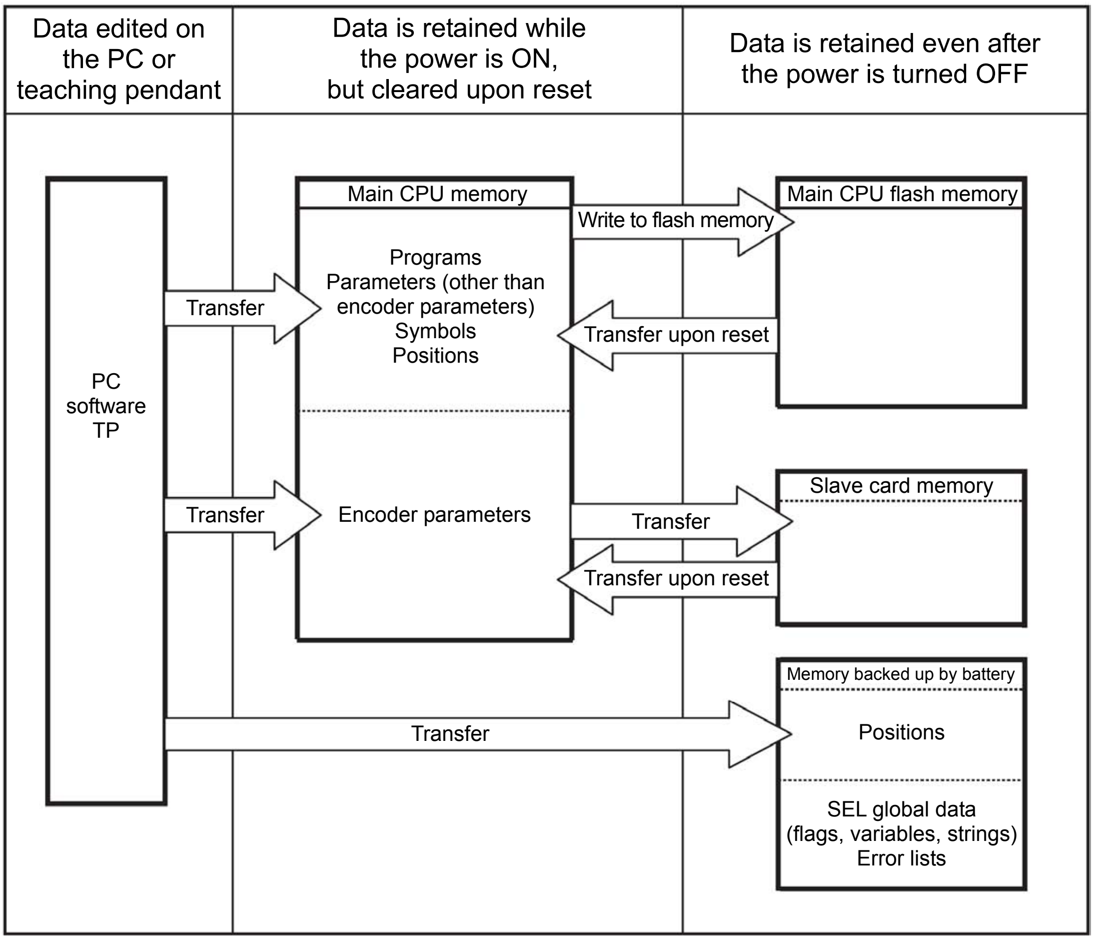



Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in the main CPU memory (excluding parameters).

Note: SEL global data cannot be retained unless the backup battery is installed.
SEL global data is cleared once the control power is turned OFF or software is reset.
Error lists are cleared once the control power is turned OFF.

[System-memory backup battery (optional) is used]

The setting of other parameter No. 20 = 2 (System-memory backup battery installed) must be changed.




Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).

[3] Notes

**⚠ Caution**

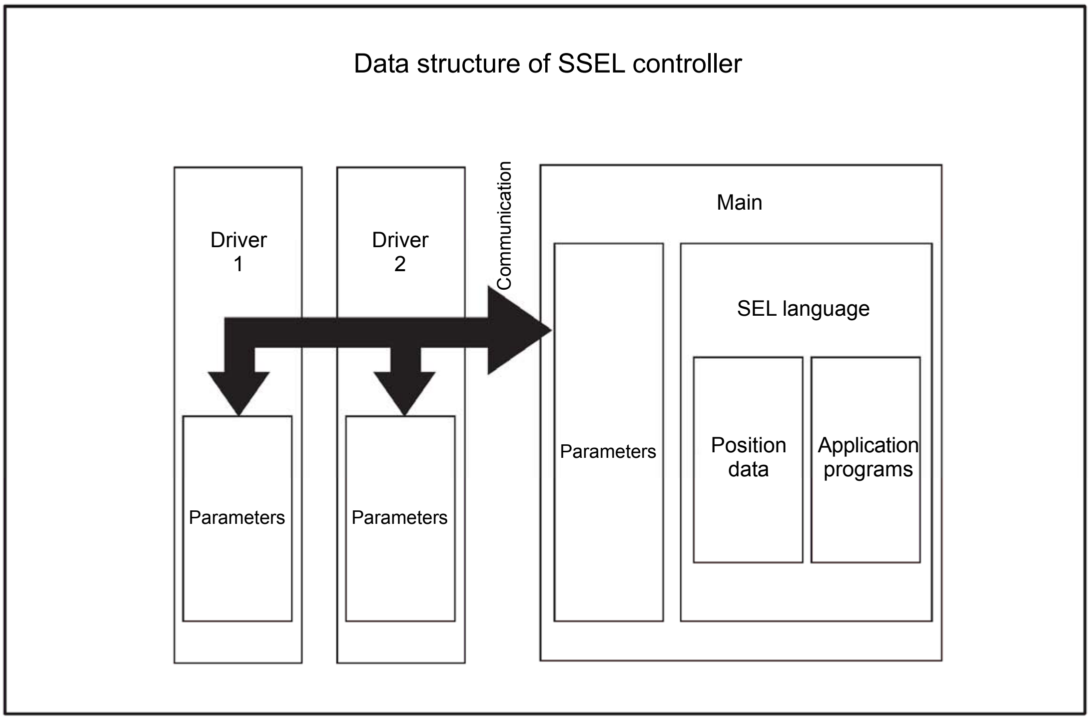
- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

### 3.7.6 SSEL

[1] Data structure
The controller contains parameters as well as position data and application programs used to use the SEL language fully.

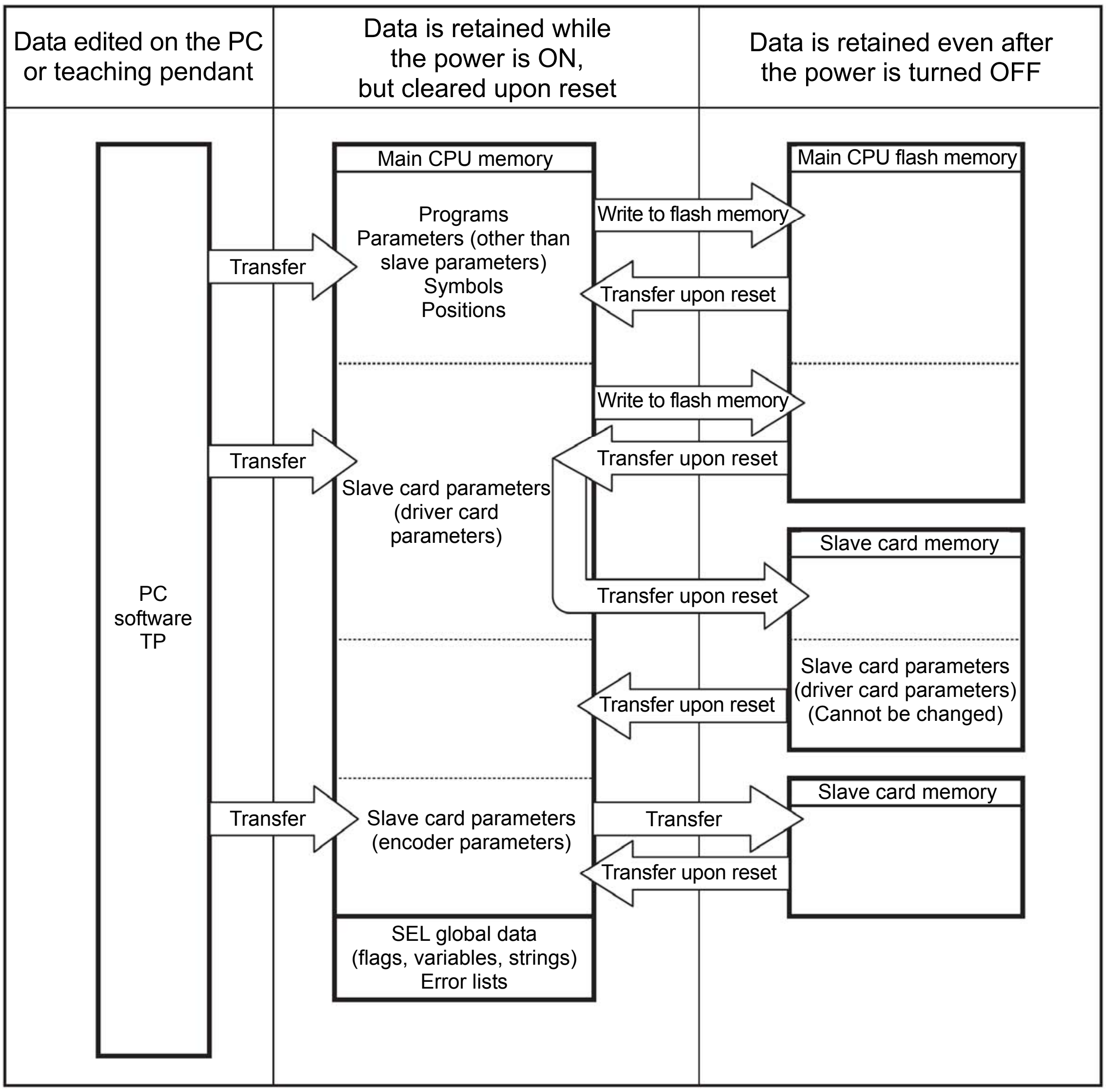



The customer must create position data and application programs.
Certain parameters can be changed according to the customer's system.

[2] Saving of data
On SSEL controllers, data is saved as shown below.
Even if you transfer data to your controller via the PC software or teaching pendant, the data is only written to the temporary memories and will be cleared once the power is turned OFF or controller is reset, as shown below.
To save the data without fail, be sure to write the data you want to save to the flash ROM.

[System-memory backup battery is not used]

Other parameter No. 20 = 0 (System-memory backup battery not installed)

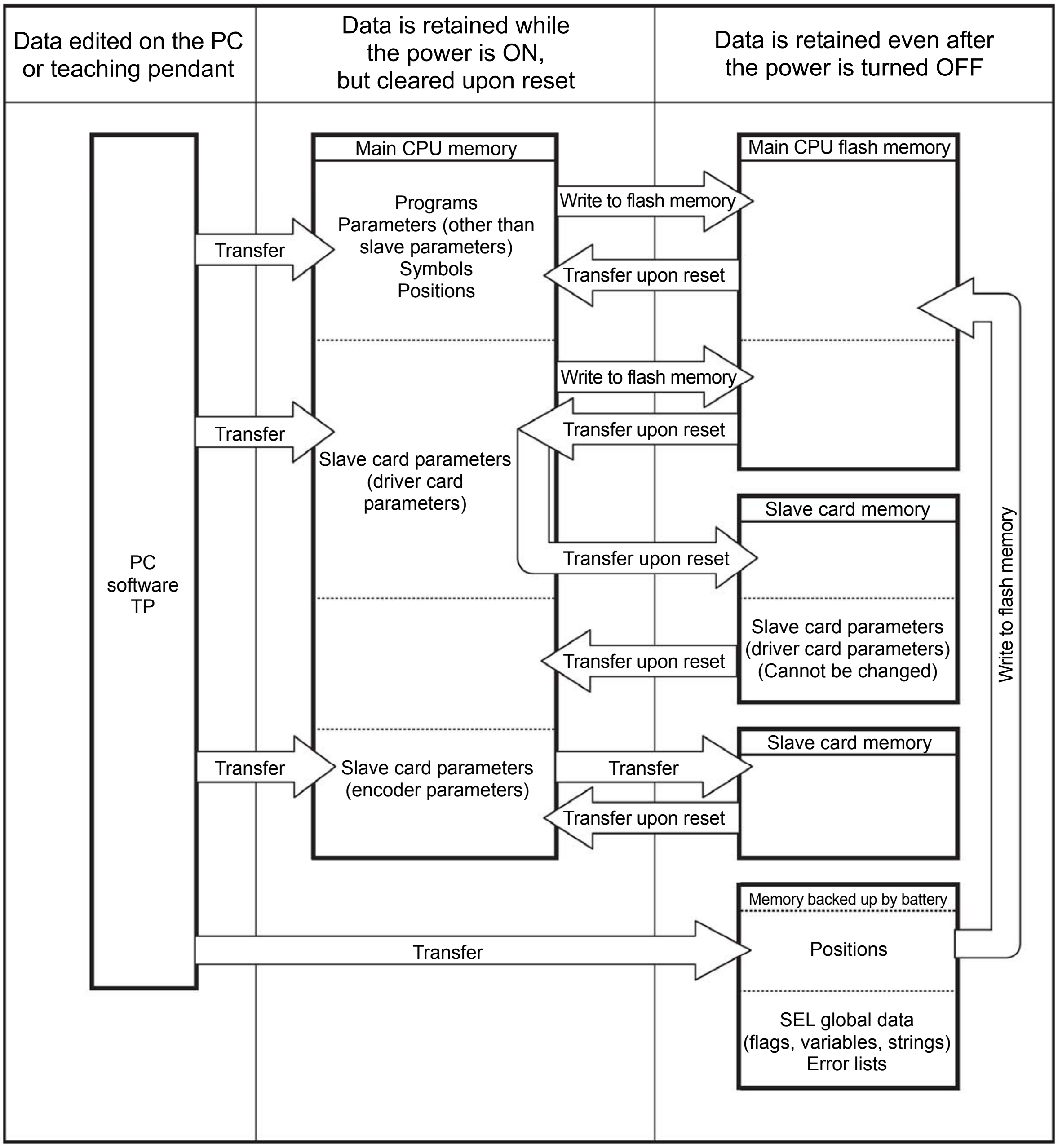



Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in the main CPU memory (excluding parameters).

Note: SEL global data cannot be retained unless the backup battery is installed.
SEL global data is cleared once the control power is turned OFF or software is reset.
Error lists are cleared once the control power is turned OFF.

[System-memory backup battery (optional) is used]

The setting of other parameter No. 20 = 2 (System-memory backup battery installed) must be changed.




Since programs, parameters and symbols are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory.
The controller always operates according to the data in each temporary memory (excluding parameters).

[3] Notes

⚠ **Caution**

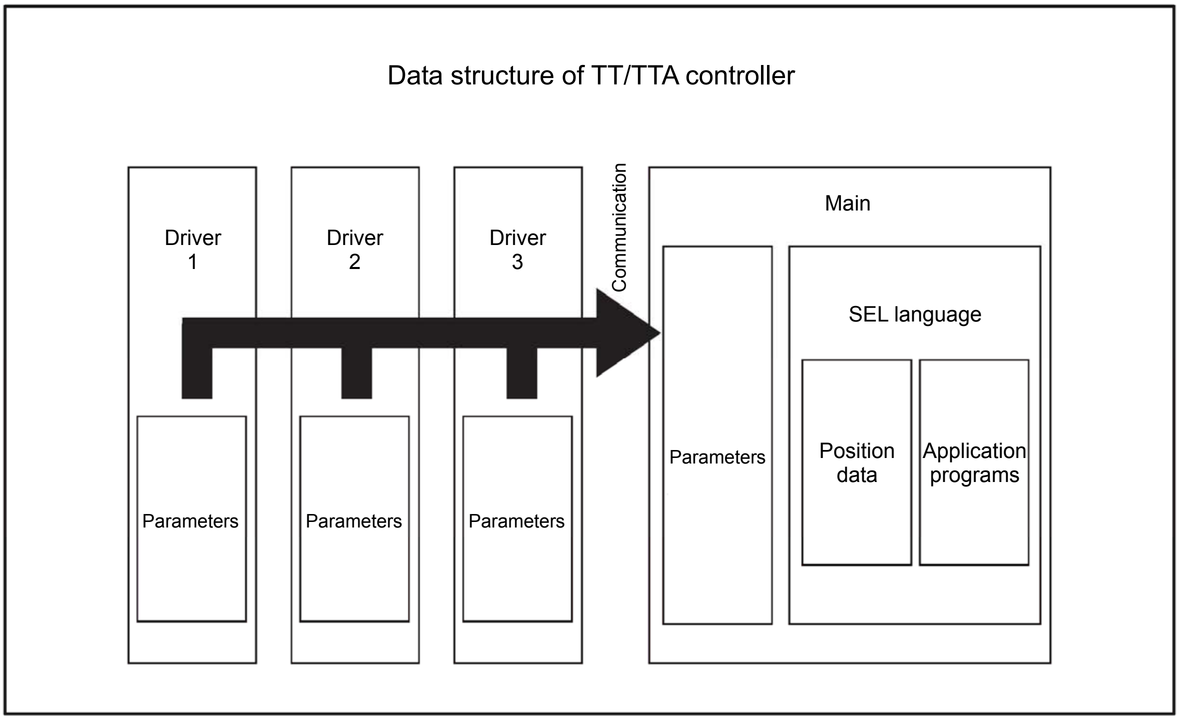
- **Notes on transferring data and writing it to the flash memory**
  Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.

- **Notes on saving parameters to a file**
  Encoder parameters are stored in the EEPROM of the actuator's encoder. (Unlike parameters of other types, these parameters are not stored in the controller's EEPROM.) When the power is turned ON or software is reset, encoder parameters are loaded from the EEPROM to the controller.
  Accordingly, if parameters are saved to a file after the controller power was turned on (or software was reset) while the actuator (encoder) was still not connected, the encoder parameters in this file will become invalid.

- **Notes on transferring a parameter file to the controller**
  When a parameter file is transferred to the controller, encoder parameters are transferred to the encoder's EEPROM (excluding manufacturing information and function information).
  Accordingly, if a parameter file is read and transferred to the controller after the controller power was turned on while the actuator was still not connected, invalid encoder parameters will be written to the encoder's EEPROM (as they are transferred to the controller to which the actuator is connected).
  To save parameters to a file, do so while the actuator is connected.

### 3.7.7 TT/TTA

[1] Data structure
The controller module of a tabletop robot contains parameters as well as position data and application programs used to drive the SEL language.

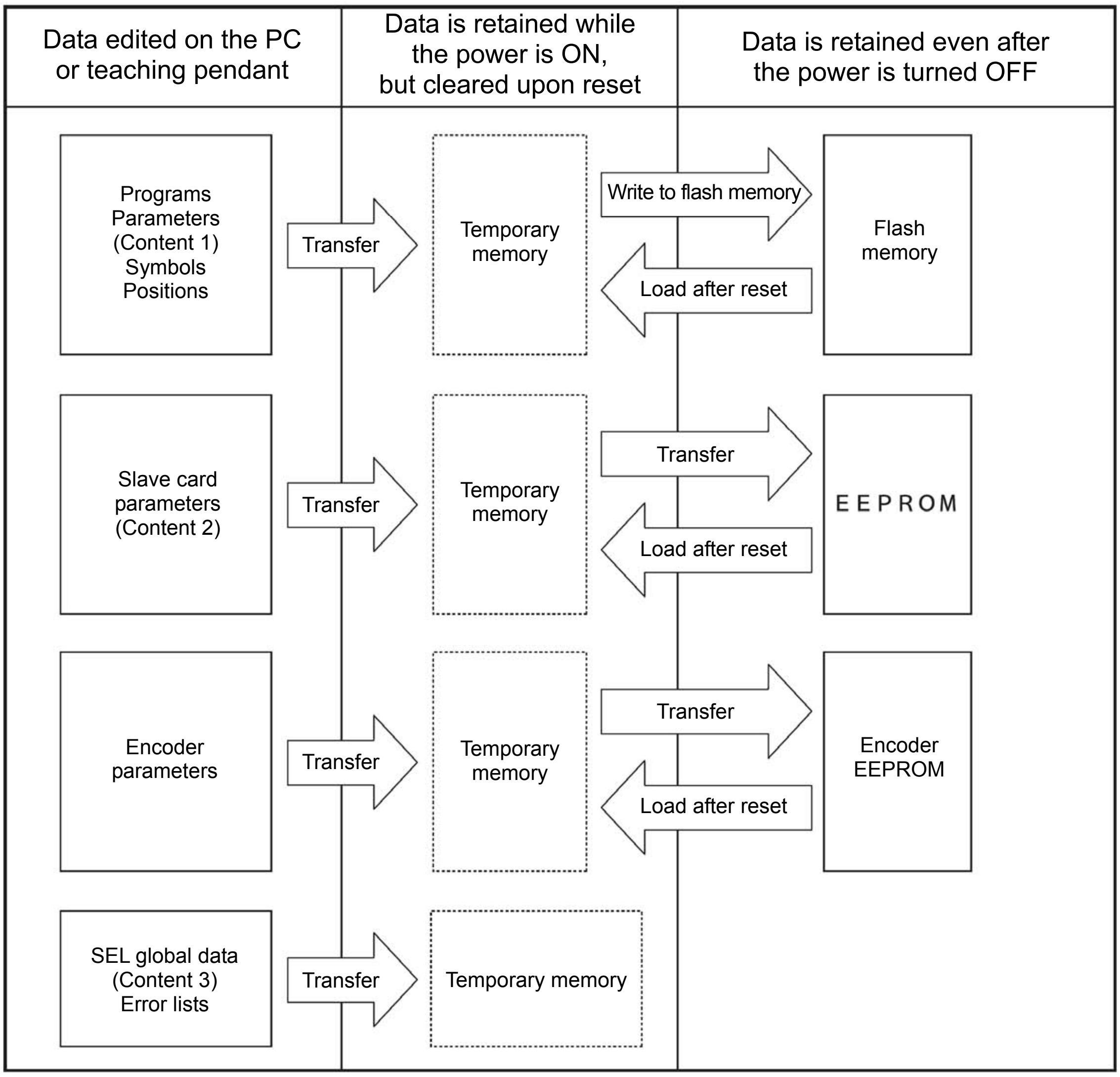



The customer must create position data and application programs.
Certain parameters can be changed according to the customer's system.
[Refer to tabletop robot TT Instruction Manual provided separately.]

[2] Data Saving of TT

When data created/edited using the PC software or teaching pendant is transferred to the controller (by pressing the Write button or WRT key if you are using the teaching pendant), the data is temporarily stored in the controller's memories. Accordingly, such data will be cleared once the power is turned off or software is reset (restarted).
If you want your data to be retained, be sure to write it to the flash memory.

Note: Global data (variables, flags, strings) is cleared once the power is turned OFF or software is reset (restarted) (global data cannot be retained after the power is turned OFF). Error lists are retained after the software is reset, but cleared if the power is turned OFF.




Content 1: Parameters other than those included in Content 2 below and encoder parameters
Content 2: Driver card and I/O slot card (power card) parameters
Content 3: Flags, variables and strings

Since programs, parameters, symbols and positions are loaded from the flash memory upon restart, these data in the temporary memories will return to the conditions before editing unless written to the flash memory. The controller always operates according to the data in each temporary memory (dotted box) (excluding parameters).

[3] Notes

⚠ **Caution**

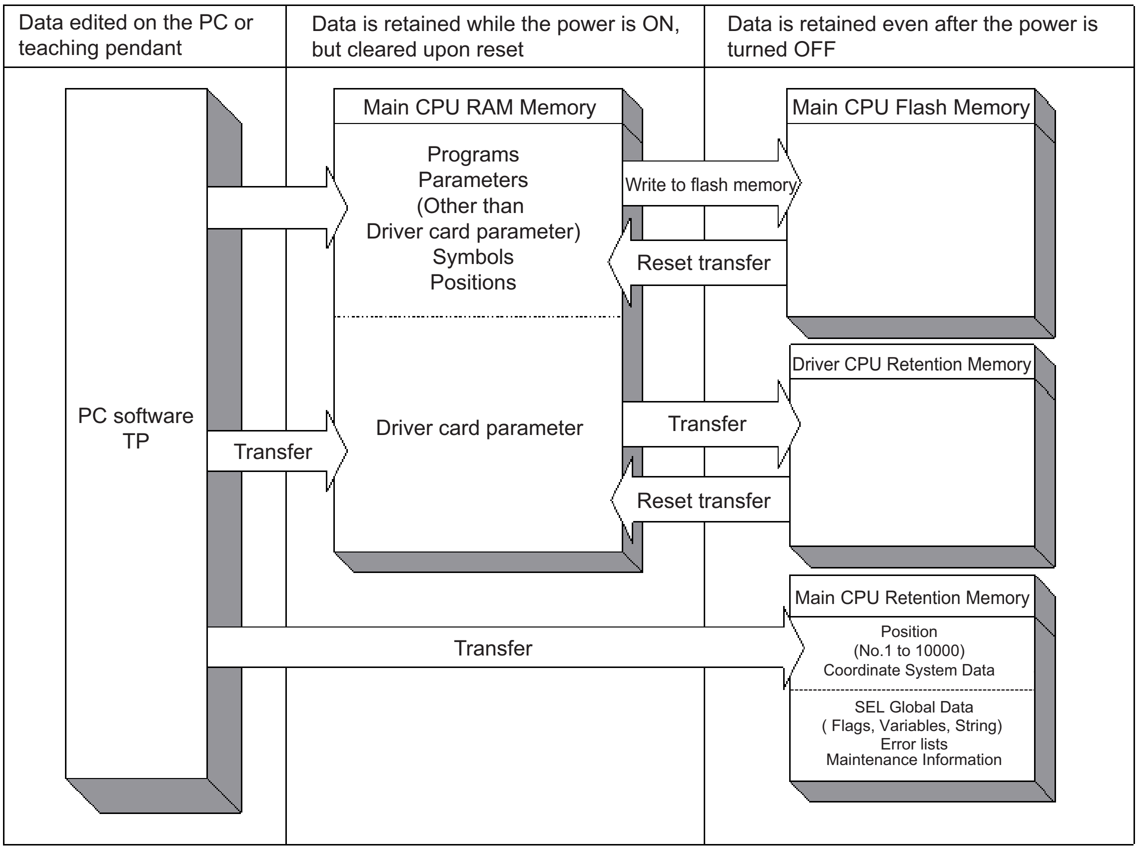
**Notes on transferring data and writing it to the flash memory**

Never turn OFF the main power while data is being transferred or written to the flash memory, because data may be lost and the controller will no longer be able to operate.



[4] Data Saving of TTA

In the retaining memory (FRAM), Position data (No. 1 to 10000), SEL global data, error list and maintenance information are stored for backup in standard with no battery.
Position (No. 1 to 10000) is stored only in the retaining memory. (It is not necessary to have the flash ROM writing.)




(Note 1) Do not attempt to turn the power OFF during the memory initialization (position, global variables and flags) or the maintenance information initialization. As the initializing process terminates incomplete, errors described below* may be generated in the next startup. Redo initializing in case the power is turned OFF accidently, and an error is generated. (* Error No. 4A4, 69E, 6C7 or 826)

(Note 2) As the position data, maintenance information data and SEL global data will not be initialized even if an error gets detected (error data can be seen as it is), do not attempt to use the data as it is. To cancel the error, initialize the memory in the data the error was detected.
For position data (No. 10001 to 30000), conduct also the flash ROM writing after initializing.

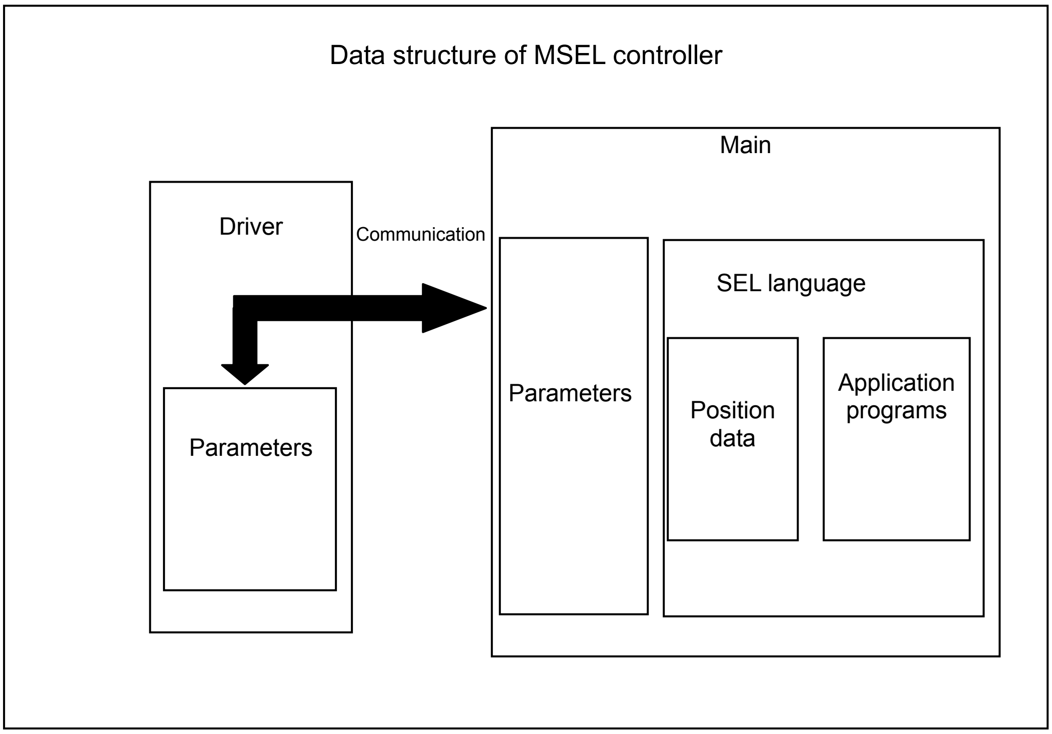
(Reference) How to Initialize Memory

- Position Data: Select [Memory Initialization] → [Position Data] Menu in the PC software
- Coordinate System Data: Select [Memory Initialization] → [Coordinate System Definition Data] Menu in the PC software
- SEL Global Data: Select [Memory Initialization] → [Global Variables/Flags] Menu in the PC software
- Maintenance Information Data: Select [Memory Initialization] → [Maintenance Information] in the PC software and select [Information Initialization]
  * Initialization available when Error No. 4A4, 4A5 or 4A6 has occurred

### 3.7.8 MSEL

[1] Data structure

The controller module of a MSEL contains parameters as well as position data and application programs used to drive the SEL language.

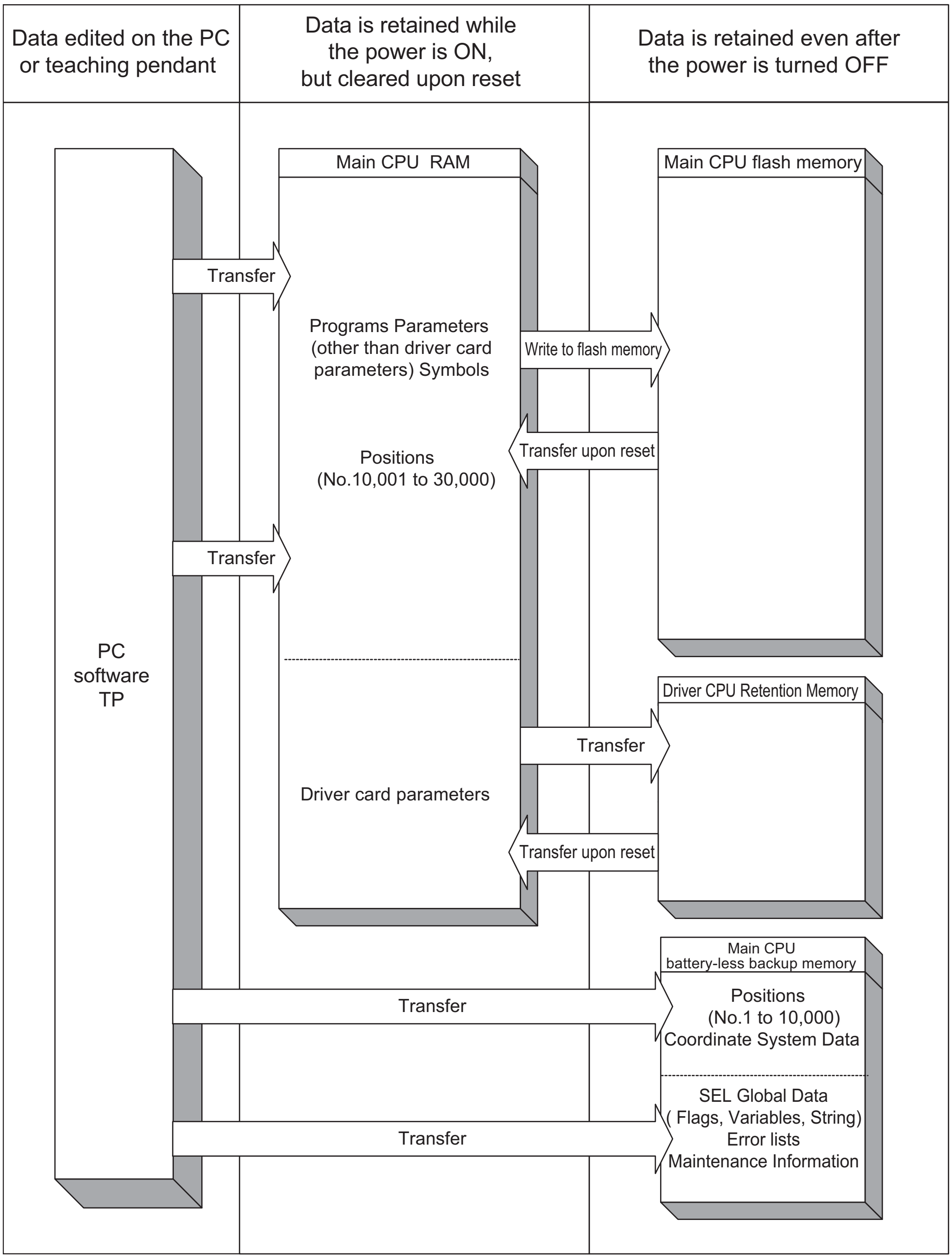



The customer must create position data and application programs.
Certain parameters can be changed according to the customer's system.

[2] Saving of data

On MSEL controllers, data is saved as shown below.
Even if you transfer data via the PC software or teaching pendant, the data, except for some (Note), is only written to the memories temporarily and will be cleared once the power is turned off or controller is reset.
To save the data without fail, be sure to write the data you want to save to the flash ROM.

(Note) The position data (No. 1 to 10000), SEL global data, error list, maintenance information and SCARA coordinate system data are stored in the battery-less backup memory (FRAM). There is no need of flash ROM writing.



(Note 1) Do not attempt to turn the power off while initializing the memories (position, global variables and flags) or maintenance information. It may cause to generate such as an error* in the next startup due to incomplete of initializing process. Have an initializing process again in case the power is turned off accidently. (* Error No. 4A4, 69E, 6C7, 826)

(Note 2) Because the position data, maintenance information data and SEL global data will not be initialized (error data remains) even after an error is detected, make sure not to use the data without canceling it. To cancel an error, initialize the memory of the data which an error has been detected.
For the position data (No. 10001 to 30000), do not fail to conduct the flash ROM writing at the same time after initializing.

(Reference) How to Initialize Memory

- Position Data: Select [Memory Initialization] → [Position Data] Menu in the PC software
- Coordinate System Data: Select [Memory Initialization] → [Coordinate System Definition Data] Menu in the PC software
- SEL Global Data: Select [Memory Initialization] → [Global Variables/Flags] Menu in the PC software
- Maintenance Information Data: Select [Memory Initialization] → [Maintenance Information] in the PC software and select [Information Initialization]
  * Initialization available when Error No. 4A4, 4A5 or 4A6 has occurred

# 4. Program Edit

## 4.1 Each Type of Data Available to Handle on the Program and its Range

In SEL language, separate areas are provided for each task such as I/O port, variables, flags, etc. Some areas are separated to the global area and local area. Data set to the global area can be read and written from multiple programs.
The global domain is backed up in the controller battery for the models except for XSEL-R*/S*, TTA, MSEL Data in local area gets cleared each time the program is booted.
In the following, explains about the area and range. However, It will not be backed up. SSEL/ASEL/PSEL will be possible backup option.

| Function | Global area | | Local area | | Remarks |
|---|---|---|---|---|---|
| | Range | Total number | Range | Total number | |
| Input port | 000 to 299 | 300 | | | |
| Output port | 300 to 599 | 300 | | | |
| Extended Input Ports | 1000 to 3999 | 3000 | | | Applied for XSEL-P/Q/PCT/QCT and XSEL-R/S/RX/SX/RXD/SXD, XSEL-RA/SA/RAX/SAX/ RAXD/SAXD |
| Extended Output Ports | 4000 to 6999 | 3000 | | | |
| Flag | 600 to 899 | 300 | 900 to 999 | 100 | |
| Variable (integer) | 200 to 299 | 100 | 1 to 99 | 99 | 99 is a special variable used in IN, INB, OUT and OUTB Variable (integer) commands, etc. |
| | 1200 to 1299 | 100 | 1001 to 1099 | 99 | |
| | 2000 to 2799 | 800 | | | XSEL-P/Q/PCT/QCT (Main application Ver.1.47 or later) Applied for PC Software V12.02.00.00 or later, Touch Panel Teaching (TB-01) V1.30 or later, Touch Panel Teaching (TB-02) XSEL-R/S (Main application Ver.1.24 or later) Applied for PC Software V12.03.04.00 or later, Touch Panel Teaching (TB-01) V1.51 or later, Touch Panel Teaching (TB-02) Applied for XSEL-RA/SA/RAX/SAX/ RAXD/SAXD |
| Variable (real number) | 300 to 399 | 100 | 100 to 199 | 100 | 199 is a special variable used in PPUT, PGET and PAPG commands, etc. |
| | 1300 to 1399 | 100 | 1100 to 1199 | 100 | |
| String | 300 to 999 | 700 | 1 to 299 | 299 | |
| Tag number | | | 1 to 256 | 256 | |
| Sub routine number | | | 1 to 99 | 99 | |
| Work coordinate system number | 0 to 31 | 32 | | | For SCARA robots, For the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56) |

| Function | Global area | | Local area | | Remarks |
|---|---|---|---|---|---|
| | Range | Total number | Range | Total number | |
| Tool coordinate system number | 0 to 127 | 128 | | | For SCARA robots, For the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56) |
| Simple contact check zone number | 1 to 10 | 10 | | | For SCARA robots |
| Zone number | 1 to 4 | 4 | | | For single-axis/ Cartesian robots |
| Palletizing number | | | 1 to 10 | 10 | Other than XSEL-RA/SA/RAX/SAX/ RAXD/SAXD |
| | | | 1 to 32 | 10 | XSEL-RA/SA/RAX/SAX/ RAXD/SAXD |
| Axis number | 1 to 8 | 8 | | | Varies depending on the controller. |
| Axis pattern | 0 to 11111111 | | | | Varies depending on the controller. |
| Program number (XSEL-P/Q/PX/QX/PCT/QCT, XSEL-R/S/RX/SX/RXD/SXD, SSEL) | 1 to 128 | 128 | | | |
| Program number (XSEL-J/K/KE/KTKET/JX/KX/KETX, TT, ASEL/PSEL) | 1 to 64 | 64 | | | |
| Program number (XSEL-RA/SA/RAX/SAX/RAXD/SAXD, TTA, MSEL) | 1 to 255 | 255 | | | |

| Function | | Global area Range | Global area Total number | Local area Range | Local area Total number | Remarks |
|---|---|---|---|---|---|---|
| Position number | XSEL-R/S/RX/SX/RXD/SXD | 1 to 53332 (MAX) | 53332 (MAX) | | | Depend on how many axes are to be used |
| | XSEL-RA/SA/RAX/SAX/RAXD/SAXD | 1 to 55000 (MAX) | 55000 (MAX) | | | Depend on how many axes are to be used |
| | XSEL-P/Q/PX/QX/PCT/QCT, SSEL | 1 to 20000 | 20000 | | | |
| | XSEL-J/K/KE/KT/KET/JX/KX/KETX, TT | 1 to 3000 | 3000 | | | |
| | ASEL/PSEL | 1 to 1500 | 1500 | | | |
| | TTA, MSEL | 1 to 30000 | 30000 | | | |
| Position comments (Half-sized 32 characters) | XSEL-R/S/RX/SX/RXD/SXD XSEL-RA/SA/RAX/SAX/RAXD/SAXD | | | 1 to 10000 | 10000 | Comment can be added only in Positions No. 1 to 10000 |
| Task level | | 0: NORMAL/ 1: HIGH | 2 | | | |
| SIO channel number | XSEL-P/Q/PCT/QCT/PX/QX/R/S/RX/SX/RXD/SXD | 1 to 2 | 2 | | | |
| | XSEL-RA/SA/RAX/SAX/RAXD/SAXD | 1 | 1 | | | |
| | XSEL-J/JK (1, 2-axis specification) TT | 1 | 1 | | | To be communized with teaching and PC software |
| | SSEL/ASEL/PSEL | 0 | 1 | | | |
| | XSEL-J/JK (3, 4-axis specification) | 1 to 3 | 3 | | | To be communized with teaching and PC software of channel No. 1 Channels after Channel No. 2 are available for use when extension SIO device is mounted |
| | XSEL-K/KE/KT/KET/KX/KETX | 1 to 7 | 7 | | | |
| | TTA | 1 to 3 | 3 | | | |
| | MSEL | 1 to 2 | 2 | | | |
| WAIT timer | | | | 1 | | TIMW command |
| 1-shot pulse timer | | | | 16 (Can be operated simultaneously.) | | BTPN, BTPF command |
| Ladder timer | | | | Use local area flags. 900 to 999 | 100 | TIMR command |
| Virtual input port (SEL system → SEL user program) | | 7000 to 7299 | 300 | | | Applied for other than TTA, MSEL, XSEL-RA/SA/RAX/SAX/RAXD/SAXD |
| Virtual output port (SEL user program → SEL system) | | 7300 to 7599 | 300 | | | Applied for other than TTA, MSEL, XSEL-RA/SA/RAX/SAX/RAXD/SAXD |
| Virtual output port | | 7000 to 7599 | 600 | | | Applied for TTA, MSEL, XSEL-RA/SA/RAX/SAX/RAXD/SAXD |
| Number of symbol definitions | XSEL-P/Q/PCT/QCT/PX/QX/R/S/RX/SX/RXD/SXD | 1000 | | | | |
| | XSEL-RA/SA/RAX/SAX/RAXD/SAXD | 1000 | | | | |
| | XSEL-J/K/KE/KT/KET/JX/KX/KETX, TT, TTA, MSEL, SSEL | 1000 | | | | |
| | ASEL/PSEL | 500 | | | | |

| Function | | Global area | | Local area | | Remarks |
|---|---|---|---|---|---|---|
| | | Range | Total number | Range | Total number | |
| Number of symbol used in commands | XSEL-P/Q/PCT/QCT/PX/QX/R/S/RX/SX/RXD/SXD | 5000 (including string literals) | | | | |
| | XSEL-RA/SA/RAX/SAX/RAXD/SAXD | 10000 (including string literals) | | | | |
| | XSEL-J/K/KE/KT/KET/JX/KX/KETX, TT, TTA, MSEL, SSEL | 5000 (including string literals) | | | | |
| | ASEL/PSEL | 2500 (including string literals) | | | | |
| Number of recorded history | XSEL-R/S/RX/SX/RXD/SXD, TTA, MSEL | 400 | | | | |
| | XSEL-RA/SA/RAX/SAX/RAXD/SAXD | 400 | | | | |
| | XSEL-P/Q/PCT/QCT/PX/QX | 200 | | | | |
| | XSEL-J/K/KE/KT/KET/JX/KX/KETX, TT | 200 | | | | |
| | SSEL/ASEL/PSEL | 100 | | | | |

* Character-string literals are used in certain string-operation commands and consist of the portion enclosed by single quotation marks (‘ ‘) (maximum eight single-byte characters).

(Note) When the number of symbols used in a command is at nearly the limit, C46 error “Blank area shortage error with source-symbol storage table” will occur.

## 4.2 Setting of Function and Values

Explanation below shows how you should handle the I/O port and how you should take the variables in your mind when you create a program with SEL language.

### 4.2.1 Handling of I/O Port

Refer to “2.1 I/O Signal” for I/O ports.

[1] Input ports
These ports are used as input ports for limit switches, sensor switches, etc.

| Input number assignment |
|---|
| 000 to 031 (standard) |

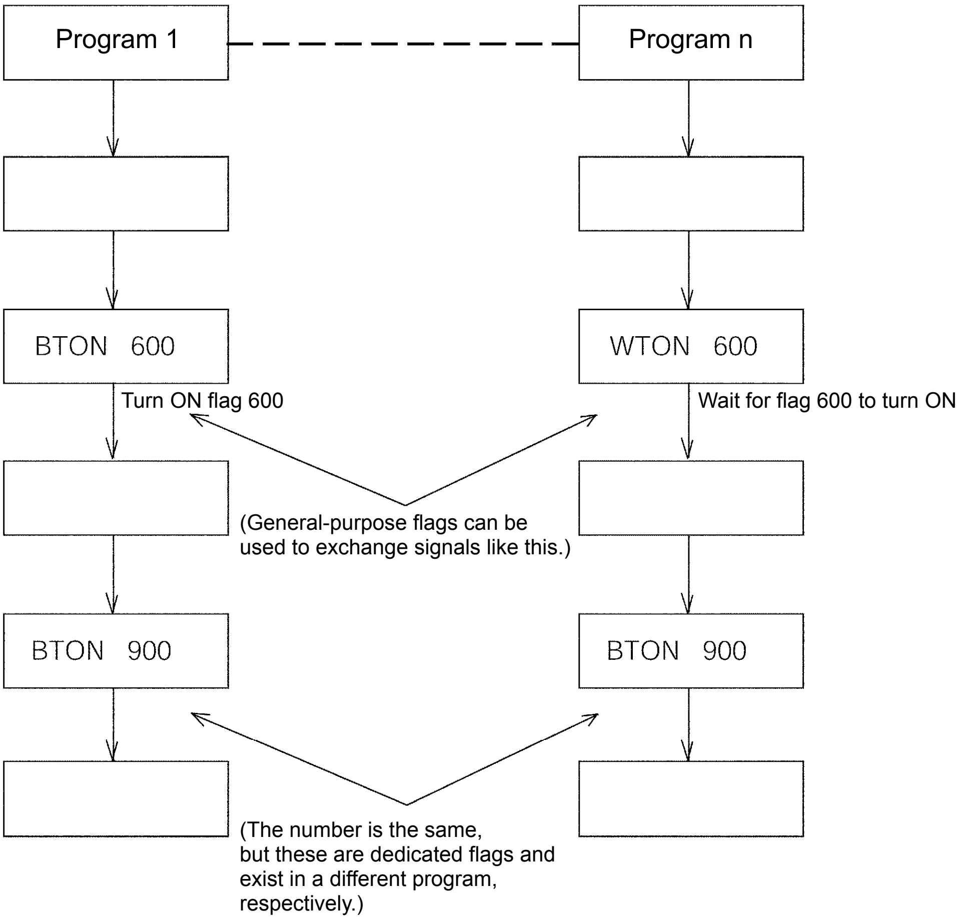
[2] Output ports
These ports are used as various output ports.

| Output number assignment |
|---|
| 300 to 315 (standard) |

### 4.2.2 Handling (Setting and Resetting) of Flags

Unlike their literal meaning, flags are actually “memories” where data is set and reset. Flags correspond to “auxiliary relays” in sequencers.
Flags are classified into two types: general-purpose flags (global flags) that are assigned numbers from 600 to 899 and usable in all programs, and dedicated flags (local flags) that are assigned numbers from 900 to 999 and usable only in each program.
The general-purposed flags (global flags) can be saved (in the battery backup or saving memory, depends on the models) even after the power is turned OFF.
Dedicated flags (local flags) will be cleared once the power is turned OFF.

| | | | |
|---|---|---|---|
| Flag No. | 600 to 899 | Usable in all programs. | "General-purpose flags (global flags)" |
| Flag No. | 900 to 999 | Usable only in each program. | "Dedicated flags (local flags)" |



### 4.2.3 How to Deal with Values and Variables

(1) How to Deal with Values

If the last digit of the set value is H, set with hexadecimal number.
Refer to the following.
Input the value of hexadecimal number transformed from the binary number.

●Binary number

Binary number expresses a numeral figure with using 2 numbers, 0 and 1.
The number increases in the order of 0, 1, and then the number of digit increases, and goes 10, 11 …

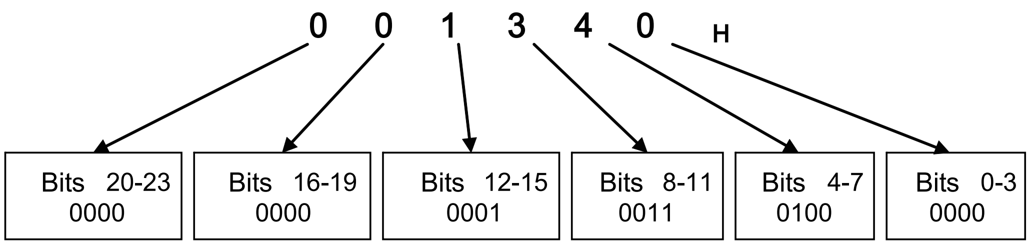
| Decimal number | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Binary number | 0 | 1 | 10 | 11 | 100 | 101 | 110 | 111 | 1000 | 1001 | 1010 |

●Hexadecimal number

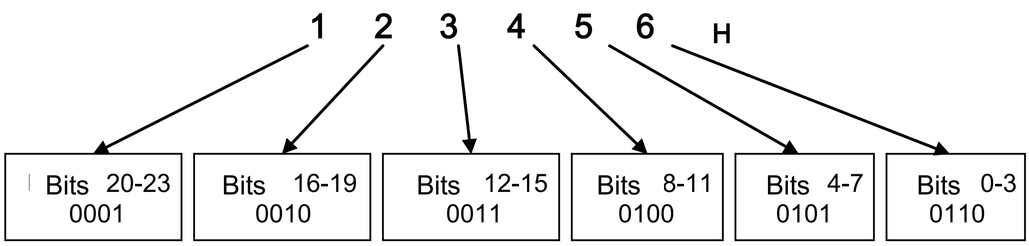
Hexadecimal number expresses a numeral figure with using numbers from 0 to 9 and alphabets from A to F. The number increases in the order of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B,C, D, E, F, and then the number of digit increases, and goes 10, 11, …

| Decimal number | 0 to 9 (Same for decimal and hexadecimal numbers) | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|
| Hexadecimal number | | A | B | C | D | E | F | 10 |

Example 1 : $001340_H$

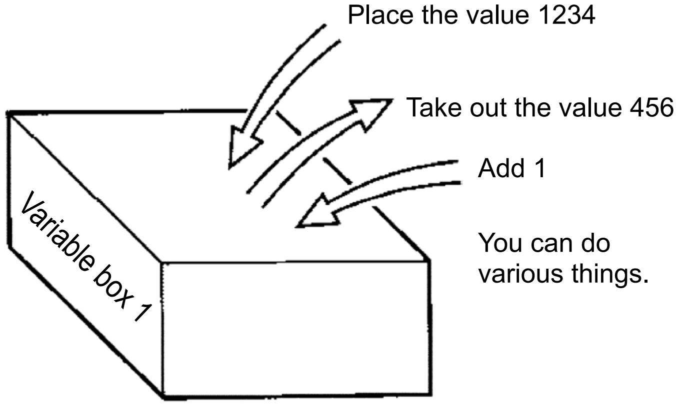



Example 2 : $123456_H$



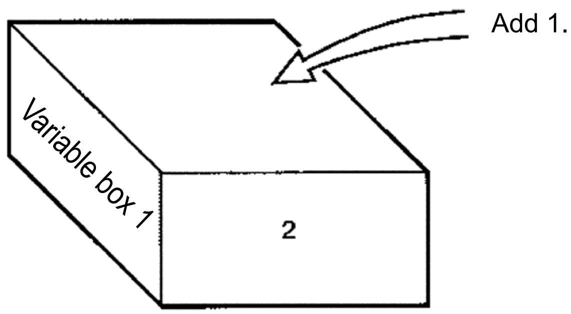
(2) Types and Handling of Variables

1) Meaning of variables
"Variable" is a technical software term. Simply put, a variable is a "container in which a value is placed". You can use variables in many different ways such as placing a value in a variable, taking a value out of a variable, and adding or subtracting a value to/from a variable, to name a few.




| Command | Operand 1 | Operand 2 |
|---|---|---|
| ADD | 1 | 1 |

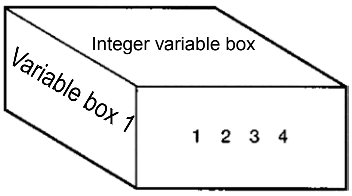
With this command, if 2 is already placed in the box of variable 1 as shown, then 1 is added and the content of variable 1 becomes 3.




(2 is already placed inside.)

2) Types of variables
Variables are classified into two types as explained below.

[Integer variables]
These variables cannot handle decimal points.
[Example] 1234

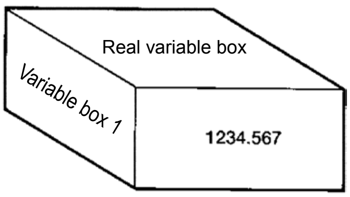



| | | | |
|---|---|---|---|
| Integer variable No. | 200 to 299<br>1200 to 1299 | Usable in all programs. | "Global integer variables" |
| Integer variable No. | 2000 to 2799 | Usable in all programs. "Global integer variables"<br>(XSEL-P/Q/PCT/QCT (Ver.1.47 or later)<br>Applied for PC Software V12.02.00.00 or later, Touch Panel Teaching (TB-01) V1.30 or later, Touch Panel Teaching (TB-02)<br>XSEL-R/S (Main application Ver.1.24 or later)<br>Applied for PC Software V12.03.04.00 or later, Touch Panel Teaching (TB-01) V1.51 or later, Touch Panel Teaching (TB-02)<br>XSEL-RA/SA/RAX/SAX/RAXD/SAXD) | |
| Integer variable No. | 1 to 99<br>1001 to 1099 | Usable only in each program. | "Local integer variables" |

**⚠ Caution**

- Values from -9,999,999 to 99,999,999 can be entered in programs.
- Variable 99 is a special register used for integer calculations by the system.

[Real variables]
These variables are actual numbers and can also handle decimal points.
[Example] 1234.567
↑
(decimal point)



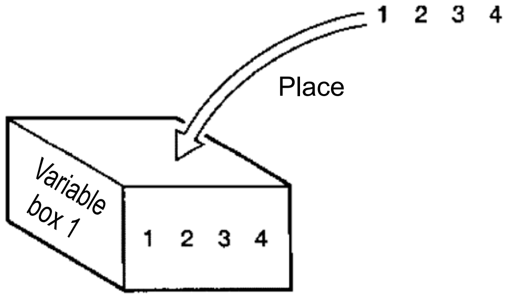

| | | | |
|---|---|---|---|
| Real variable No. | 300 to 399<br>1300 to 1399 | Usable in all programs. | "Global real variables" |
| Real variable No. | 100 to 199<br>1100 to 1199 | Usable only in each program. | "Local real variables" |

**⚠ Caution**

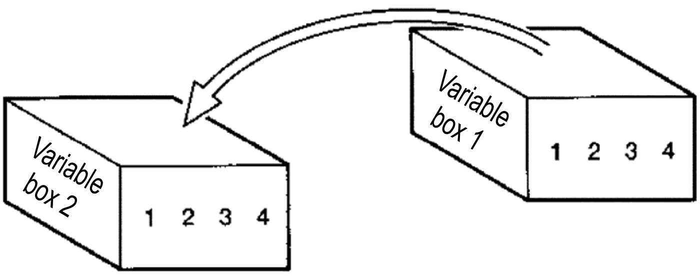
- Values from -99,999.9 to 999,999.9 (up to eight digits including the sign and decimal point) can be entered in programs.
- Variable 199 is a special register used for integer calculations by the system.

[Indirect specification of variables]
Variables are specified with a "*" (asterisk) appended to them.
In the example below, the content of variable box 1 is placed in variable box 2. If "1234" is in variable box 1, "1234" is placed in variable box 2.

| Command | Operand 1 | Operand 2 |
|---|---|---|
| LET | 1 | 1234 |




| Command | Operand 1 | Operand 2 |
|---|---|---|
| LET | 2 | *1 |




This usage is called "indirect specification".

"*" is to be applied also when making an indirect specification of symbolized variables.

| Command | Operand 1 | Operand 2 |
|---|---|---|
| LET | ABC | 1 |
| LET | BCD | 2 |
| ADD | ABC | *BCD |

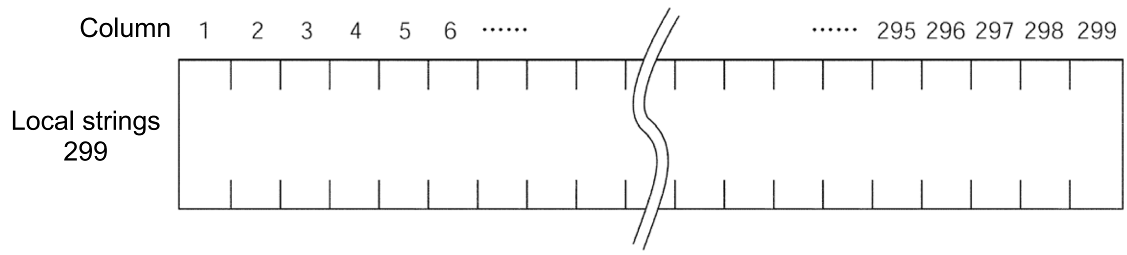
Place 1 in variable ABC.
Place 2 in variable BCD.
Add the content of variable BCD, or 2, to variable ABC.
(The content of variable ABC becomes 3.)

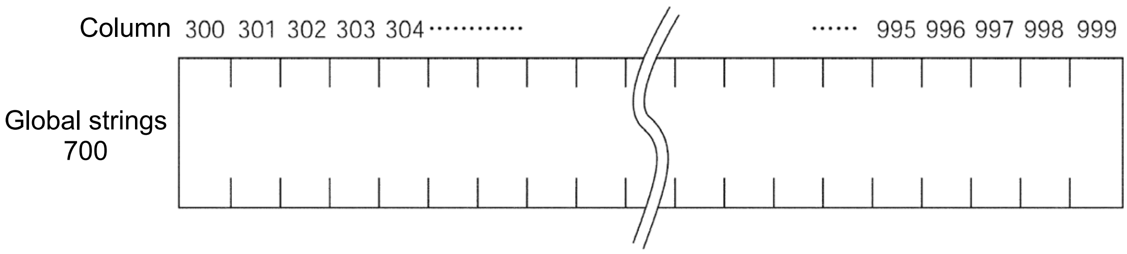
3) Character-String Literals
Character-string literals are used in certain string-operation commands and consist of the portion enclosed by single quotation marks (' ') (maximum eight single-byte characters).
With the PC software, single-byte ASCII code characters from 20h to 7Eh (limited to those that can be input via keyboard) can be used inside the single quotation marks. With the teaching pendant, single-byte alphanumeric characters and single-byte underscores can be used.

## 4.2.4 Specification Method for Local String and Global String

Serial communication is implemented basically by means of exchange of character strings. These character strings are called “string”.
Strings sent in the communication transmission format can be used freely in programs, or specifically they are stored in boxes (columns) in which strings are placed.
These string are classified into global string that can be read or written in all programs, and local string that can be read or written only in each program.
String are differentiated by the range of their number.

| | Column number |
|---|---|
| Global string | 300 to 999 (700) |
| Local string | 1 to 299 (299) |

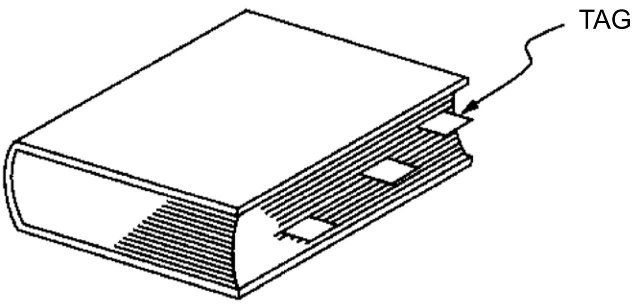



The characters constituting a string are stored one by one in each of these fields.
The position of a given field in a string is expressed by column X, and the column to store each character in can be set freely using a command.

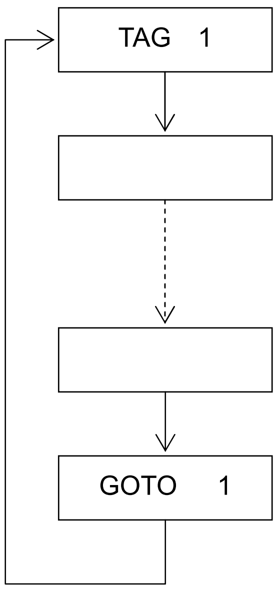
### 4.2.5 Handling of Tag Numbers

A “TAG” is a “heading”.
You may stick labels on pages you want to read frequently. Tags are used for the same purpose.
The destination to jump to where you specify in the jump command “GOTO” is a “TAG”.




| Command | Operand 1 |
|---|---|
| TAG | Tag number (integer of 1 to 256) |

Usable only in each program.



# 5. SEL Commands

## 5.1 How to Read Explanation of Command

How a command is explained is described using an example of LET command.

● LET (Assign)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | LET | Variable number | Data | ZR |

[1] SEL language structure

| Applicable models | | | | | | | |
|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -R/S | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/SXD | ASEL PSEL SSEL | TT |
| ○ | ○ | ○ | ○ | ○ | ○ | ○ | ○ |

[2] Applicable models

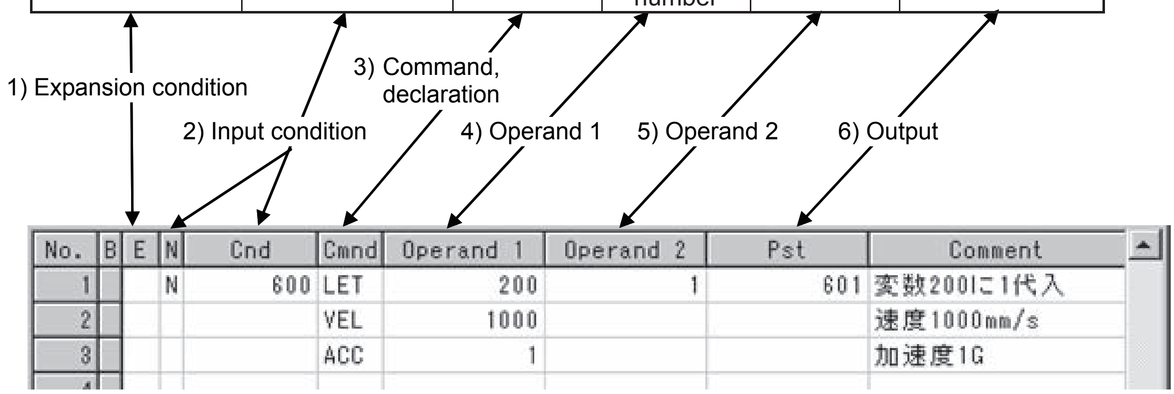
[Function] Assign the value specified in operand 2 to the variable specified in operand 1. The output will turn ON when 0 is assigned to the variable specified in operand 1.

[3] Description of Functions

| | | | | |
|---|---|---|---|---|
| [Example 1] | LET | 1 | 10 | Assign 10 to variable 1. |
| [Example 2] | LET | 1 | 2 | Assign 2 to variable 1. |
| | | 3 | 10 | Assign 10 to variable 3. |
| | LET | *1 | *3 | Assign the content 10 of variable 3 to the variable that corresponds to the content 2 of variable 1. |

### [1] SEL language structure

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | LET | Variable number | Data | ZR |




Program screen

The details of SEL language structure components are explained.

1) Expansion condition (LD, A, O, AB, OB)
   Free ······You can freely set a desired expansion condition for simulated ladder tasks by selecting LD, A, O, AB and OB. This condition can also be set as an expansion condition for tasks other than simulated ladder tasks.
   LD : LOAD
   A : AND
   O : OR
   AB : ANDBLOCK
   OB : ORBLOCK

2) Input condition (I/O, flag)
Free ······ You can freely set a desired input condition by selecting an input port, output port, input/output ports or flag (global area or local area).

3) Command/declaration
State a command/declaration command*1. The command explained in the applicable section is described.
*1 Once executed in the program, "Actuator Control Declaration" Command (VEL command, VELS command, etc.) will remain effective while the program is running, until the command is changed. If you want to change a value (operand 1, operand 2, etc.) previously set by an "Actuator Control Declaration" Command, you must reset (change) the value at the necessary location in the program.

4) Operand 1,

5) Operand 2
What is set in these items varies depending on the command. Set an appropriate item according to each command.

6) Output (output port, flag)
This is where the result of command execution is shown, and the output 6) turns ON and OFF*2. You can freely set an output port, input/output ports or flag (global area or local area) in which to store the condition of this output. What is turned ON/OFF in output varies depending on the command.
In the 6) Output (output port, flag), the following types are to be shown depending on the operational conditions.
(Output operation types)
CC ······ Command successful
ZR ······ Calculation result zero
PE ······ Operation complete
CP ······ Command passing
TU ······ Timeout
(CP☐☐ comparison command)
EQ ······ Operand 1 = Operand 2
NE ······ Operand 1 ≅ Operand 2
GT ······ Operand 1 > Operand 2
GE ······ Operand 1 ≥ Operand 2
LT ······ Operand 1 < Operand 2
LE ······ Operand 1 ≤ Operand 2

*2 The output turns OFF when the command is executed. After the command has been executed, the output turns ON depending on the condition specified as the output operation type. (If the condition is not met, the output remains OFF.)
Take note that the output of a CP☐☐ comparison command does not turn OFF when the command is executed.

## [2] Applicable models

Controllers that support the command are denoted by a "○".
Controllers that do not support the command are denoted by a "×".
The following controllers are applicable when described as "Applicable for all models".
• XSEL-J/K/JX/KX
• XSEL-P/Q/PX/QX/PCT/QCT
• XSEL-R/S/RX/SX/RXD/SXD
• XSEL-RA/SA/RAX/SAX/RAXD/SAXD
• TT/TTA
• ASEL/PSEL/SSEL
• MSEL-PC/PG/PCX/PGX

## [3] Description of functions

Explanation of the function is provided for the corresponding command.

## 5.2 SEL Language Code Table for each Function

For Operand 1, Operand 2 and the output, the variable indirect specification is available.
For the condition, Operand 1, Operand 2 and the output, an input with symbols is available.
Input into ( ) for Operand 1 and Operand 2 is not compulsory.
"Actuator control declaration" command is kept effective though the program run once it is executed during the program. A reconstruction of the settings is required for the appropriate areas in the program if a change to the values (Operand 1, Operand 2, etc.) already set by "actuator control declaration" command is needed. It means that the values set by the executed command in the last operation are effective.
The output section is turned OFF when the command is executed. After the command execution, it may get turned ON depending on the condition of the output section operation type. (It is turned OFF if the condition does not meet the requirement.)

> ⚠ Caution: Comparative command CP☐☐ (CPEQ, CPNE, CPGT, CPGE, CPLT, CPLE) output section does not get turned OFF during the command execution.

Output operation types
CC : Command successful, ZR: Calculation result zero
PE : Operation complete, CP: Command passing, TU: Timeout
EQ : Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT : Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT : Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

| Category | Condition | Command | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|
| Variable Assignment | Optional | LET | Variable Assignment | Assignment number | ZR | Assign | 264 |
| | Optional | TRAN | Variable to copy data to | Variable to copy data from | ZR | Copy | 265 |
| | Optional | CLR | Clear start variable | Clear finish variable | ZR | Clear variable | 266 |
| Arithmetic Operation | Optional | ADD | Augend variable | Addend | ZR | Add | 267 |
| | Optional | SUB | Minuend variable | Subtrahend | ZR | Subtract | 268 |
| | Optional | MULT | Multiplicand variable | Multiplier | ZR | Multiply | 269 |
| | Optional | DIV | Dividend variable | Divisor | ZR | Divide | 270 |
| | Optional | MOD | Modulus assignment variable | Divisor | ZR | Modulus calculation | 271 |
| Function Operation | Optional | SIN | Sine assignment variable | Operant [Radian] | ZR | Sine | 272 |
| | Optional | ASIN | Arc sine assignment variable | Operant | ZR | Inverse-sine | 273 |
| | Optional | COS | Cosine assignment variable | Operant [Radian] | ZR | Cosine | 274 |
| | Optional | ACOS | Arc cosine assignment variable | Operant | ZR | Inverse-cosine | 275 |
| | Optional | TAN | Tangent assignment variable | Operant [Radian] | ZR | Tangent | 276 |
| | Optional | ATN | Arc tangent assignment variable | Operant | ZR | Inverse-tangent | 277 |
| | Optional | SQR | Square root assignment variable | Operant | ZR | Root | 278 |
| | Optional | DTOR | Angle assignment variable | Operant | ZR | Angle conversion (degrees to radians) | 279 |
| | Optional | RTOD | Angle assignment variable | Operant | ZR | Angle conversion (radians to degrees) | 279 |
| | Optional | ABS | Absolute value assignment variable | Operant | ZR | Absolute value calculation | 280 |
| | Optional | SGN | Sign assignment variable | Operant | ZR | Get signs | 280 |
| Logical Operation | Optional | AND | Logical conjunction variable | Operant | ZR | Logical AND | 281 |
| | Optional | OR | Logical disjunction variable | Operant | ZR | Logical OR | 282 |
| | Optional | EOR | Logical operation exclusive disjunction | Operant | ZR | Logical exclusive-OR | 283 |
| | Optional | NOT | Variable No. | Data | ZR | Deny | 284 |
| | Optional | LSFT | Variable No. | Number of shifted bits | ZR | Logic Shifted to Left | 285 |
| | Optional | RSFT | Variable No. | Number of shifted bits | ZR | Logic Shifted to Right | 286 |
| Comparison | Optional | CP☐☐ | Comparative variable | Compared number | EQ NE GT GE LT LE | Compare [EQ/NE/GT/GE/LT/LE] | 287 |

Output operation types
CC : Command successful, ZR: Calculation result zero
PE : Operation complete, CP: Command passing, TU: Timeout
EQ : Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT : Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT : Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

| Category | Condition | Command | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|
| Timer | Optional | TIMW | Waiting time [sec] | Prohibited | TU | Wait for certain time | 288 |
| | Optional | TIMC | Program No. | Prohibited | CP | Cancel waiting | 289 |
| | Optional | GTTM | Time assignment variable | Prohibited | CP | Get time | 290 |
| I/O, Flag Operation | Optional | BT□□ | Start output, Flag | (Complete output, Flag) | CP | Output, flag [ON/OF/NT] | 291 |
| | Optional | BTPN | Output port, Flag | Timer setting | CP | Output ON pulse | 292 |
| | Optional | BTPF | Output port, Flag | Timer setting | CP | Output OFF pulse | 293 |
| | Optional | WT□□ | I/O, Flag | (Waiting time) | TU | Wait for input and output, flag [ON/OF] | 294 |
| | Optional | IN | Head I/O, Flag | Complete input and output, Flag | CC | Input binary number (Max 32 bit) | 295 |
| | Optional | INB | Head I/O, Flag | Convertible digits | CC | Input BCD (Max eight digits) | 296 |
| | Optional | OUT | Head output, Flag | Complete input and output, Flag | CC | Output binary number (Max 32 bit) | 297 |
| | Optional | OTPS | Output port No. | Axis No. | CC | Output current position data | 303 |
| | Optional | OUTB | Head output, Flag | Convertible digits | CC | Output BCD (Max eight digits) | 298 |
| | Optional | FMIO | Format type | Prohibited | CP | IN (B) OUT (B) command format | 299 |
| Program Control | Optional | GOTO | Tag No. to jump to | Prohibited | CP | Jump | 304 |
| | Prohibited | TAG | Declaration tag No. | Prohibited | CP | Declaration of destination to jump to | 305 |
| | Optional | EXSR | Execution sub routine No. | Prohibited | CP | Execute subroutine | 306 |
| | Prohibited | BGSR | Declaration sub routine No. | Prohibited | CP | Start subroutine | 307 |
| | Prohibited | EDSR | Prohibited | Prohibited | CP | End subroutine | 308 |
| Task Management | Optional | EXIT | Prohibited | Prohibited | CP | End program | 309 |
| | Optional | EXPG | Execution program No. | (Execution program No.) | CC | Start other program | 310 |
| | Optional | ABPG | Termination program No. | (Termination program No.) | CC | Abort other program | 311 |
| | Optional | SSPG | Pause program No. | (Pause program No.) | CC | Pause program | 312 |
| | Optional | RSPG | Resume program No. | (Resume program No.) | CC | Resume program | 313 |
| Position Operation | Optional | PGET | Axis No. | Position No. | CC | Assign position to Variable 199 | 314 |
| | Optional | PPUT | Axis No. | Position No. | CP | Assign Variable 199 value | 315 |
| | Optional | PCLR | Start position No. | Termination position No. | CP | Clear position data | 316 |
| | Optional | PCPY | Position No. to copy data to | Position No. to copy data from | CP | Copy position data | 317 |
| | Optional | PRED | Axis pattern read | Position No. to save data to | CP | Read current axis position | 318 |
| | Optional | PRDQ | Axis No. | Variable No. | CP | Read current axis position (single-axis direct) | 319 |
| | Optional | PTST | Axis pattern confirmation | Confirmation position No. | CC | Check position data | 321 |
| | Optional | PVEL | Speed [mm/sec] | Position No. to assign to | CP | Assign position speed | 322 |
| | Optional | PACC | Acceleration [G] | Position No. to assign to | CP | Assign position acceleration | 323 |
| | Optional | PDCL | Deceleration [G] | Position No. to assign to | CP | Assign position deceleration | 324 |
| | Optional | PAXS | Axis pattern assignment variable No. | Position No. | CP | Read axis pattern | 325 |
| | Optional | PSIZ | Size assignment variable No. | Prohibited | CP | Check position data size | 326 |
| | Optional | PTAM | Variable No. | Position No. | CP | Substitution of target arm system data | 320 |
| | Optional | GTAM | Variable No. | Position No. | CP | Acquirement of target arm system data | 327 |
| | Optional | GVEL | Variable No. | Position No. | CP | Get speed data | 328 |
| | Optional | GACC | Variable No. | Position No. | CP | Get acceleration data | 329 |
| | Optional | GDCL | Variable No. | Position No. | CP | Get deceleration data | 330 |

Output operation types
CC : Command successful, ZR: Calculation result zero
PE : Operation complete, CP: Command passing, TU: Timeout
EQ : Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT : Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT : Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

| Category | Condition | Command | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|
| Actuator Control Declaration | Optional | VEL | Speed [mm/sec] | Prohibited | CP | Set speed | 331 |
| | Optional | OVRD | Speed ratio [%] | (OVRD type) | CP | Speed coefficient settings | 333 |
| | Optional | ACC | Acceleration [G] | Prohibited | CP | Set acceleration | 335 |
| | Optional | DCL | Deceleration [G] | Prohibited | CP | Set deceleration | 337 |
| | Optional | SCRV | Ratio [%] | Prohibited | CP | Set sigmoid motion ratio | 339 |
| | Optional | OFST | Setting axis pattern | Offset value [mm] | CP | Set offset | 343 |
| | Optional | DEG | Division angle [deg] | Prohibited | CP | Division angle settings | 344 |
| | Optional | BASE | Datum axis No. | Prohibited | CP | Datum axis setting | 345 |
| | Optional | GRP | Effective axis pattern | Prohibited | CP | Set group axes | 346 |
| | Optional | HOLD | (Pause input port) | (HOLD type) | CP | Declare port to pause | 347 |
| | Optional | CANC | (Cancel complete input port) | (CANC type) | CP | Declare port to abort | 348 |
| | Optional | CLLV | Axis pattern | Collision Detection Level | CP | Collision Detection Level Setting | 349 |
| | Optional | COL | 0 or 1 | (Axis pattern) | CP | Collision Detection Feature Valid / Invalid Setting | 350 |
| | Optional | VLMX | Prohibited | Prohibited | CP | Specify VLMX speed | 355 |
| | Optional | ACMX | ACMX Acceleration No. | Prohibited | CP | Indicate ACMX acceleration | 352 |
| | Optional | DIS | Distance | Prohibited | CP | Set division distance at spline movement | 356 |
| | Optional | POTP | 0 or 1 | Prohibited | CP | Set PATH output type | 357 |
| | Optional | PAPR | Distance | Speed | CP | PUSH Command distance and speed settings | 358 |
| | Optional | QRTN | 0 or 1 | Prohibited | CP | Set quick-return mode | 359 |
| | Optional | ACCS | Ratio | Prohibited | CP | Set acceleration ratio | 336 |
| | Optional | DCLS | Ratio | Prohibited | CP | Set deceleration ratio | 338 |
| | Optional | DFIF | Contact check zone No. | Position No. | CP | Define simple contact check zone coordinate | 386 |
| | Optional | DFTL | Tool coordinate system No. | Position No. | CP | Define tool coordinate system | 365 |
| | Optional | DFWK | Work coordinate system No. | Position No. | CP | Define work coordinate system | 372 |
| | Optional | GCLX | Variable No. | Axis No. | CP | Acquiring Max. Collision Level | 351 |
| | Optional | GTIF | Contact check zone No. | Position No. | CP | Get simple contact check zone definition coordinate | 390 |
| | Optional | GTTL | Tool coordinate system No. | Position No. | CP | Get tool coordinate system definition data | 370 |
| | Optional | GTWK | Work coordinate system No. | Position No. | CP | Get work coordinate system definition number | 377 |
| | Optional | NBND | Axis pattern | Close distance | CP | Set close distance | 395 |
| | Optional | PTPD | Prohibited | Prohibited | CP | Specify PTP target arm system to current arm | 384 |
| | Optional | SLTL | Tool coordinate system No. | Prohibited | CP | Select tool coordinate system | 368 |
| | Optional | SEIF | Contact check zone No. | 0 to 2 | CP | Specify type of simple contact check zone | 389 |
| | Optional | RIGH | Prohibited | Prohibited | PE | Change current arm system to right arm | 380 |
| | Optional | LEFT | Prohibited | Prohibited | PE | Change current arm system to left arm | 381 |
| | Optional | PTPR | Prohibited | Prohibited | CP | Specify PTP target arm system to right arm | 382 |
| | Optional | PTPE | Prohibited | Prohibited | CP | Specify PTP target arm system to current arm | 385 |
| | Optional | WGHT | Mass | (Inertial moment) | CP | Set tip work mass, inertial moment | 391 |
| | Optional | WGT2 | Mass | (Inertial moment) | CP | Tip load condition setting 2 | 393 |
| | Optional | VELS | Ratio | Prohibited | CP | Set speed ratio | 332 |
| | Optional | SOIF | Contact check zone No. | Output, global flag No. | CP | Specify output for simple contact check zone | 388 |
| | Optional | SLWK | Work coordinate system No. | Prohibited | CP | Select work coordinate system | 375 |
| | Optional | PTPL | Prohibited | Prohibited | CP | Specify PTP target arm system to left arm | 383 |
| | Optional | CNTP | Other numbers | Prohibited | CP | PTP Continuous Operation Mode Setting | 410 |

Output operation types
CC : Command successful, ZR: Calculation result zero
PE : Operation complete, CP: Command passing, TU: Timeout
EQ : Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT : Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT : Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

| Category | Condition | Command | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|
| Actuator Control Command | Optional | SV☐☐ | Operation axis pattern | Prohibited | PE | Turn ON/OFF servo | 397 |
| | Optional | HOME | Home-return axis pattern | Prohibited | PE | Home return | 398 |
| | Optional | MOVP | Position No. to move to | Prohibited | PE | Move by specifying position data | 399 |
| | Optional | MOVL | Position No. to move to | Prohibited | PE | Position-indicated interpolation movement | 401 |
| | Optional | MVPI | Movement amount position No. | Prohibited | PE | Position-relative movement | 403 |
| | Optional | MVLI | Movement amount position No. | Prohibited | PE | Position-relative interpolation movement | 405 |
| | Optional | PATH | Start position No. | End position No. | PE | Move along path | 409 |
| | Optional | J☐W☐ | Drive axis pattern | Start input and output, flag | PE | Jog [FN/FF/BN/BF] | 414 |
| | Optional | STOP | Stop axis pattern | Prohibited | CP | Deceleration and stop of axis | 417 |
| | Optional | PSPL | Start position No. | End position No. | PE | Move along spline | 418 |
| | Optional | PUSH | Target position No. | Prohibited | PE | Move by push motion | 419 |
| | Optional | PTRQ | Axis pattern | Ratio [%] | CC | Change push torque limit parameter | 421 |
| | Optional | CIR2 | Passing position 1 No. | Passing position 2 No. | PE | Circle movement 2 (Arc interpolation) | 422 |
| | Optional | ARC2 | Passing position No. | End position No. | PE | Arc movement 2 (Arc interpolation) | 424 |
| | Optional | CIRS | Passing position 1 No. | Passing position 2 No. | PE | Move along circle three-dimensionally | 426 |
| | Optional | ARCS | Passing position No. | End position No. | PE | Move along arc three-dimensionally | 428 |
| | Optional | CHVL | Axis pattern | Speed | CP | Change speed | 430 |
| | Optional | ARCD | End position No. | Center angle [°(degree)] | PE | Termination position center angle indicated arc movement | 432 |
| | Optional | ARCC | Center position No. | Center angle [°(degree)] | PE | Center position center angle indicated arc movement | 434 |
| | Optional | PBND | Axis pattern | Distance | CP | Set positioning width | 436 |
| | Optional | CIR | Passing position 1 No. | Passing position 2 No. | PE | Circle movement (CIR2 is recommended) | 439 |
| | Optional | ARC | Passing position No. | End position No. | PE | Arc movement (ARC2 is recommended) | 441 |
| | Optional | PEND | Prohibited | Prohibited | PE | Wait for end of operation by axes currently used by program | 443 |
| | Optional | MOVD | Target position | (Axis pattern) | PE | Move via direct value specification | 407 |
| | Optional | MVDI | Travel distance | (Axis pattern) | PE | Move relatively via direct value specification | 408 |
| | Optional | TMLI | Position No. | Prohibited | PE | Move incrementally to position on tool coordinate system via CP operation | 438 |
| | Optional | TMPI | Position No. | Prohibited | PE | Move incrementally to position on tool coordinate system via PTP operation | 437 |
| IF structure | Optional | IF☐☐ | Comparative variable | Compared No. | CP | Compare [EQ/NE/GT/GE/LT/LE] | 444 |
| | Optional | IS☐☐ | Column No. | Column No., character literal | CP | Compare strings | 445 |
| | Prohibited | ELSE | Prohibited | Prohibited | CP | Declaration of IF Command unsuccessful execution destination | 446 |
| | Prohibited | EDIF | Prohibited | Prohibited | CP | IF termination declaration | 447 |
| Structural DO | Optional | DW☐☐ | Comparative variable | Compared No. | CP | Loop [EQ/NE/GT/GE/LT/LE] | 448 |
| | Optional | LEAV | Prohibited | Prohibited | CP | Pull out of DO | 449 |
| | Optional | ITER | Prohibited | Prohibited | CP | Repeat of DO | 450 |
| | Prohibited | EDDO | Prohibited | Prohibited | CP | DO termination declaration | 451 |
| Multi-Branching | Optional | SLCT | Prohibited | Prohibited | CP | Start declaration for multi-branching | 452 |
| | Prohibited | WH☐☐ | Comparative variable | Compared No. | CP | Branch values [EQ/NE/GT/GE/LT/LE] | 453 |
| | Prohibited | WS☐☐ | Column No. | Column No., character literal | CP | Branch character line [EQ/NE] | 454 |
| | Prohibited | OTHE | Prohibited | Prohibited | CP | Declaration of condition unsuccessful branching destination | 455 |
| | Prohibited | EDSL | Prohibited | Prohibited | CP | SLCT termination declaration | 456 |
| Systeml nformation Acquisition | Optional | AXST | Variable No. | Axis No. | CP | Get axis status | 457 |
| | Optional | PGST | Variable No. | Program No. | CP | Get program status | 458 |
| | Optional | SYST | Variable No. | Prohibited | CP | Get system status | 459 |
| | Optional | GARM | Variable No. | Prohibited | CP | Get current arm system | 460 |

Output operation types
CC : Command successful, ZR: Calculation result zero
PE : Operation complete, CP: Command passing, TU: Timeout
EQ : Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT : Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT : Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

| Category | Condition | Command | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|
| Zone | Optional | WZNA | Zone No. | Axis pattern | CP | Wait for zone ON based on AND gate | 461 |
| | Optional | WZNO | Zone No. | Axis pattern | CP | Wait for zone ON based on OR gate | 462 |
| | Optional | WZFA | Zone No. | Axis pattern | CP | Wait for zone OFF based on AND gate | 463 |
| | Optional | WZFO | Zone No. | Axis pattern | CP | Wait for zone OFF based on OR gate | 464 |
| Communication | Optional | OPEN | Channel No. | Prohibited | CP | Open channel | 465 |
| | Optional | CLOS | Channel No. | Prohibited | CP | Close channel | 466 |
| | Optional | READ | Channel No. | Column No. | CC | Input from channel | 467 |
| | Optional | TMRW | Read timer setting | (Write timer setting) | CP | Set READ timeout value | 471 |
| | Optional | WRIT | Channel No. | Column No. | CC | Output to channel | 473 |
| | Optional | SCHA | Character code | Prohibited | CP | Character setting for sending and receiving | 474 |
| | Optional | TMRD | Timer period | Prohibited | CP | Set READ timeout value | 469 |
| | Optional | IPCN | Channel No. | Integer variable No. | CP | Connected Destination IP address / Port Number Setting | 475 |
| String Operation | Optional | SCPY | Column No. | Column No., character literal | CC | Copy character string | 476 |
| | Optional | SCMP | Column No. | Column No., character literal | EQ | Compare character strings | 477 |
| | Optional | SGET | Variable No. | Column No., character literal | CP | Get character | 478 |
| | Optional | SPUT | Column No. | Data | CP | Set character | 479 |
| | Optional | STR | Column No. | Data | CC | Convert character string; decimal | 480 |
| | Optional | STRH | Column No. | Data | CC | Convert character string; hexadecimal | 481 |
| | Optional | VAL | Variable No. | Column No., character literal | CC | Convert character string data; decimal | 482 |
| | Optional | VALH | Variable No. | Column No., character literal | CC | Convert character string data; hexadecimal | 483 |
| | Optional | SLEN | Character string length | Prohibited | CP | Set length | 484 |
| Palletizing Definition | Optional | BGPA | Palletizing No. | Prohibited | CP | Declare start of palletizing setting | 491 |
| | Prohibited | EDPA | Prohibited | Prohibited | CP | Declare end of palletizing setting | 492 |
| | Optional | PAPI | Count | Count | CP | Set palletizing counts | 493 |
| | Optional | PAPN | Pattern No. | Prohibited | CP | Set palletizing pattern | 494 |
| | Optional | PASE | Axis No. | Axis No. | CP | Declare palletizing axes | 495 |
| | Optional | PAPT | Pitch | Pitch | CP | Set palletizing pitches | 496 |
| | Optional | PAST | (Position No.) | Prohibited | CP | Set palletizing reference point | 497 |
| | Optional | PAPS | Position No. | Palletizing position setting type | CP | Set palletizing points for 3-point or 4-point teaching | 498 |
| | Optional | PSLI | Offset amount | (Count) | CP | Set zigzag | 501 |
| | Optional | PCHZ | (Axis No.) | Prohibited | CP | Declare palletizing Z-axis | 502 |
| | Optional | PTRG | Position No. | Position No. | CP | Set palletizing arch triggers | 503 |
| | Optional | PEXT | (Position No.) | Prohibited | CP | Set composite palletizing | 504 |
| | Optional | OFPZ | Offset amount | Prohibited | CP | Set palletizing Z-axis offset | 505 |
| | Optional | ACHZ | Axis No. | Prohibited | CP | Declare arch-motion Z-axis | 487 |
| | Optional | ATRG | Position No. | Position No. | CP | Set arch triggers | 488 |
| | Optional | AEXT | (Position No.) | Prohibited | CP | Set composite arch motion | 489 |
| | Optional | OFAZ | Offset amount | Prohibited | CP | Set arch-motion Z-axis offset | 490 |
| | Optional | PTNG | Palletizing No. | Variable No. | CP | Get palletizing position number | 506 |
| | Optional | PINC | Palletizing No. | Prohibited | CC | Increment palletizing position number by 1 | 507 |
| | Optional | PDEC | Palletizing No. | Prohibited | CC | Decrement palletizing position number by 1 | 508 |
| | Optional | PSET | Palletizing No. | Data | CC | Set palletizing position number directly | 509 |
| | Optional | PARG | Palletizing No. | Axis No. | CP | Get palletizing angle | 510 |
| | Optional | PAPG | Palletizing No. | Position No. | CP | Get palletizing calculation data | 511 |
| | Optional | PMVP | Palletizing No. | (Position No.) | PE | Move to palletizing points via PTP | 512 |
| | Optional | PMVL | Palletizing No. | (Position No.) | PE | Move to palletizing points via interpolation | 513 |
| | Optional | PACH | Palletizing No. | Position No. | PE | Arch motion to palletizing point | 514 |
| | Optional | ARCH | Position No. | Position No. | PE | Arch motion | 485 |

Output operation types
CC : Command successful, ZR: Calculation result zero
PE : Operation complete, CP: Command passing, TU: Timeout
EQ : Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT : Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT : Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

| Category | Condition | Command | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|
| Building of Pseudo-Ladder Task | Optional | CHPR | 0 or 1 | Prohibited | CP | Change task level | 516 |
| | Prohibited | TPCD | 0 or 1 | Prohibited | CP | Specify processing to be performed when input condition is not specified | 517 |
| | Prohibited | TSLP | Time | Prohibited | CP | Task sleep | 518 |
| | Optional | OUTR | Output, flag No. | Prohibited | CP | Ladder output relay | 175 |
| | Optional | TIMR | Local flag No. | Timer setting | CP | Ladder timer relay | 175 |
| Extended Command | Optional | ECMD | 1 | Axis No. | CC | Get motor current value | 519 |
| | Optional | ECMD | 2 | Axis No. | CC | Get home sensor status | 520 |
| | Optional | ECMD | 3 | Axis No. | CC | Get overrun sensor status | 521 |
| | Optional | ECMD | 4 | Axis No. | CC | Get creep sensor status | 522 |
| | Optional | ECMD | 5 | Axis No. | CC | Get axis operation status | 523 |
| | Optional | ECMD | 6 | Axis No. | CC | Current position acquirement on each axis system | 524 |
| | Optional | ECMD | 7 | Axis No. | CC | Get total movement count | 525 |
| | Optional | ECMD | 8 | Axis No. | CC | Get total mileage | 526 |
| | Optional | ECMD | 9 | Axis No. | CC | Get position deviation | 527 |
| | Optional | ECMD | 10 | Axis No. | CC | Acquirement of Overload Level | 528 |
| | Optional | ECMD | 11 | Axis No. | CC | Acquirement of Encoder Overheated Level | 529 |
| | Optional | ECMD | 20 | Axis No. | CC | Get parameter value | 530 |
| | Optional | ECMD | 250 | Integer variable No. | CC | Set torque limit/torque limit over detection time | 532 |
| | Optional | ECMD | 280 | Integer variable No. | CC | Conversion from each axis coordinates to work coordinates in wrist unit equipped robot | 535 |
| | Optional | ECMD | 281 | Integer variable No. | CC | Conversion from work coordinates to each axis coordinates in wrist unit equipped robot | 536 |
| | Optional | ECMD | 282 | Integer variable No. | CC | Conversion from tool coordinates to work coordinates in wrist unit equipped robot | 538 |
| | Optional | ECMD | 290 | Integer variable No. | CC | Conversion from each axis coordinates to work coordinates in wrist unit equipped robot | 535 |
| | Optional | ECMD | 291 | Integer variable No. | CC | Conversion from work coordinates to each axis coordinates in wrist unit equipped robot | 536 |
| | Optional | ECMD | 292 | Integer variable No. | CC | Conversion from tool coordinates to work coordinates in wrist unit equipped robot | 538 |
| | Optional | ECMD | 300 | Integer variable No. | CC | User system error output | 540 |
| Vision System I/F Related | Optional | SLVS | Select using Vision System I/F | (Timeout time) | CC | Select Vision System I/F | 631 |
| | Optional | GTVD | Capturing Trigger Classification | Variable No. | CC | Vision System I/F Image-Capture Data Acquirement | 633 |
| Conveyor Tracking Related | Optional | TRMD | Select using Tracking Mode | TRAC Command timeout time | CC | Tracking Mode Setting | 627 |
| | Optional | TRAC | 0 or 1 | Position No. to save the work position information | CC | Tracking Operation Setting & Datum Point Position Information Obtainment in Work | 628 |
| Anti-Vibration Control Related | Optional | NTCH | Axis pattern | Parameter set number | CC | Anti-Vibration Control Parameter Set Select | 635 |
| Compliance Control | Optional | COMP | Mode Type | (Axis pattern) | CP | Compliance Mode Setting | 636 |
| | Optional | SCLO | 0 | Integer variable No. | CP | Compliance Mode Option Feature Setting (Searching Operation Setting) | 638 |
| | Optional | SCLO | 1 | Integer variable No. | CP | Compliance Mode Option Feature Setting (J1 & J2-Axes Torque Limit Mode Setting) | 640 |
| | Optional | SCLG | Integer variable No. | Prohibited | CP | Compliance Gain Setting | 641 |

# RC Gateway Function Commands (Controller with Gateway Function Only)

* Refer to “XSEL Controller P/Q/PX/QX RC Gateway Function Instruction Manual” for the commands related to RC gateway functions.

Output operation types
CC : Command successful, ZR: Calculation result zero
PE : Operation complete, CP: Command passing, TU: Timeout
EQ : Operand 1 = Operand 2, NE: Operand 1 ≅ Operand 2
GT : Operand 1 > Operand 2, GE: Operand 1 ≥ Operand 2
LT : Operand 1 < Operand 2, LE: Operand 1 ≤ Operand 2

| Category | Condition | Command | RC position-data use mode | | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|---|---|
| | | | XSEL | RC | | | | | |
| RC axis position operation | Optional | RPGT | ○ | × | RC-axis No. | Position No. | CC | Assign RC axis position location to Variable 199 | 541 |
| | Optional | RPPT | ○ | × | RC-axis No. | Position No. | CP | Assign Variable 199 to RC axis position location | 542 |
| | Optional | RPCR | ○ | × | RC-axis No. | Variable No. | CP | Clear RC-axis position data | 543 |
| | Optional | RPCP | ○ | × | RC-axis No. | Variable No. | CP | Copy RC-axis position data | 544 |
| | Optional | RPRD | ○ | × | Position No. | Prohibited | CP | Read current RC-axis position | 545 |
| | Optional | RPRQ | ○ | ○ | RC-axis No. | Variable No. | CP | Read current RC-axis position (single-axis direct) | 546 |
| | Optional | RPVL | ○ | × | RC-axis No. | Position No. | CP | Assign Variable 199 to RC axis position speed | 547 |
| | Optional | RPAD | ○ | × | RC-axis No. | Position No. | CP | Assign Variable 199 to RC axis position acceleration/deceleration | 548 |
| | Optional | RPIP | ○ | × | RC-axis No. | Position No. | CP | Assign Variable 199 to RC axis position positioning width | 549 |
| | Optional | RPTQ | ○ | × | RC-axis No. | Position No. | CP | Assign Variable 199 to RC axis position current limitation | 550 |
| | Optional | RGVL | ○ | × | RC-axis No. | Position No. | CP | Assign RC axis position speed to Variable 199 | 551 |
| | Optional | RGAD | ○ | × | RC-axis No. | Position No. | CP | Assign RC axis position acceleration/deceleration to Variable 199 | 552 |
| | Optional | RGIP | ○ | × | RC-axis No. | Position No. | CP | Assign RC position positioning width to Variable 199 | 553 |
| | Optional | RGTQ | ○ | × | RC-axis No. | Position No. | CP | Assign RC position current limitation to Variable 199 | 554 |
| RC actuator control command | Optional | RAXS | ○ | ○ | Axis pattern, upper | Axis pattern, lower | CP | Set axis pattern for RC axis | 555 |
| | Optional | RSON | ○ | ○ | Prohibited | Prohibited | PE | Turn ON RC-axis servo | 556 |
| | Optional | RSOF | ○ | ○ | Prohibited | Prohibited | PE | Turn OFF RC-axis servo | 557 |
| | Optional | RHOM | ○ | ○ | Prohibited | Prohibited | PE | Return RC-axis to home | 558 |
| | Optional | RMVP | ○ | ○ | Position No. | Prohibited | PE | Move RC-axis by position specification | 559 |
| | Optional | RMPI | ○ | × | Position No. | Prohibited | PE | Move RC-axis incrementally by position specification | 560 |
| | Optional | RMVD | ○ | × | RC-axis No. | Variable No. | PE | Move RC axis with direct specification | 561 |
| | Optional | RMDI | ○ | × | RC-axis No. | Variable No. | PE | Move RC axis to directly specified relative position | 562 |
| | Optional | RPUS | ○ | × | RC-axis No. | Position No. | PE | Move RC-axis via push motion | 563 |
| | Optional | RSTP | ○ | ○ | Prohibited | Prohibited | PE | Decelerate and stop RC axis | 564 |
| RC axis information acquisition | Optional | RCST | ○ | ○ | Variable No. | RC-axis No. | PE | Acquire RC axis status | 565 |

# Extension Motion Control Function Related Commands

* Refer to "XSEL Controller P/Q/PCT/QCT Electronic Cam function Instruction Manual" (Controller with Electronic Cam Function Only) for the details of the commands related to the electronic cam functions.
* Refer to "Extension Motion Control Function Instruction Manual" for the details of the extension motion control commands.

Output operation types
CC: Command successful, CP: Command passing
PE: Operation complete

| Category | Condition | Command | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|
| Extension Motion Control Board Input Operations | Optional | XCRP | Pulse input channel No. | Prohibited | CP | Clear input counter record for extension motion control | 567 |
| | Optional | XGTP | Pulse input channel No. | Prohibited | CP | Acquire current record of extension motion control input counter | 568 |
| Extension Motion Control Board Axis Position Operations | Optional | XPGT | Axis No. | Position No. | CC | Read extension motion control axis position data | 569 |
| | Optional | XPPT | Axis No. | Position No. | CP | Write extension motion control axis position data | 570 |
| | Optional | XPCR | Axis No. | Variable No. | CP | Erase extension motion control axis position data | 571 |
| | Optional | XPCP | Axis No. | Variable No. | CP | Copy extension motion control axis position data | 572 |
| | Optional | XPRD | Position No. | Prohibited | CP | Read extension motion control axis current command position | 573 |
| | Optional | XPRQ | Axis No. | Variable No. | CP | Read extension motion control axis current command position (single-axis direct) | 574 |
| | Optional | XPVL | Axis No. | Position No. | CP | Write extension motion control axis speed data | 575 |
| | Optional | XPAC | Axis No. | Position No. | CP | Write extension motion control axis acceleration data | 576 |
| | Optional | XPDC | Axis No. | Position No. | CP | Write extension motion control axis deceleration data | 577 |
| | Optional | XPIP | Axis No. | Position No. | CP | Write extension motion control axis positioning complete width data | 578 |
| | Optional | XGVL | Axis No. | Position No. | CP | Read extension motion control axis speed data | 579 |
| | Optional | XGAC | Axis No. | Position No. | CP | Read extension motion control axis acceleration data | 580 |
| | Optional | XGDC | Axis No. | Position No. | CP | Read extension motion control axis deceleration data | 581 |
| | Optional | XGIP | Axis No. | Position No. | CP | Read extension motion control axis positioning complete width data | 582 |
| Extension Motion Control Board Axis Actuator Control Declarations | Optional | XAXS | Axis pattern, upper | Axis pattern, lower | CP | Set each pulse I/O axis pattern (0 to 15 axis) | 583 |
| | Optional | XA16 | Axis pattern, upper | Axis pattern, lower | CP | Set each pulse I/O axis pattern (16 to 31 axis) | 584 |
| Extension Motion Control Board Axis Actuator Control Commands | Optional | XSON | Prohibited | Prohibited | PE | Extension motion control axis to servo ON | 585 |
| | Optional | XSOF | Prohibited | Prohibited | PE | Extension motion control axis to servo OFF | 586 |
| | Optional | XHOM | Prohibited | Prohibited | PE | Extension motion control axis to home return | 587 |
| | Optional | XMVP | Position No. | Prohibited | PE | Move extension motion control axis to indicated position | 588 |
| | Optional | XMPI | Position No. | Prohibited | PE | Perform extension motion control axis position relative movement | 589 |
| | Optional | XMVL | Position No. | Prohibited | PE | Move extension motion control axis for position indicated interpolation | 590 |
| | Optional | XMLI | Position No. | Prohibited | PE | Move extension motion control axis for position relative interpolation | 591 |
| | Optional | XMVD | Axis No. | Variable No. | PE | Move extension motion control axis to directly indicated absolute position | 592 |
| | Optional | XMDI | Axis No. | Variable No. | PE | Move extension motion control axis to directly indicated relative position | 593 |
| | Optional | XPTH | Start position No. | End position No. | PE | Extension motion control axis path operation | 594 |
| | Optional | XJ□□ | Input, output, flag No. | Prohibited | PE | Perform extension motion control axis jog operation [FN/FF/BN/BF] | 596 |
| | Optional | XPED | Prohibited | Prohibited | PE | Waiting for extension motion control axis to finish positioning operation of axis used by self-program | 597 |

Output operation types
CC: Command successful, CP: Command passing
PE: Operation complete

| Category | Condition | Command | Operand 1 | Operand 2 | Output | Function | Page |
|---|---|---|---|---|---|---|---|
| Extension Motion Control Board Axis Actuator Control Commands | Optional | XSTP | Prohibited | Prohibited | PE | Cancel operation of extension motion control axis | 598 |
| | Optional | XWIP | Prohibited | Prohibited | CP | Waiting for extension motion control axis positioning complete signal to be turned ON | 599 |
| | Optional | XCAS | Slave Shaft No. | Variable No. | PE | Start synchronizing extension motion control axis electronic cam (indicating main axis) | 601 |
| | Optional | XCTM | Slave Shaft No. | Variable No. | PE | Move extension motion control axis individual electronic cam (indicating time) | 612 |
| | Optional | XSFS | Slave Shaft No. | Variable No. | PE | Start synchronizing of extension motion control axis electronic shaft | 614 |
| | Optional | XSYE | Slave Shaft No. | (Complete Type) | PE | Cancel operation of extension motion control axis | 617 |
| Extension Motion Control Board Axis Status Acquirement | Optional | XAST | Variable No. | Axis No. | CP | Acquire extension motion control axis status | 619 |
| Arch Motion Related | Optional | XACH | Position No. | Position No. | PE | Extension motion control axis arch motion | 622 |
| | Optional | XACZ | Axis No. | Prohibited | CP | Extension motion control axis arch motion Z-axis declaration | 624 |
| | Optional | XAEX | (Position No.) | Prohibited | CP | Extension motion control axis arch motion composition setting | 624 |
| | Optional | XATG | Position No. | Position No. | CP | Extension motion control axis arch trigger setting | 625 |
| | Optional | XOAZ | Offset value | Prohibited | CP | Extension motion control axis arch motion Z-axis offset setting | 626 |

## 5.3 Explanation of Commands

### [1] Variable Assignment

### ● LET (Assign)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | LET | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Assign the value specified in operand 2 to the variable specified in operand 1.
The output will turn ON when 0 is assigned to the variable specified in operand 1.

[Example 1] LET 1 10 Assign 10 to variable 1.

[Example 2]

```
LET    1     2      Assign 2 to variable 1.
LET    3     10     Assign 10 to variable 3.
LET    *1    *3     Assign the content 10 of variable 3 to the variable
                    that corresponds to the content 2 of variable 1.
```

## ● TRAN (Copy)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TRAN | Variable number | Variable number | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Assign the content of the variable specified in operand 2 to the variable specified in operand 1.
The output will turn ON when 0 is assigned to the variable specified in operand 1.

[Example 1]

| | | | |
|---|---|---|---|
| TRAN | 1 | 2 | Assign the content of variable 2 to variable 1. The above operation can be performed with a LET command as follows. |
| LET | 1 | *2 | |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| LET | 3 | 4 | Assign 4 to variable 3. |
| LET | 4 | 10 | Assign 10 to variable 4. |
| TRAN | *1 | *3 | Assign 10 of the content 4 of variable 3 to the variable that corresponds to the content 2 of variable 1. |

The variables change as follows.

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| 2 | 3 | 4 | 10 |

→

| 1 | 2 | 3 | 4 |
|---|---|---|---|
| 2 | 10 | 4 | 10 |

## ● CLR (Clear variable)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CLR | Variable number | Variable number | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Clear the variables from the one specified in operand 1 through the other specified in operand 2.
The contents of the variables that have been cleared become 0.
The output will turn ON when 0 is assigned to the variable specified in operand 1.

| | | | | |
|---|---|---|---|---|
| [Example 1] | CLR | 1 | 5 | Clear variables 1 through 5. |
| [Example 2] | LET | 1 | 10 | Assign 10 to variable 1. |
| | LET | 2 | 20 | Assign 20 to variable 2. |
| | CLR | *1 | *2 | Clear the variables from the contents 10 in variable 1 through the contents 20 in variable 2. |

## [2] Arithmetic Operation

### ● ADD (Add)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ADD | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Add the content of the variable specified in operand 1 and the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

| | | | |
|---|---|---|---|
| LET | 1 | 3 | Assign 3 to variable 1. |
| ADD | 1 | 2 | Add 2 to the content of variable 1 (3).<br>5 (3 + 2 = 5) will be stored in variable 1. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 2 | Assign 3 to variable 2. |
| LET | 3 | 2 | Assign 2 to variable 3. |
| ADD | *1 | *3 | Add the content of variable 3, or 2, to the variable that corresponds to the content of variable 1, or 2.<br>3 + 2, or 5, is stored in variable 2. |

## ● SUB (Subtract)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SUB | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Subtract the value specified in operand 2 from the content of the variable specified in operand 1, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

| | | | |
|---|---|---|---|
| LET | 1 | 3 | Assign 3 to variable 1. |
| SUB | 1 | 2 | Subtract 2 from the content of variable 1 (3).<br>1 (3 – 2 = 1) will be stored in variable 1. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| LET | 3 | 2 | Assign 2 to variable 3. |
| SUB | *1 | *3 | Subtract the content of variable 3 (2), from the variable corresponding to the content of variable 1 (2). 1 (3 - 2 = 1) will be send in variable 2. |

## ● MULT (Multiply)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | MULT | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Multiply the content of the variable specified in operand 1 by the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

```
LET   1  3     Assign 3 to variable 1.
MULT  1  2     Multiply the content of variable 1 (3) by 2.
               3 × 2, or 6, is stored in variable 1.
```

[Example 2]

```
LET   1   2    Assign 2 to variable 1.
LET   2   3    Assign 3 to variable 2.
LET   3   2    Assign 2 to variable 3.
MULT  *1  *3   Multiply the variable that corresponds to the content
               of variable 1, or 2, by the content of variable 3, or 2.
               3 × 2, or 6, is stored in variable 2.
```

## ● DIV (Divide)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DIV | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Divide the content of the variable specified in operand 1 by the value specified in operand 2, and assign the result to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

(Note) If the variable specified in operand 1 is an integer variable, any decimal places will be rounded off.

[Example 1]

```
LET   1    6     Assign 6 to variable 1.
DIV   1    2     Divide the content of variable 1 (6) by 2.
                 6 / 2, or 3, is stored in variable 1.
```

[Example 2]

```
LET   1    2     Assign 2 to variable 1.
LET   2    6     Assign 6 to variable 2.
LET   3    2     Assign 2 to variable 3.
DIV   *1   *3    Divide the variable that corresponds to the content
                 of variable 1, or 2, by the content of variable 3, or 2.
                 6 / 2, or 3, is stored in variable 2.
```

## ● MOD (Remainder)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | MOD | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Assign, to the variable specified in 1, the remainder obtained by dividing the content of the variable specified in operand 1 by the value specified in operand 2. The output will turn ON when the operation result becomes 0.

(Note) A MOD command is used with integer variables.

[Example 1]

| | | | |
|---|---|---|---|
| LET | 1 | 7 | Assign 7 to variable 1. |
| MOD | 1 | 3 | Obtain the remainder of dividing the content of variable 1 (7) by 3.<br>The remainder of 7 / 3 = 2, or 1, is assigned to variable 1. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 7 | Assign 7 to variable 2. |
| LET | 3 | 3 | Assign 3 to variable 3. |
| MOD | *1 | *3 | Obtain the remainder of dividing the variable that corresponds to the content of variable 1, or 2, by the content of variable 3, or 3.<br>The remainder of 7 / 3 = 2, or 1, is assigned to variable 2. |

## [3] Function Operation

### ● SIN (Sine operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SIN | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Assign the sine of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle $\times$ $\pi$ / 180

[Example 1]

```
SIN     100    0.523599
```

Assign the sine of 0.523599 (0.5) to variable 100.

[Example 2]

```
LET     1      100
LET     101    30
MULT    101    3.141592
DIV     101    180
SIN     *1     *101
```

Assign 100 to variable 1.
30 $\times$ $\pi$ / 180 (radian)
(30° is converted to radian and the result is assigned to variable 101.)
Assign the sine of the content of variable 101, or 0.5, to the variable that corresponds to the content of variable 1, or 100.

## ● ASIN (Inverse-sine (Arc sine) operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ASIN | Variable number | Data | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.10 or later) |



[Function] Assign the inverse sine (arc sine) of the data specified in operand 2 to the variable specified in operand 1.
A number indicated in operand 2 should be in the range from -1 to 1. An error will occur if the number in operand 2 is smaller than -1 or bigger than 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of inverse sine is radian. The range should be from $-\pi/2$ to $\pi/2$.

[Example] ASIN 100 1 Assign the inverse sine of 1 to variable 100.

## ● COS (Cosine operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | COS | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Assign the cosine of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle × π / 180

| | | | | |
|---|---|---|---|---|
| [Example 1] | COS | 100 | 1.047197 | Assign the cosine of 1.047197 (0.5) to variable 100. |
| [Example 2] | LET | 1 | 100 | Assign 100 to variable 1. |
| | LET | 101 | 60 | 60 × π / 180 (radian) |
| | MULT | 101 | 3.141592 | (60° is converted to radian and the result is |
| | DIV | 101 | 180 | assigned to variable 101.) |
| | COS | *1 | *101 | Assign the sine of the content of variable 101, or 0.5, to the variable that corresponds to the content of variable 1, or 100. |

## ● ACOS (Inverse-cosine (Arc cosine) operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ACOS | Variable number | Data | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.10 or later) |



[Function] Assign the inverse cosine (arc cosine) of the data specified in operand 2 to the variable specified in operand 1.
A number indicated in operand 2 should be in the range from -1 to 1. An error will occur if the number in operand 2 is smaller than -1 or bigger than 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of inverse cosine is radian. The range should be from 0 to $\pi$.

[Example] ACOS 100 1 Assign the inverse cosine of 1 to variable 100.

## ● TAN (Tangent operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TAN | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Assign the tangent of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of data in operand 2 is radian.

(Note 1) Radian = Angle × π / 180

[Example 1]

| | | | |
|---|---|---|---|
| TAN | 100 | 0.785398 | Assign the tangent of 0.785398 (1) to variable 100. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 100 | Assign 100 to variable 1. |
| LET | 101 | 45 | 45 × π / 180 (radian) |
| MULT | 101 | 3.141592 | (45° is converted to radian and the result is |
| DIV | 101 | 180 | assigned to variable 101.) |
| TAN | *1 | *101 | Assign the sine of the content of variable 101, or 1, to the variable that corresponds to the content of variable 1, or 100. |

## ● ATN (Inverse-tangent operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ATN | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Assign the inverse tangent of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.
The unit of inverse tangent is radian.

(Note 1) Radian = Angle $\times \pi$ / 180

[Example 1]

| | | | |
|---|---|---|---|
| ATN | 100 | 1 | Assign the inverse tangent of 1 (0.785398) to variable 100. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 100 | Assign 100 to variable 1. |
| LET | 101 | 1 | Assign 1 to variable 101. |
| ATN | *1 | *101 | Assign the inverse arc tangent of the content of variable 101, or 0.785398, to the variable that corresponds to the content of variable 1, or 100. |

## ● SQR (Root operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SQR | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Assign the root of the data specified in operand 2 to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

| | | | | |
|---|---|---|---|---|
| [Example 1] | SQR | 1 | 4 | Assign the root of 4 (2) to variable 1. |
| [Example 2] | LET | 1 | 10 | Assign 10 to variable 1. |
| | LET | 2 | 4 | Assign 4 to variable 2. |
| | SQR | *1 | *2 | Assign the square root of the content of variable 2, or 4, to the variable that corresponds to the content of variable 1, or 10. |

## ● DTOR (Angle conversion (Degrees to Radians))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DTOR | Variable number | Data | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.10 or later) |

[Function] The angle (unit: degrees) in operand 2 of the variable in operand 1 gets converted into radians.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.

[Example] DTOR 100 90 A number converted into radians from 90 degrees of an angle gets substituted to the variable 100.

## ● RTOD (Angle conversion (Radians to Degrees))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RTOD | Variable number | Data | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.10 or later) |

[Function] The angle (unit: radians) in operand 2 of the variable in operand 1 gets converted into degrees.
The setting in operand 1 must be a real variable in a range of 100 to 199, 1100 to 1199, 300 to 399 or 1300 to 1399.

[Example] RTOD 100 1.57079 A number converted into degrees from 1.57079 radians of an angle gets substituted to the variable 100.

## ● ABS (Absolute value calculation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ABS | Variable number | Data | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.10 or later) |

[Function] An absolute value in the data in operand 2 gets substituted to operand 1.

[Example] ABS 1 -3 The absolute value of -3 gets substituted to the variable 1.

## ● SGN (Get signs)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SGN | Variable number | Data | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.10 or later) |

[Function] Sign in operand 2 data gets searched and substituted in the variable in operand 1.
The values substituted in the variable in operand 1 are as follows.

| Value Substituted in Operand 1 Variable | Operand 2 Data |
|---|---|
| 1 | Positive Value |
| 0 | 0 |
| -1 | Negative Value |

[Example] SGN 1 -3 The sign of -3 gets substituted in the variable 1.

## [4] Logical Operation

### ● AND (Logical AND)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | AND | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Assign the logical AND operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

| | | | | |
|---|---|---|---|---|
| [Example 1] | LET | 1 | 204 | Assign 204 to variable 1. |
| | AND | 1 | 170 | Assign the logical AND operation result (136) of the content of variable 1 (204) and 170, to variable 1. |

| | | | | |
|---|---|---|---|---|
| [Example 2] | LET | 1 | 2 | Assign 2 to variable 1. |
| | LET | 2 | 204 | Assign 204 to variable 2. |
| | LET | 3 | 170 | Assign 170 to variable 3. |
| | AND | *1 | *3 | Assign the logical product 136 of the content 204 of the variable that corresponds to the content of variable 1, or 2, and the content of variable 3, or 170, to the variable that corresponds to the content of variable 1, or 2. |

| Decimal | Binary |
|---|---|
| 204 | 11001100 |
| AND 170 | AND 10101010 |
| 136 | 10001000 |

## ● OR (Logical OR)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OR | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Assign the logical OR operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

[Example 1]

```
LET   1    204    Assign 204 to variable 1.
OR    1    170    Assign the logical OR operation result (238) of the
                  content of variable 1 (204) and 170, to variable 1.
```

[Example 2]

```
LET   2    2      Assign 2 to variable 1.
LET   2    204    Assign 204 to variable 2.
LET   3    170    Assign 170 to variable 3.
OR    *1   *3     Assign the logical sum 238 of the content 204 of
                  the variable that corresponds to the content of
                  variable 1, or 2, and the content of variable 3, or
                  170, to the variable that corresponds to the content
                  of variable 1, or 2.
```

```
      Decimal             Binary

        204             11001100
   OR   170        OR   10101010
   ----------      ---------------
        238             11101110
```

## ● EOR (Logical exclusive-OR)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | EOR | Variable number | Data | ZR |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Assign the logical exclusive-OR operation result of the content of the variable specified in operand 1 and the value specified in operand 2, to the variable specified in operand 1.
The output will turn ON when the operation result becomes 0.

| | | | | |
|---|---|---|---|---|
| [Example 1] | LET | 1 | 204 | Assign 204 to variable 1. |
| | EOR | 1 | 170 | Assign the logical exclusive-OR operation result (102) of the content of variable 1 (204) and 170, to variable 1. |

| | | | | |
|---|---|---|---|---|
| [Example 2] | LET | 1 | 2 | Assign 2 to variable 1. |
| | LET | 2 | 204 | Assign 204 to variable 2. |
| | LET | 3 | 170 | Assign 170 to variable 3. |
| | EOR | *1 | *3 | Assign the exclusive logical sum 102 of the content 204 of the variable that corresponds to the content of variable 1, or 2, and the content of variable 3, or 170, to the variable that corresponds to the content of variable 1, or 2. |

| Decimal | Binary |
|---|---|
| 204 | 11001100 |
| EOR 170 | EOR 10101010 |
| 102 | 01100110 |

## ● NOT (Deny)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | NOT | Variable number | Data | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.12 or later) |

[Function] The variable number in Operation 1 should be substituted by one's complement (reversed of each bit) in the data in Operation 2.
The output turns on when the arithmetic result gets to 0.

[Example] NOT 1 1 1 (decimal number) of one's complement should be substituted to Variable 1

1 ···0x00000001 (hexadecimal number)

Each bit reversed

Variable 1 (decimal number)

—2 ···0xFFFFFFFE (hexadecimal number)

## ● LSFT (Logic Shifted to Left)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | LSFT | Variable number | Number of shifted bits | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.12 or later) |

[Function] Variable number in Operation 1 is to be shifted to left for the number of bits in Operation 2, and substituted to the variable in Operation 1.
The number of shifted bits is available to indicate from 0 to 31. 0 should be set to the low-order bit for the amount of shift.

[Example]

| | | | |
|---|---|---|---|
| LET | 1 | 1 | 1 (decimal number) substituted to Variable 1 |
| LSFT | 1 | 2 | Variable 1 shifted to left for two bits substituted to Variable 1 |

Variable 1(decimal number)

1 ･･･00000001 (binary number)

Shifted to left for two bits

4 ･･･00000100 (binary number)

## ● RSFT (Logic Shifted to Right)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | RSFT | Variable number | Number of shifted bits | ZR |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later)<br>MSEL (Main application V2.12 or later) |



[Function] Variable number in Operation 1 is to be shifted to right for the number of bits in Operation 2, and substituted to the variable in Operation 1.
The number of shifted bits is available to indicate from 0 to 31. 0 should be set to the high-order bit for the amount of shift.

[Example]

| | | | |
|---|---|---|---|
| LET | 1 | 16 | 16 (decimal number) substituted to Variable 1 |
| RSFT | 1 | 2 | Variable 1 shifted to right for two bits substituted to Variable 1 |

Variable 1(decimal number)

16 ···00010000 (binary number)

↓ Shifted to right for two bits

4 ···00000100 (binary number)

## [5] Comparison Operation

### ● CP□□ (Compare)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CP□□ | Variable number | Data | EQ NE<br>GT GE<br>LT LE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] The output will be turned ON if the comparison result of the content of the variable specified in operand 1 and the value specified in operand 2 satisfies the condition.
The value in the variable does not change.
The output will be turned OFF if the condition is not satisfied.

(Note) The output will not be turned OFF when the command is executed.

CP□□

| | | |
|---|---|---|
| EQ | ........... | Operand 1 = Operand 2 |
| NE | ........... | Operand 1 ≠ Operand 2 |
| GT | ........... | Operand 1 > Operand 2 |
| GE | ........... | Operand 1 ≥ Operand 2 |
| LT | ........... | Operand 1 < Operand 2 |
| LE | ........... | Operand 1 ≤ Operand 2 |

[Example 1]

| | | | | | |
|---|---|---|---|---|---|
| | LET | 1 | 10 | | Assign 10 to variable 1. |
| | CPEQ | 1 | 10 | 600 | Turn ON flag 600 if the content of variable 1 is 10. |
| 600 | ADD | 2 | 1 | | Add 1 to variable 2 if flag 600 is ON. |

[Example 2]

| | | | | |
|---|---|---|---|---|
| LET | 1 | 2 | | Assign 2 to variable 1. |
| LET | 2 | 10 | | Assign 10 to variable 2. |
| LET | 3 | 10 | | Assign 10 to variable 3. |
| CPNE | *1 | *3 | 310 | Turn ON output 310 if the variable that corresponds to the content of variable 1, or 2, is not equal to the content of variable 3. |

[6] Timer

## ● TIMW (Timer)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TIMW | Time | Prohibited | TU |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Stop the program and wait for the time specified in operand 1.
The setting range is 0.01 to 99, and the unit is second.
The output will turn ON when the specified time has elapsed and the program proceeds to the next step.

| | | | | |
|---|---|---|---|---|
| [Example 1] | TIMW | 1.5 | | Wait for 1.5sec. |
| [Example 2] | LET | 1 | 10 | Assign 10 to variable 1. |
| | TIMW | *1 | | Wait for the content of variable 1 (10sec). |

## ● TIMC (Cancel timer)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TIMC | Program number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Cancel a timer in other program running in parallel.

(Note) Timers in TIMW, WTON, WTOF and READ commands can be cancelled. In the case of WTON, WTOF and READ commands, even if timeout is not specified it is assumed that an unlimited timer has been specified and the wait time will be cancelled.

[Example 1]

```
TIMC  10          Cancel the wait time in program 10.
```

[Example 2]

```
LET   1    10     Assign 10 to variable 1.
TIMC  *1          Cancel the wait time in the content of variable 1
                  (program 10).
```

[Example 3]

```
Program 1     Program 10

   :             :
   :          WTON 8   20          Program 10 waits for input 8 for 20
   :          (Wait for input 8)   seconds.
TIMC  10      (Wait for input 8)   Cancel the wait time in program 10.
   :             :
```

(Note) The steps shown in the above example represent those executed simultaneously in different programs.

## ● GTTM (Get time)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GTTM | Variable number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Read system time to the variable specified in operand 1. The time is specified in units of 10msec.
The time obtained here has no base number. Therefore, this command is called twice and the difference will be used to calculate the elapsed time.

(Note) The system time is the time counted by 32 bits with the controller startup timing as 0. Therefore, the time passed can be defined by the time difference acquired while in the continuous operation for approximately 248 days (21474836.47 seconds) after a controller startup.

[Example 1]

| | | | |
|---|---|---|---|
| LET | 1 | 5 | Assign 5 to variable 1. |
| GTTM | *1 | | Store the current system time in the content of variable 1 (variable 5). |

## [7] I/O, Flag Operation

### ● BT□□ (Output port, flag operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | BT□□ | Output, flag | (Output, flag) | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Reverse the ON/OFF status of the output ports or flags from the one specified in operand 1 through the other specified in operand 2.

(Note) Dedicated outputs (system outputs) other than general-purpose outputs cannot be specified for operands 1 and 2.

BT□□
- ON ........... Switch the status to ON.
- OF ........... Switch the status to OFF.
- NT ........... Reverse the status.

| | | | | |
|---|---|---|---|---|
| [Example 1] | BTON | 300 | | Turn ON output port 300. |
| [Example 2] | BTOF | 300 | 307 | Turn OFF output ports 300 through 307. |
| [Example 3] | LET | 1 | 600 | Assign 600 to variable 1. |
| | BTNT | *1 | | Reverse the content of variable 1 (flag 600). |
| [Example 4] | LET | 1 | 600 | Assign 600 to variable 1. |
| | LET | 2 | 607 | Assign 607 to variable 2. |
| | BTON | *1 | *2 | Turn ON the flags from the content of variable 1 (flag 600) through the content of variable 2 (flag 607). |

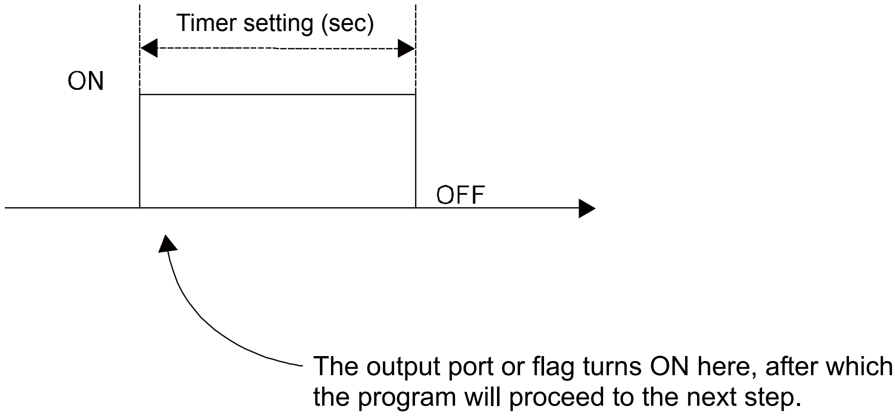
## ● BTPN (Output ON pulse)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | BTPN | Output port, flag | Timer setting | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Turn ON the specified output port or flag for the specified time.
When this command is executed, the output port or flag specified in operand 1 will be turned ON and then the program will proceed to the next step. The output port or flag will be turned OFF automatically upon elapse of the timer setting specified in operand 2.
The timer is set in a range from 0.01 to 99.00sec (including up to two decimal places).




(Note 1) If this command is executed with respect to an output port or flag already ON, the output port or flag will be turned OFF upon elapse of the timer setting.

(Note 2) If the program ends after the command has been executed but before the timer is up, the output port or flag will not be turned OFF.

(Note 3) This command will not be cancelled by a TIMC command.

(Note 4) A maximum of 16 timers, including BTPN and BTPF, can be operated simultaneously in a single program.
(There is no limitation as to how many times these timers can be used in a single program.)

(Note 5) Dedicated outputs (system outputs) other than general-purpose outputs cannot be specified for operand 1.

(Note 6) If other task interrupts after a port is turned ON until it is subsequently turned OFF, an error will generate in pulse output time, in which case pulse output cannot be used for a specified period.

[Example]

| | | | |
|---|---|---|---|
| BTPN | 300 | 1 | Turn ON output port 300 for 1sec. |
| BTPN | 600 | 10 | Turn ON flag 600 for 10sec. |

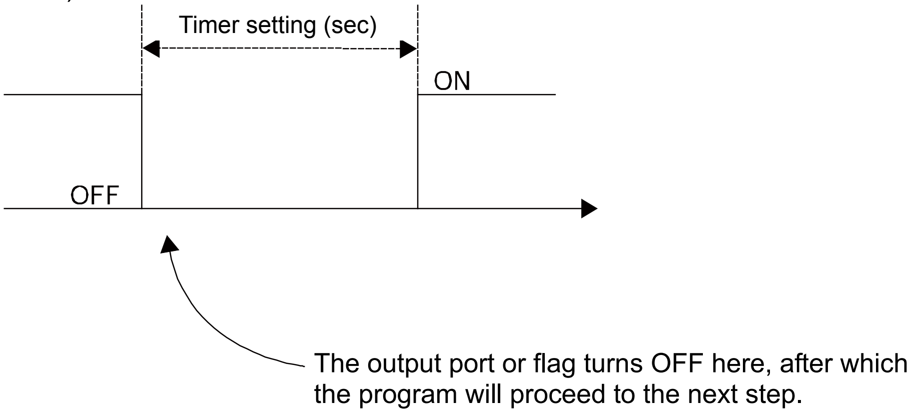
## ● BTPF (Output OFF pulse)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | BTPF | Output port, flag | Timer setting | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Turn OFF the specified output port or flag for the specified time.
When this command is executed, the output port or flag specified in operand 1 will be turned OFF and then the program will proceed to the next step. The output port or flag will be turned ON automatically upon elapse of the timer setting specified in operand 2.
The timer is set in a range from 0.01 to 99.00sec (including up to two decimal places).




(Note 1) If this command is executed with respect to an output port or flag already OFF, the output port or flag will be turned ON upon elapse of the timer setting.

(Note 2) If the program ends after the command has been executed but before the timer is up, the output port or flag will not be turned ON.

(Note 3) This command will not be cancelled by a TIMC command.

(Note 4) A maximum of 16 timers, including BTPN and BTPF, can be operated simultaneously in a single program.
(There is no limitation as to how many times these timers can be used in a single program.)

(Note 5) Dedicated outputs (system outputs) other than general-purpose outputs cannot be specified for operand 1.

(Note 6) If other task interrupts after a port is turned ON until it is subsequently turned OFF, an error will generate in pulse output time, in which case pulse output cannot be used for a specified period.

[Example]

| | | | |
|---|---|---|---|
| BTPF | 300 | 1 | Turn OFF output port 300 for 1sec. |
| BTPF | 600 | 10 | Turn OFF flag 600 for 10sec. |

## ● WT□□ (Wait for I/O port, flag)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | WT□□ | I/O, flag | (Time) | TU |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Wait for the I/O port or flag specified in operand 1 to turn ON/OFF.
The program can be aborted after the specified time by setting the time in operand 2.
The setting range is 0.01 to 99sec.
The output will turn ON upon elapse of the specified time (only when operand 2 is specified).

(Note) A local flag cannot be entered in operand 1.

```
WT□□
  ├ON ………… Wait for the applicable I/O port or flag to turn ON.
  └OF ………… Wait for the applicable I/O port or flag to turn OFF.
```

| | | | | |
|---|---|---|---|---|
| [Example 1] | WTON | 15 | | Wait for input port 15 to turn ON. |
| [Example 2] | WTOF | 308 | 10 | Wait for 10sec for output port 308 to turn OFF. |
| [Example 3] | LET | 1 | 600 | Assign 600 to variable 1. |
| | WTON | *1 | | Wait for the content of variable 1 (flag 600) to turn ON. |
| [Example 4] | LET | 1 | 8 | Assign 8 to variable 1. |
| | LET | 2 | 5 | Assign 5 to variable 2. |
| | WTOF | *1 | *2 | Wait for the content of variable 2 (5sec) for the content of variable 1 (input port 8) to turn OFF. |

## ● IN (Read I/O, flag as binary)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | IN | I/O, flag | I/O, flag | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Read the I/O ports or flags from the one specified in operand 1 through the other specified in operand 2, to variable 99 as a binary.

| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | • • • Binary |
|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | • • • Input port number |
| ON | OFF | OFF | OFF | OFF | ON | OFF | ON | |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | • • • Binary |

$$2^7 + 0 + 0 + 0 + 0 + 2^2 + 0 + 2^0$$

$$128 + 0 + 0 + 0 + 0 + 4 + 0 + 1 = 133$$

133 • • • • • • • • • • • • • • • • • • Variable 99

(Note 1) A maximum of 32 bits can be input.

(Note 2) When 32 bits have been input and the most significant bit is ON, the value read to variable 99 will be treated as a negative value.

(Note 3) The read data format can be changed using a FMIO command (refer to the section on FMIO command).

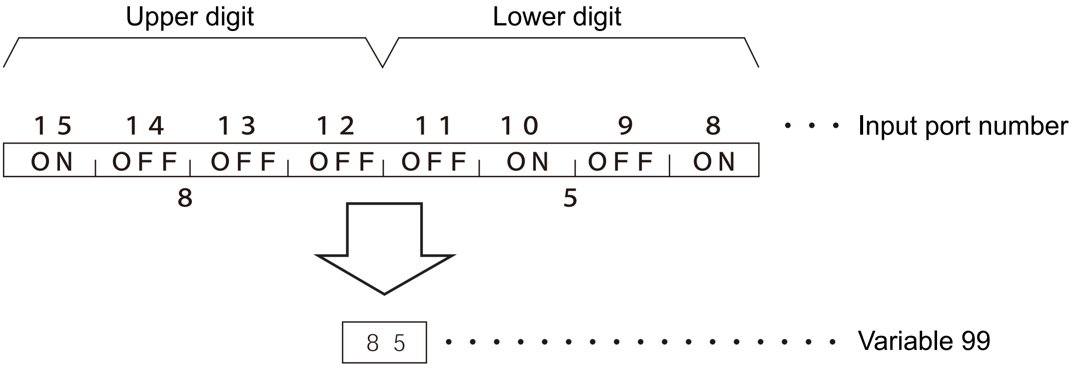
| | | | | |
|---|---|---|---|---|
| [Example 1] | IN | 8 | 15 | Read input ports 8 through 15, to variable 99 as a binary. |
| [Example 2] | LET | 1 | 8 | Assign 8 to variable 1. |
| | LET | 2 | 15 | Assign 15 to variable 2. |
| | IN | *1 | *2 | Read the input ports from the content of variable 1 (input port 8) through the content of variable 2 (input port 15), to variable 99 as a binary. |

## ● INB (Read I/O, flag as BCD)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | INB | I/O, flag | BCD digits | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Read the I/O ports or flags from the one specified in operand 1 for the number of digits specified in operand 2, to variable 99 as a BCD.




(Note 1) A maximum of eight digits (32 bits) can be input.

(Note 2) The number of I/O ports and flags that can be used is $4 \times n$ (digits).

(Note 3) The read data format can be changed using a FMIO command (refer to the section on FMIO command).

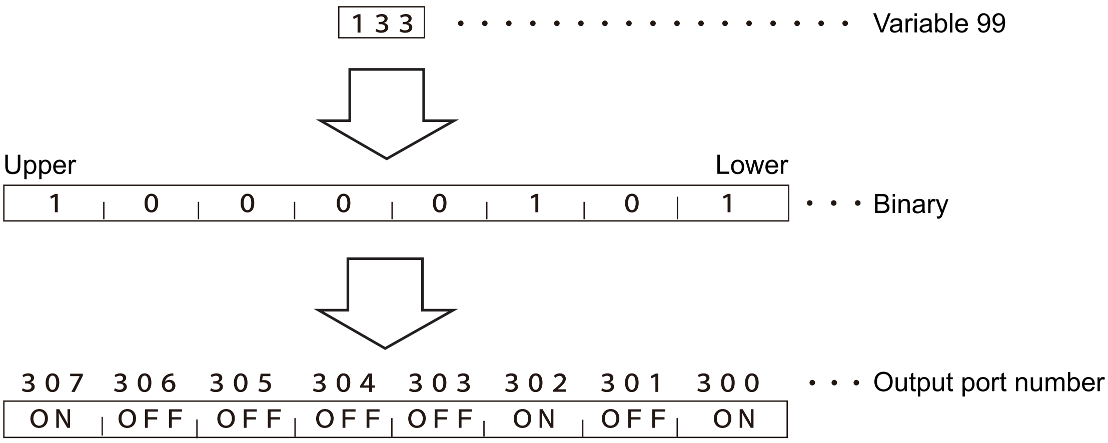
| | | | | |
|---|---|---|---|---|
| [Example 1] | INB | 8 | 2 | Read input ports 8 through 15, to variable 99 as a BCD. |
| [Example 2] | LET | 1 | 8 | Assign 8 to variable 1. |
| | LET | 2 | 2 | Assign 2 to variable 2. |
| | INB | *1 | *2 | Read the input ports from the content of variable 1 (input port 8) for the content of variable 2 (two digits) (until input port 15), to variable 99 as a BCD. |

## ● OUT (Write output, flag as binary)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OUT | Output, flag | Output, flag | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Write the value in variable 99 to the output ports or flags from the one specified in operand 1 through the other specified in operand 2.




(Note 1) A maximum of 32 bits can be output.

(Note 2) The write data format can be changed using a FMIO command (refer to the section on FMIO command).

| | | | | |
|---|---|---|---|---|
| [Example 1] | OUT | 300 | 307 | Write the value in variable 99 to output ports 300 through 307 as a binary. |
| [Example 2] | LET | 1 | 300 | Assign 300 to variable 1. |
| | LET | 2 | 307 | Assign 307 to variable 2. |
| | OUT | *1 | *2 | Write the value in variable 99 to the output ports from the content of variable 1 (output port 300) through the content of variable 2 (output port 307) as a binary. |

## ● OUTB (Write output, flag as BCD)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OUTB | Output, flag | BCD digits | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Write the value in variable 99 to the output ports or flags from the one specified in operand 1 for the number of digits specified in operand 2 as a BCD.

85 · · · · · · · · · · · · · · · · · · Variable 99

| 307 | 306 | 305 | 304 | 303 | 302 | 301 | 300 | · · · Output port number |
|---|---|---|---|---|---|---|---|---|
| ON | OFF | OFF | OFF | OFF | ON | OFF | ON | |

(Note 1) A maximum of eight digits (32 bits) can be output.

(Note 2) The number of output ports and flags that can be used is 4 × n (digits).

(Note 3) The write data format can be changed using a FMIO command (refer to the section on FMIO command).

[Example 1]

```
OUTB  300  2     Write the value in variable 99 to the output ports
                 from 300 for two digits (until output port 307) as a
                 BCD.
```

[Example 2]

```
LET   1    300   Assign 300 to variable 1.
LET   2    2     Assign 2 to variable 2.
OUTB  *1   *2    Write the value in variable 99 to the output ports
                 from the content of variable 1 (output port 300) for
                 the content of variable 2 (two digits) (until output
                 port 307) as a BCD.
```

## ● FMIO (Set IN, INB, OUT, OUTB, OTPS command format)

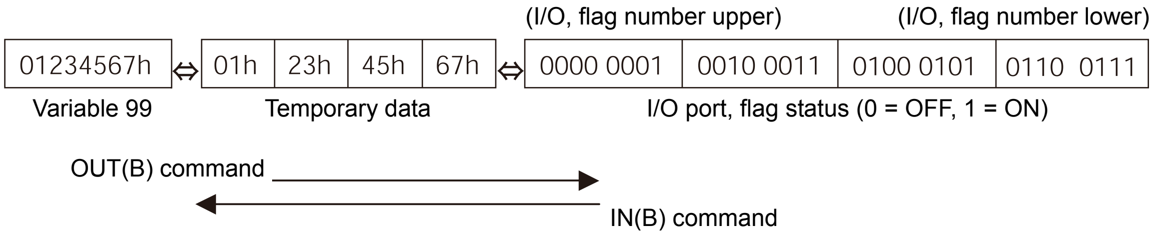
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | FMIO | Format type | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

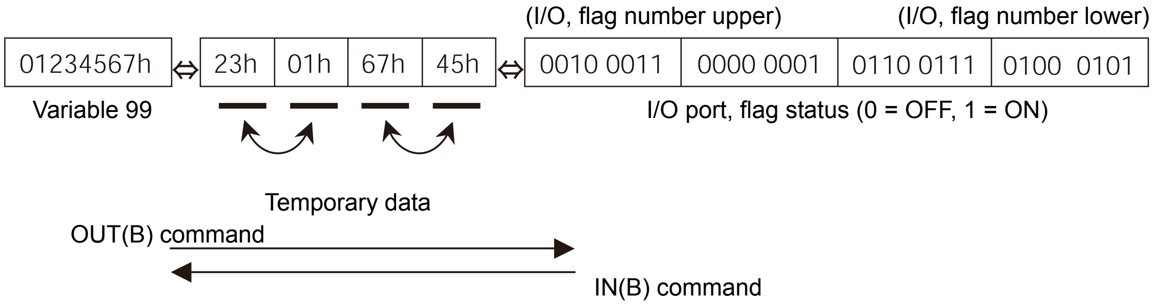


[Function] Set the data format for reading or writing I/O ports and flags with an IN, INB, OUT OUTB or OTPS command.
Details of data in each format type are shown for the IN, INB, OUT and OUTB commands. Data details of the OTPS command are the same as those of the OUT command, where the only difference is that variable 99 in the OUT command is replaced with current position data in the OTPS command.

1) Operand 1 = 0 (Default status when a FMIO command has not been executed)
   Data is read or written without being reversed.




2) Operand 1 = 1
   Data is read or written after its upper 8 bits and lower 8 bits are reversed every 16 bits.



3) Operand 1 = 2
Data is read or written after its upper 16 bits and lower 16 bits are reversed every 32 bits.

| Variable 99 | ⇔ | Temporary data | | | | ⇔ | (I/O, flag number upper) | | (I/O, flag number lower) | |
|---|---|---|---|---|---|---|---|---|---|---|
| 01234567h | ⇔ | 45h | 67h | 01h | 23h | ⇔ | 0100 0101 | 0110 0111 | 0000 0001 | 0010 0011 |

I/O port, flag status (0 = OFF, 1 = ON)

OUT(B) command →

← IN(B) command

4) Operand 1 = 3
Data is read or written after its upper 16 bits and lower 16 bits are reversed every 32 bits and its upper 8 bits and lower 8 bits are reversed every 16 bits.

| Variable 99 | ⇔ | Temporary data | | | | ⇔ | (I/O, flag number upper) | | (I/O, flag number lower) | |
|---|---|---|---|---|---|---|---|---|---|---|
| 01234567h | ⇔ | 67h | 45h | 23h | 01h | ⇔ | 0110 0111 | 0100 0101 | 0010 0011 | 0000 0001 |

I/O port, flag status (0 = OFF, 1 = ON)

OUT(B) command →

← IN(B) command

(Note) The FMIO command is supported by:
Main application Ver.0.56 or later
PC software Ver.2.0.45 or later
Teaching pendant :

| | |
|---|---|
| IA-T-X (D) | Ver.1.13 or later |
| SEL-T (D) | First edition or later |
| TB-01 (D) | First edition or later |
| TB-02 (D) | First edition or later |

[Example 1] Variable 99 = 00123456h (Decimal: 1193046, BCD: 123456)

00123456h ⇄ (OUT(B) command / IN(B) command) Variable 99 1193046 (IN/OUT command)
123456 (INB/OUTB command)

OUT(B) command ↓↑ IN(B) command

| | Temporary data | | | | ⇔ | (I/O, flag number upper) | | | (I/O, flag number lower) |
|---|---|---|---|---|---|---|---|---|---|
| FMIO=0 | 00h | 12h | 34h | 56h | ⇔ | 0000 0000 | 0001 0010 | 0011 0100 | 0101 0110 |
| FMIO=1 | 12h | 00h | 56h | 34h | ⇔ | 0001 0010 | 0000 0000 | 0101 0110 | 0011 0100 |
| FMIO=2 | 34h | 56h | 00h | 12h | ⇔ | 0011 0100 | 0101 0110 | 0000 0000 | 0001 0010 |
| FMIO=3 | 56h | 34h | 12h | 00h | ⇔ | 0101 0110 | 0011 0100 | 0001 0010 | 0000 0000 |

Temporary data ⇄ (OUT(B) command → / ← IN(B) command) I/O port, flag status (0 = OFF, 1 = ON)

[Example 2] Variable 99 = 00001234h (Decimal: 4660, BCD: 1234)

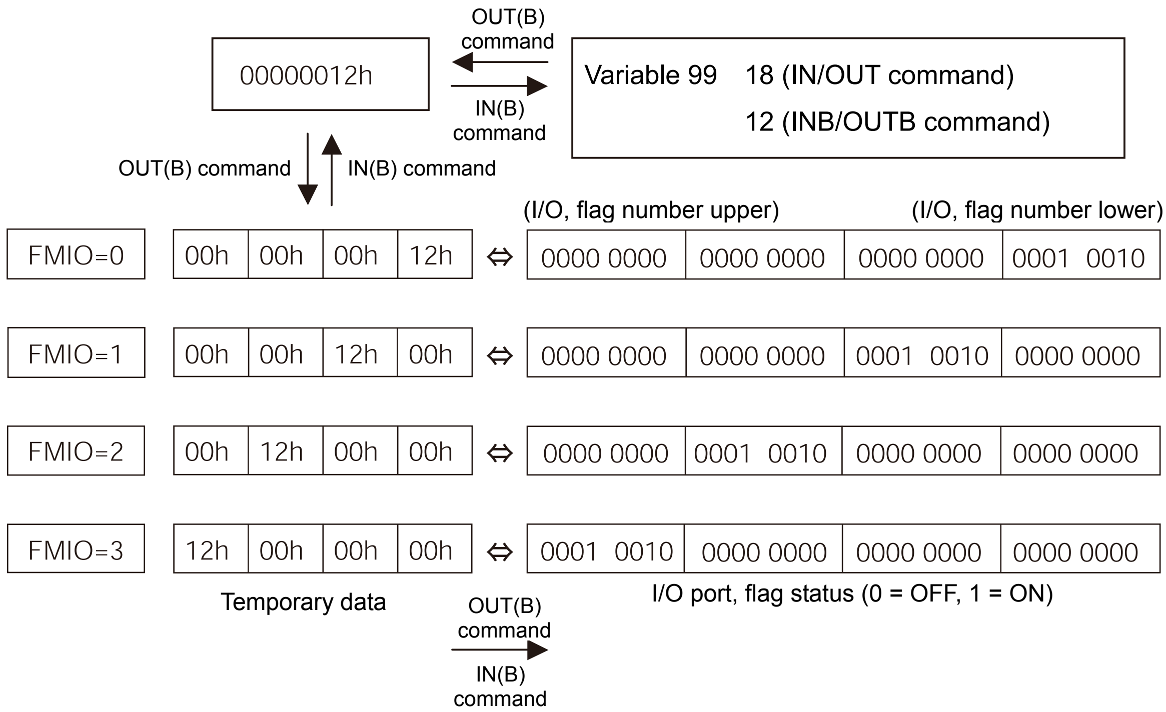
00001234h ⇄ (OUT(B) command / IN(B) command) Variable 99 4660 (IN/OUT command)
1234 (INB/OUTB command)

OUT(B) command ↓↑ IN(B) command

| | Temporary data | | | | ⇔ | (I/O, flag number upper) | | | (I/O, flag number lower) |
|---|---|---|---|---|---|---|---|---|---|
| FMIO=0 | 00h | 00h | 12h | 34h | ⇔ | 0000 0000 | 0000 0000 | 0001 0010 | 0011 0100 |
| FMIO=1 | 00h | 00h | 34h | 12h | ⇔ | 0000 0000 | 0000 0000 | 0011 0100 | 0001 0010 |
| FMIO=2 | 12h | 34h | 00h | 00h | ⇔ | 0001 0010 | 0011 0100 | 0000 0000 | 0000 0000 |
| FMIO=3 | 34h | 12h | 00h | 00h | ⇔ | 0011 0100 | 0001 0010 | 0000 0000 | 0000 0000 |

Temporary data ⇄ (OUT(B) command → / ← IN(B) command) I/O port, flag status (0 = OFF, 1 = ON)

[Example 3]  Variable 99 = 00000012h (Decimal: 18, BCD: 12)



## ● OTPS (Output current position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OTPS | Output port number | Axis number | CC |

* This command is supported by SSEL controller main application Ver.0.22 or later.

| Applicable models |
|---|
| SSEL ○<br>Other than SSEL × |



[Function]
- Output current position data to an output port.
- The current position data corresponding to the axis number specified in operand 2 is output to 32 bits of ports starting from the output port specified in operand 1.
- If the command is executed with 0 specified in operand 1, the command will become invalid and refreshing of current position data at the specified output will stop.
- When this command is executed, current position data will be refreshed continuously at the specified output port until the program in which this command was input is stopped or otherwise the command becomes invalid.
- 32 bits binary data (extended by sign) is output. The minimum unit is 0.001mm.

(Note)
- Only output ports of No. 300 or higher port numbers (multiples of 8) can be specified in operand 1.
- Only network output ports are supported.
- Even if this command is executed, output data remains indeterminable if home return is not yet completed.
- The output data format can be changed using the FMIO command (refer to the section on “FMIO command”). Note, however, that data is output in the FMIO-specified format when this command is called.

[Example 1]
When OTPS 300 1 is executed:
If the current position is -0.012mm, it is expressed as -12 (decimal) or FFFFFFF4 (binary) in units of 0.001mm.
Accordingly, FFFFFFF4 is output to output port No. 300 onward.

If the current position is 125.305mm, it is expressed as 125305 (decimal) or 0001E979 (binary) in units of 0.001mm.

Accordingly, 0001E979 is output to output port No. 300 onward.
The statuses of output ports are shown below.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 307 | 306 | 305 | 304 | 303 | 302 | 301 | 300 |
| OFF | ON | ON | ON | ON | OFF | OFF | ON |
| 315 | 314 | 313 | 312 | 311 | 310 | 309 | 308 |
| ON | ON | ON | OFF | ON | OFF | OFF | ON |
| 323 | 322 | 321 | 320 | 319 | 318 | 317 | 316 |
| OFF | OFF | OFF | OFF | OFF | OFF | OFF | ON |
| 331 | 330 | 329 | 328 | 327 | 326 | 325 | 324 |
| OFF | OFF | OFF | OFF | OFF | OFF | OFF | OFf |

## [8] Program Control

### ● GOTO (Jump)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GOTO | Tag number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Jump to the position of the tag number specified in operand 1.

(Note 1) A GOTO command is valid only within the same program.
(Note 2) Do not create a program that contains an infinite series of continuous movement commands using TAG-GOTO statements. Doing so will cause coordinate conversion errors to accumulate.

[Example 1]

```
TAG     1     Set a tag.
 :
 :
 :
GOTO    1     Jump to tag 1.
```

Using a GOTO command to branch out of or into any of the syntaxes listed below is prohibited.
Since the maximum number of nests is defined for each conditional branching command or subroutine call, a nest will be infinitely repeated if an ED☐☐ is not passed, and a nest (repetition) overflow error will generate. In the case of palletizing setting, an error will generate if the second BGPA is declared after the first BGPA declaration without passing an EDPA.

(1) IF☐☐ or IS☐☐ and EDIF syntax
(2) DWXX and EDDO syntax
(3) SLCT and EDSL syntax
(4) BGSR and EDSR syntax
(5) BGPA and EDPA syntax

## ● TAG (Declare tag)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | TAG | Tag number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the tag number specified in operand 1.

[Example 1] Refer to the section on GOTO command.

## ● EXSR (Execute subroutine)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | EXSR | Subroutine number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Execute the subroutine specified in operand 1.
A maximum of 15 nested subroutine calls are supported.

(Note) This command is valid only for subroutines within the same program.

[Example 1]

```
EXSR    1                   Execute subroutine 1.
  :
  :
EXIT
BGSR    1                   Start subroutine 1.
  :
  :
  :
EDSR                        End subroutine 1.
```

[Example 2]

```
LET     1       10          Assign 10 to variable 1.
EXSR    *1                  Execute the content of variable 1 (subroutine 10).
```

## ● BGSR (Start subroutine)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | BGSR | Subroutine number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Declare the start of the subroutine specified in operand 1.

[Example 1] Refer to the section on EXSR command.

(Note) Using a GOTO command to branch out of or into a BGSR-EDSR syntax is prohibited.

## ● EDSR (End subroutine)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | EDSR | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Declare the end of a subroutine.
This command is always required at the end of a subroutine.
Thereafter, the program will proceed to the step next to the EXSR that has been called.

[Example 1] Refer to the section on EXSR command.

## [9] Task Management

### ● EXIT (End program)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | EXIT | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] End the program.
If the last step has been reached without encountering any EXIT command, the program will return to the beginning.

(Note) Status at program end
- Output ports ············· Retained
- Local flags ··············· Cleared
- Local variables ········· Cleared
- Current values ·········· Retained
- Global flags·············· Retained
- Global variables ········ Retained

## ● EXPG (Start other program)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | EXPG | Program number | (Program number) | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Start the programs from the one specified in operand 1 through the other specified in operand 2, and run them in parallel. Specification in operand 1 only is allowed.

[Example 1] EXPG 10 12 Start program No. 10, 11 and 12.

Error-generation/output-operation conditions
When one EXPG program is specified (only operand 1 is specified)

| Status of the specified program | No program number error *1 | | | Program number error *1 |
|---|---|---|---|---|
| | Program already registered | | Program not yet registered | |
| | Program running | Program not running | | |
| Error | A57 "Multiple program start error" | None | C03 "Non-registered program specification error" | C2C "Program number error" |
| Output operation | ON | ON | OFF | OFF |

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 1… Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple EXPG programs are specified (both operands 1 and 2 are specified)

| Status of the specified program | No program number error *2 | | | Program number error *1 |
|---|---|---|---|---|
| | Registered program exists inside the specified range *3 | | None of programs inside the specified range are registered | |
| | Running program exists inside the specified range | None of programs inside the specified range are running | | |
| Error | A57 "Multiple program start error" | None | C03 "Non-registered program specification error" | C2C "Program number error" |
| Output operation | ON | ON | OFF | OFF |

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 2… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 3… In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.

## ● ABPG (Abort other program)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ABPG | Program number | (Program number) | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Abort other program.

(Note 1) If an ABPG command is issued while a movement command is being executed, the axes will immediately decelerate and stop.
(Note 2) Not only the operation but also the execution of the step itself will be terminated.

[Example 1] ABPG 10 12 End program No. 10, 11 and 12.

Error-generation/output-operation conditions
When one ABPG program is specified (only operand 1 is specified)

| Status of the specified program | No program number error *1 | | | Program number error *1 |
|---|---|---|---|---|
| | Program already registered | | Program not yet registered | |
| | Program running | Program not running | | |
| Error | None | None | None | C2C "Program number error" |
| Output operation | ON (OFF *2) | ON | OFF | OFF |

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 1… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 2… If an own task (own program) is specified in an ABPG command, the own task will be terminated and then deleted. The output will turn OFF.

When multiple ABPG programs are specified (both operands 1 and 2 are specified)

| Status of the specified program | No program number error *3 | | | Program number error *1 |
|---|---|---|---|---|
| | Registered program exists inside the specified range *4 | | None of programs inside the specified range are registered | |
| | Running program exists inside the specified range | None of programs inside the specified range are running | | |
| Error | None | None | None | C2C "Program number error" |
| Output operation | ON (OFF *5) | ON | OFF | OFF |

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 3… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 4… In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.

* 5… If an own task (own program) is included in the specified range, the own task will be terminated, upon which the processing of the ABPG command will end. Since the own task will be deleted, the result of ending the processing of specified programs will become indeterminable. Exercise caution. The output will always turn OFF regardless of the result.

## ● SSPG (Pause program)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SSPG | Program number | (Program number) | CC |

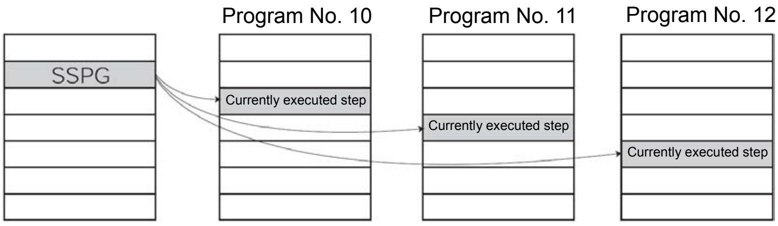
| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Pause the program from the one specified in operand 1 through the other specified in operand 2, at the current step. Specification in operand 1 only is allowed.

(Note 1) Pausing a program will also pause the operation the program has been executing.

(Note 2) Not only the operation but also the execution of the step itself will be paused.

[Example 1] SSPG 10 12 Pause program No. 10, 11 and 12 at the current step.




Error-generation/output-operation conditions

When one SSPG program is specified (only operand 1 is specified)

| Status of the specified program | No program number error *1 | | | Program number error *1 |
|---|---|---|---|---|
| | Program already registered | | Program not yet registered | |
| | Program running | Program not running | | |
| Error | None | None | C03 "Non-registered program specification error" | C2C "Program number error" |
| Output operation | ON | OFF | OFF | OFF |

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 1… Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple SSPG programs are specified (both operands 1 and 2 are specified)

| Status of the specified program | No program number error *2 | | | Program number error *1 |
|---|---|---|---|---|
| | Registered program exists inside the specified range *3 | | None of programs inside the specified range are registered | |
| | Running program exists inside the specified range *4 | None of programs inside the specified range are running | | |
| Error | None | None | C03 "Non-registered program specification error" | C2C "Program number error" |
| Output operation | ON | OFF | OFF | OFF |

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 2… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 3… In this case, non-registered programs inside the specified range are not treated as a target of operation with EXPG, ABPG, SSPG and RSPG commands. This will not affect error generation or output operation.

* 4… In this case, programs not running (but already registered) inside the specified range are not treated as a target of operation with SSPG and RSPG commands. This will not affect error generation or output operation.

## ● RSPG (Resume program)

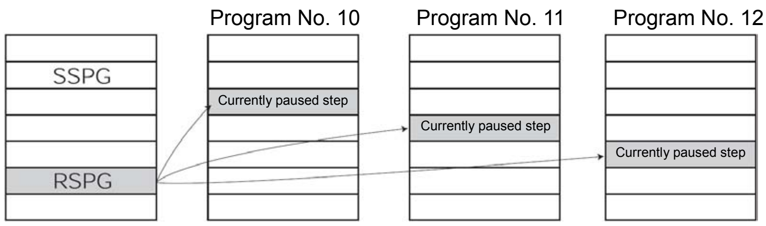
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RSPG | Program number | (Program number) | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Resume the programs from the one specified in operand 1 through the other specified in operand 2. Specification in operand 1 only is allowed.

(Note 1) Resuming a program will also resume the operation the program had been executing before the pause.

[Example 1] RSPG 10 12 Resume program No. 10, 11 and 12 from the paused step.




Error-generation/output-operation conditions

When one RSPG program is specified (only operand 1 is specified)

| Status of the specified program | No program number error *1 | | | Program number error *1 |
|---|---|---|---|---|
| | Program already registered | | Program not yet registered | |
| | Program running | Program not running | | |
| Error | None | None | C03 "Non-registered program specification error" | C2C "Program number error" |
| Output operation | ON | OFF | OFF | OFF |

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 1… Program number error indicates specification of a number smaller than 1 or exceeding 64.

When multiple RSPG programs are specified (both operands 1 and 2 are specified)

| Status of the specified program | No program number error *2 | | | Program number error *1 |
|---|---|---|---|---|
| | Registered program exists inside the specified range *3 | | None of programs inside the specified range are registered | |
| | Running program exists inside the specified range *4 | None of programs inside the specified range are running | | |
| Error | None | None | C03 "Non-registered program specification error" | C2C "Program number error" |
| Output operation | ON | OFF | OFF | OFF |

* The errors shown in the table represent those that generate in accordance with the status of the specified program. Errors caused by other factors are excluded.

* 2… Program number error indicates specification of a number smaller than 1 or exceeding 64.

* 3… In this case, non-registered programs inside the specified range are not treated as a target of operation. This will not affect error generation or output operation.

* 4… In this case, programs not running (but already registered) inside the specified range are not treated as a target of operation with SSPG and RSPG commands. This will not affect error generation or output operation.

## [10] Position Operation

### ● PGET (Read position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PGET | Axis number | Position number | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Read to variable 199 the data of the axis number specified in operand 1 in the position data specified in operand 2.
If the position data table contains no data to be loaded (= the position data display on the teaching pendant shows X.XXX, blank (for each model) or position data display fields in the PC software are blank) when the PGET command is executed, no data will placed in variable 199 (= the PGET command will not be executed).

[Example 1]

```
PGET    2     3      Read to variable 199 the data of Y-axis (axis 2) at
                     position 3.
```

[Example 2]

```
LET     1     2      Assign 2 to variable 1.
LET     2     3      Assign 3 to variable 2.
PGET    *1    *2     Read to variable 199 the data of the content Y-axis
                     (axis 2) of variable 1 at the content 3 of variable 2
                     at the position number.
```

## ● PPUT (Write position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PPUT | Axis number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Write the value in variable 199 to the axis number specified in operand 1 in the position data specified in operand 2.

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 150 | Assign 150 to variable 199. |
| PPUT | 2 | 3 | Write the content 150 of variable 199 to Y-axis (axis 2) at position 3. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 150 | Assign 150 to variable 199. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2 |
| PPUT | *1 | *2 | Write the content 150 of variable 199 to the content Y-axis (axis 2) of variable 1 at the content 3 of variable 2 at the position number. |

## ●PCLR (Clear position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PCLR | Position number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Clear the position data from the one specified in operand 1 through the other specified in operand 2.
When data is cleared, the field no longer contains data, which is different from the value of 0.000. The position data display on the teaching pendant changes to ×.××× or blank (for each model) while position data fields in the PC software become blank.

(Note 1) The comment on each position data are also subject to delete. If the position data with a comment is deleted by PCLR Command and software reset is conducted or the power is turned OFF without flash ROM writing being conducted, 22B “Position Data Comment Lost Error” will occur.

[Example 1]

```
PCLR   10   20     Clear the data from position No. 10 through 20.
```

[Example 2]

```
LET    1    10     Assign 10 to variable 1.
LET    2    20     Assign 20 to variable 2.
PCLR   *1   *2     Clear the data of the content of variable 1 (position
                   10) through the content of variable 2 (position 20).
```

## ● PCPY (Copy position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PCPY | Position number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Copy the position data specified in operand 2 to the position number specified in operand 1.

(Note 1) The comment on each position data are also subject to copy. If a change is made to the position data comment of the area to recover at by PCPY Command and software reset is conducted or the power is turned OFF without flash ROM writing being conducted, 22B "Position Data Comment Lost Error" will occur.

[Example 1]

```
PCPY    20    10      Copy the data of position No. 10 to position No. 20.
```

[Example 2]

```
LET     1     20      Assign 20 to variable 1.
LET     2     10      Assign 10 to variable 2.
PCPY    *1    *2      Copy the data of the content of variable 2 (position
                      10) to the content of variable 1 (position 20).
```

## ● PRED (Read current position)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PRED | Axis pattern | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Read the current position of the axis specified in operand 1 to the position specified in operand 2.

[Example 1] PRED 11 10 Read the current positions of X and Y-axis to position No. 10.

[Example 2] The axis pattern can be specified indirectly using a variable. When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    1    3     Assign 3 to variable 1.
PRED   *1   10
```

[Example 3]

```
LET    1    10    Assign 10 to variable 1.
PRED   11   *1    Read the current positions of X and Y-axis to the
                  content of variable 1 (position 10).
```

## ● PRDQ (Read current axis position (single-axis direct))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PRDQ | Axis number | Variable number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Read the current position of the axis number specified in operand 1 to the variable specified in operand 2.

[Example] PRDQ 2 100 Read the current position of Y-axis (axis) 2 to variable 100.

## ● PTAM (Substitution of target arm system data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTAM | Variable No. (Two variables used in a row) | Position number | CP |

| Applicable models |
|---|
| XSEL-RX/SX/RXD/SXD ○<br>XSEL-RAX/SAX/RAXD/SAXD ○<br>MSEL-PCX/PGX ○<br>Other than above × |

[Function] It writes the two types of arm system indications in a row from the variable number indicated in Operand 1 into the position data indicated in Operand 2.

| Arm System Indication Type | Operand 1 Variable Setting |
|---|---|
| Right Arm System Substitution | 1 |
| Left Arm System Substitution | -1 |
| Arm System Data Clear | 0 |

Variation No. n in Operand 1 is the target arm system indication of the 1st to 4th axes or 1st to 3rd axes, and variable No. n+1 is that of the 5th to 8th axes. For the type to connect one unit of SCARA, make sure to set 0 to the indicated variable No. n+1.

| Variable No. | Description | Setting Range | |
|---|---|---|---|
| | | 1 unit of SCARA connected | 2 units of SCARA connected |
| n | 1st to 4th axes (1st to 3rd axes) SCARA target arm system | -1, 0, 1 | -1, 0, 1 |
| n + 1 | 5th to 8th axes SCARA target arm system | 0<br>(Reserved by the system) | -1, 0, 1 |

[Example]

| | | | |
|---|---|---|---|
| LET | 20 | 1 | Set right arm system to 1st to 4th axes (1st to 3rd axes) |
| LET | 21 | 0 | Set 0 to 5th to 8th axes (system reservation as it is not connected) |
| PTAM | 20 | 10 | Write the arm system data stored in Variable No. 20 and 21 to Position No. 10. |

Variable No.20 | 1 | : Right arm system indicated in 1st to 4th axes (1st to 3rd axes)

Variable No.21 | 0 | : System reserved (not connected) in 5th to 8th axes

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Arm1-4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|---|
| 10( ) | 250.000 | 250.000 | 100.000 | 0.000 | Right | | | |

## ● PTST (Check position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTST | Axis pattern | Position number | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Check if valid data is contained in the axis pattern specified in operand 1 at the position number specified in operand 2.
The output turns ON when the data specified by the axis pattern is not entirely available (= the position data display on the teaching pendant is ×.×××, blank (for each model) or position data fields in the PC software are blank).
0 is treated as valid data.

[Example 1] PTST 11 10 300 Turn ON output 300 if there are no valid values of X and Y-axis at position 10.
Output 300 will turn OFF if the position data is given as follows:

[Example 2] The axis pattern can be specified indirectly using a variable. When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET   1    3          Assign 3 to variable 1.
PTST  *1   10   300
```

[Example 3]

```
LET   1     11        Assign 11 to variable 1.
PTST  1011  *1   600  Turn ON flag 600 if there are no valid values
                      in the data of X, Y and R-axis at the content of
                      variable 1 (position 11).
                      Flag 600 will turn ON if the position data is
                      given as follows:
```

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 9( ) | | | | | | | |
| 10( ) | 200.000 | 100.000 | | | | | |
| 11( ) | | | 150.000 | | | | |
| 12( ) | | | | | | | |

## ● PVEL (Assign speed data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PVEL | Speed | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Write the CP operation speed/linear axis speed specified in operand 1 to the position number specified in operand 2. The unit of operand 1 is [mm/sec].

(Note 1) If a negative value is written in PVEL Command, an alarm will be generated when this position is indicated in a movement.

(Note 2) If 0 is set in Operation 1, the speed setting in the indicated position number is deleted.

[Example 1]

| | | | |
|---|---|---|---|
| PVEL | 100 | 10 | Write speed 100mm/s to position No. 10. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 100 | Assign 100 to variable 1. |
| LET | 2 | 10 | Assign 10 to variable 2. |
| PVEL | *1 | *2 | Write the content of variable 1 (speed 100mm/s) to the content of variable 2 (position 10). |

## ● PACC (Assign acceleration data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PACC | Acceleration | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Write the acceleration in CP operation/acceleration in linear axis operation specified in operand 1 to the position number specified in operand 2. The unit of operand 1 is [G] and the entered value is valid to the second decimal point.

[Example 1]

| | | | |
|---|---|---|---|
| PACC | 0.3 | 10 | Write acceleration 0.3G to position No. 10. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 100 | 0.3 | Assign 0.3 to variable 100. |
| LET | 2 | 10 | Assign 10 to variable 2. |
| PACC | *100 | *2 | Write the content of variable 100 (acceleration 0.3G) to the content of variable 2 (position 10). |

(Note 1) Range check is not performed for a PACC command.
(Note 2) If Operation 1 is set to 0, the acceleration setting on the indicated position number gets deleted.

## ● PDCL (Assign deceleration data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PDCL | Deceleration | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Write the CP operation deceleration/linear axis deceleration specified in operand 1, into the position number specified in operand 2.
The unit of operand 1 is [G], and the set value is effective to two decimal points.

[Example 1] PDCL 0.3 3 Assign 0.3 to the deceleration data at position No. 3.

(Note 1) If Operation 1 is set to 0, the deceleration setting on the indicated position number gets deleted.

## ● PAXS (Read axis pattern)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PAXS | Variable number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Store the axis pattern at the position specified in operand 2 to the variable specified in operand 1.

| | | | | |
|---|---|---|---|---|
| [Example 1] | PAXS | 1 | 98 | Read the axis pattern at position 98 to variable 1. If the position is given as follows, “3” (binary 0011) will be read to variable 1. |
| [Example 2] | LET | 1 | 3 | Assign 3 to variable 1. |
| | LET | 2 | 101 | Assign 101 to variable 2. |
| | PAXS | *1 | *2 | Read the axis pattern at the content of variable 2 (position 101) to the content of variable 1 (variable 3). If the point is given as follows, “8” (binary 1000) will be stored in variable 3. |

The table below shows different positions and corresponding values stored in a variable.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | |
|---|---|---|---|---|---|
| 98( ) | 200.000 | 100.000 | | | 0 0 1 1 = 2 + 1 = 3 |
| 99( ) | 350.000 | | 120.000 | | 0 1 0 1 = 4 + 1 = 5 |
| 100( ) | | | | | 0 |
| 101( ) | | | | 180.000 | 1 0 0 0 = 8 |

## ● PSIZ (Check position data size)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PSIZ | Variable number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Set an appropriate value in the variable specified in operand 1 in accordance with the parameter setting.

- When “Other parameter No. 23, PSIZ function type” = 0
  The maximum number of position data that can be stored in the controller will be set.
  (Regardless of whether the data are used or not.)
- When “Other parameter No. 23, PSIZ function type” = 1
  The number of position data used will be set.

[Example]

```
PSIZ      1
```

When “Other parameter No. 23, PSIZ function type” = 0
The maximum number of position data that can be stored in variable 1 will be set.
When “Other parameter No. 23, PSIZ function type” = 1
The number of position data currently used will be set in variable 1.

## ● GTAM (Acquirement of target arm system data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GTAM | Variable No. (Two variables used in a row) | Position number | CP |

| Applicable models |
|---|
| XSEL-RX/SX/RXD/SXD ○ |
| XSEL-RAX/SAX/RAXD/SAXD ○ |
| MSEL-PCX/PGX ○ |
| Other than above × |

[Function] Acquire the target arm system indication from the position data indicated in Operand 2, and set it in the two variables in a row indicated in Operand 1.

| Arm System Indication Type | Operand 1 Variable Setting |
|---|---|
| Right Arm System Substitution | 1 |
| Left Arm System Substitution | -1 |
| Not to be indicated | 0 |

Variation No. n in Operand 1 is the target arm system indication of the 1st to 4th axes or 1st to 3rd axes, and variable No. n+1 is that of the 5th to 8th axes. For the type to connect one unit of SCARA, make sure to set 0 to the indicated variable No. n+1.

| Variable No. | Description | Output Range | |
|---|---|---|---|
| | | 1 unit of SCARA connected | 2 units of SCARA connected |
| n | 1st to 4th axes (1st to 3rd axes) SCARA target arm system | -1, 0, 1 | -1, 0, 1 |
| n + 1 | 5th to 8th axes SCARA target arm system | Indefinite | -1, 0, 1 |

[Example] GTAM 20 10 Set the arm system data in Position No. 10 to Variable No. 20.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Arm1-4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|---|
| 10( ) | 250.000 | 250.000 | 100.000 | 0.000 | Right | | | |

Variable No.20 | 1 | : 1st to 4th axes (1st to 3rd axes) arm system

Variable No.21 | 0 | : 5th to 8th axes arm system (not connected)

## ● GVEL (Get speed data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GVEL | Variable number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Obtain speed data from the speed item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GVEL 100 10 Set the speed data at position No. 10 in variable 100.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 9( ) | | | | | | | |
| 10( ) | 250.000 | 100.000 | 100.000 | 30.000 | 100 | 0.80 | 0.80 |
| 11( ) | | | | | | | |

If the position data is set as above when the command is executed, 100 will be set in variable 100.

## ● GACC (Get acceleration data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GACC | Variable number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Obtain acceleration data from the acceleration item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GACC 100 10 Set the acceleration data at position No. 10 in variable 100.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 9( ) | | | | | | | |
| 10( ) | 250.000 | 100.000 | 100.000 | 30.000 | 100 | 0.80 | 0.80 |
| 11( ) | | | | | | | |

If the position data is set as above when the command is executed, 0.8 will be set in variable 100.

## ● GDCL (Get deceleration data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GDCL | Variable number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Obtain deceleration data from the deceleration item in the position data specified in operand 2, and set the value in the variable specified in operand 1.

[Example] GDCL 100 10 Set the deceleration data at position No. 10 in variable 100.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 9( ) | | | | | | | |
| 10( ) | 250.000 | 100.000 | 100.000 | 30.000 | 100 | 0.80 | 0.80 |
| 11( ) | | | | | | | |

If the position data is set as above when the command is executed, 0.8 will be set in variable 100.

## [11] Actuator Control Declaration

### ● VEL (Set speed)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | VEL | Speed | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the actuator travel speed in the value specified in operand 1.
In the case of a SCARA, set the operating speed for CP operation.
The unit is [mm/sec].
The maximum speed will vary depending on the model of the actuator connected.
Set a speed not exceeding the applicable maximum speed.

(Note 1) Decimal places cannot be used. An error will generate
(Note 2) The minimum speed is 1mm/sec.

[Example 1]

| | | |
|---|---|---|
| VEL | 100 | Set the speed to 100mm/sec. |
| MOVL | 1 | Move to point 1 at 100mm/sec. |

[Example 2]

| | | |
|---|---|---|
| VEL | 500 | Set the speed to 500mm/sec. |
| MOVL | 2 | Move to point 2 at 500mm/sec. |

## ● VELS (Dedicated SCARA command/Set speed ratio)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | VELS | Ratio | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/ PGX only) |

[Function] Set in operand 1 the moving speed for SCARA PTP operation command (angular speed for all axes other than Z) as a ratio of the maximum PTP speed. Operand 1 must be set with an integer (unit: %).

(Note 1) If a RIGH or LEFT command is used, the speed must be set with VELS even when a SCARA PTP operation command is not used.

[Example 1]

| | | |
|---|---|---|
| VELS | 50 | Set the moving speed for PTP operation command to 50% of the maximum value. |
| MOVP | 1 | Move to position No. 1 via PTP at 50% of the maximum speed. |

## ● OVRD (Override)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OVRD | Speed ratio | (OVRD type) | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models]<br>However, (OVRD Type) in Operand 2 is available to set only for XSEL-RA/SA, MSEL Controllers (PC, PG only) and TTA |
| ○ |

[Function] Change the speed in accordance with the ratio specified in operand 1 (speed coefficient setting). The speed ratio is set in a range from 1 to 100%.
However, XSEL-RA/SA, MSEL Controller, and TTA is 1 to 150%.
A speed command specifying a speed below 1mm/sec can be generated using OVRD.
Speed-command smoothing limit speed : 1pulse/msec
Speed-command generation limit speed : 1pulse/256msec
(Smoothing in actual operation is not guaranteed. It must be confirmed with the actual equipment.)
1pulse: Lead [mm]/16384 … Standard model with 1 : 1 gear ratio
(The speed set by a PAPR command (push-motion approach speed) is clamped at the lower-limit speed 1mm/sec.)

Operand 2: OVRD type
- Operand 2 = 0 or No indication
  Track concentration invalid (No adjustment Acceleration/Decceleration)
- Operand 2 = 1
  Acceleration and deceleration get adjusted automatically in response to the velocity ratio in the continuous operation system interpolation command in order to minimize the variance from the command track at 100% of overriding.

(Note 1) OVRD Type in Operand 2 (Operand 2) is available to set only for XSEL-RA/SA, MSEL Controllers and TTA.
(Note 2) The continuous operation system interpolation commands that the track concentration is valid are as shown below;
PATH, CIR, ARC, PSPL, CIR2, ARC2, ARCD, ARCC, CIRS, ARCS
(Note 3) The command track may vary due to the following causes when the track concentration is valid.
- Influence of the servo system parameter setting (such as gain)
- When values above the actuator specification are set in the acceleration and deceleration settings
  It may cause generation of an error, malfunction or shorten product life as well as variance in command track.
- When command velocity was changed by CHVL (velocity change) command
- When command velocity has varied by safety velocity valid / invalid

- Operation 1 (1 to 150%) and Operation 2 (OVRD Type) in MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-02(D) : first edition or later, TB-01(D) of Ver. 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D))
- It is applicable for Operation 1 (1 to 150%) and Operation 2 (OVED) of XSEL-RA/SA in the following versions and later.
  PC software 13.00.00.00, TB-02 1.30, TB-01 1.60
  Not applicable for SEL-T (D) and IA-T-X (D).

| | | | |
|---|---|---|---|
| [Example 1] | VEL | 100 | Set the speed to 100mm/s. |
| | OVRD | 50 | Reduce the speed to 50%. As a result, the actual speed will become 50mm/s. |

[Example 2: SCARA robot]

| | | | |
|---|---|---|---|
| VEL | 150 | Set the SCARA CP operation speed / linear axis speed to 150mm/sec. |
| VELS | 90 | Set the SCARA PTP operation speed ratio to 90%. |
| OVRD | 50 | Lower the speed to 50%. The SCARA CP operation speed/linear axis speed becomes 75mm/sec, while the SCARA PTP operation speed ratio becomes 45%. |

Command limit speed for smooth operation:
Travel distance per encoder pulse [mm/pulse] / time [msec]
Command limit speed that can be generated:
Travel distance per encoder pulse [mm/pulse] / time [msec]
(Smoothness of actual operation cannot be guaranteed. Movement must be checked on the actual machine.)

[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse] = (Screw lead [0.001mm] × Gear ratio numerator)
/ (Encoder resolution [pulses/rev] × Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse] = Encoder resolution (0.001μm/pulse) × 1000
/ (2 ^ Encoder division ratio)

(Reference) Use the values of the following parameters for the above calculation formulas:

| | |
|---|---|
| Encoder resolution: | Axis-specific parameter No. 42 |
| Encoder division ratio: | Axis-specific parameter No. 43 |
| Screw lead: | Axis-specific parameter No. 47 |
| Gear ratio numerator: | Axis-specific parameter No. 50 |
| Gear ratio denominator: | Axis-specific parameter No. 51 |

Example for setting of OVRD type available for setting only in XSEL-RA/SA, MSEL Controllers and TTA is shown in [Example 3].

[Example 3]

| | | | |
|---|---|---|---|
| VEL | 200 | | Set the velocity to 200mm/s. |
| OVRD | 80 | 1 | Set the velocity ratio to 80% (velocity should be 160mm/s) (track concentration). |
| PATH | 10 | 12 | Move Potion No. 10 to 12 continuously⋯(1) |
| : | | | |
| OVRD | 110 | 1 | Set the velocity ratio to 110% (velocity should be 220mm/s) (track concentration). |
| PATH | 10 | 12 | Move Potion No. 10 to 12 continuously⋯(2) |
| : | | | |
| VEL | 150 | | Set the velocity to 150mm/s. |
| OVRD | 100 | 1 | Set the velocity ratio to 100% (velocity should be 150mm/s) (track concentration). |
| PATH | 10 | 12 | Move Potion No. 10 to 12 continuously⋯(3) |

It should be the equivalent directive tracks for (1) and (2) (as the standard velocity is the same 200mm/s).
It should be different directive tracks for (1) and (3) (as the standard velocity is different as 200mm/s for (1) while 150mm/s for (3)).

## ● ACC (Set acceleration)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ACC | Acceleration | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the acceleration for actuator operation in operand 1.
For SCARA robot, the setting is the operational acceleration speed for CP operation.
The unit of operand 1 is [G], and the set value is effective to two decimal points.

(Note) [Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD/RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX]
If no acceleration is set in the position data or by an ACC command when the actuator moves, the actuator uses the default value registered in all-axis parameter No. 11, “Default acceleration”.
[XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD/RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX]
If no acceleration is set in the position data or by an ACC command during CP operation, a SCARA robot uses the default value registered in all-axis parameter No. 11, “Default CP acceleration for SCARA axis”, while a linear axis uses the default value registered in all-axis parameter No. 200, “Default acceleration for linear axis”.

[Example 1] ACC 0.3 Set the acceleration to 0.3G.

(Note) Setting an acceleration exceeding the specified range for the actuator may generate an error. It may also result in a failure or shorter product life.

## ● ACCS (Dedicated SCARA command/Set acceleration ratio)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ACCS | Ratio | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/ PGX only) |

[Function] Set in operand 1 the acceleration for movement by SCARA PTP operation command (angular acceleration for all axes other than Z) as a ratio of the maximum PTP acceleration. Operand 1 must be set with an integer (unit: %).

(Note 1) For the acceleration ratio setting, make sure to refer to “Caution for Use” in Vertical Articulated Robot IX Series Instruction Manual provided separately.

[Example] ACCS 50 Set the acceleration for movement by PTP operation command to 50% of the maximum value.

## ● DCL (Set deceleration)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DCL | Deceleration | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Set the deceleration for actuator operation in operand 1.
For SCARA robot, the setting is the operational deceleration speed for CP operation.
The unit of operand 1 is [G], and the set value is effective to two decimal points.

(Note) [Other than XSEL- JX/KX/PX/QX/RX/SX/RXD/SXD/RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX]
If the position data contains no deceleration AND deceleration is not set by a DCL command, the actuator will move based on the default value set in “All-axis parameter No. 12, Default deceleration”.
A DCL command cannot be used with CIR and ARC commands.
[XSEL- JX/KX/PX/QX/RX/SX/RXD/SXD/RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX]
If no deceleration is set in the position data or by a DCL command during CP operation, a SCARA robot uses the default value registered in all-axis parameter No. 12, “Default CP deceleration for SCARA”, while a linear axis uses the default value registered in all-axis parameter No. 201, “Default acceleration for linear axis”.
DCL is invalid with respect to a CIR or ARC command.

[Example] DCL 0.3 Set the deceleration to 0.3G.

(Note) Setting a deceleration exceeding the specified range for the actuator may generate an error. It may also result in a failure or shorter product life.

## ● DCLS (Dedicated SCARA command/Set deceleration ratio)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DCLS | Ratio | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/ PGX only) |

[Function] Set in operand 1 the deceleration for movement by SCARA PTP operation command (angular deceleration for all axes other than Z) as a ratio of the maximum PTP deceleration. Operand 1 must be set with an integer (unit: %).

(Note 1) For the deceleration ratio setting, make sure to refer to “Caution for Use” in Vertical Articulated Robot IX Series Instruction Manual provided separately.

[Example] DCLS 50 Set the acceleration for movement by PTP operation command to 50% of the maximum value.

## ● SCRV (Set sigmoid motion ratio) • • • 1/3

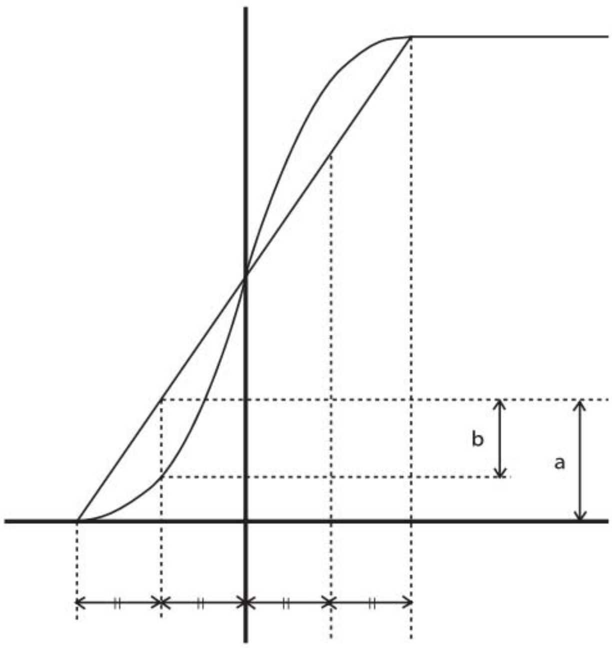
| Extension Condition (LD, A, O, AB, OB) | Input condition (I/O flag) | Command,declaration | | | Output (Output,flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand1 | Operand2 | |
| E | N, Cnd | Cmnd | Operand1 | Operand2 | Pst |
| Optional | Optional | SCRV | Ratio | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| ○ | × | × | × | ○ | × | ○ | ○ | ○ | TT:○, TTA:× | × |

[Function] Set the ratio of sigmoid motion control of the actuator in the value specified in operand1.
The ratio is set as integer in a range from 0 to 50(%)

$$\frac{b}{a} \times 100\ (\%)$$

If the ratio is not set using this command or 0% is set,a trapezoid motion will be implemented.
A SCRV command can be used with the following commands :
MOVP, MOVL, MVPI, MVLI, JBWF, JBWN, JFWF, JFWN, TMPI, TMLI, RIGH, LEFT




[Example] SCRV 30 Set the sigmoid motion ratio to 30%

## ● SCRV (Set sigmoid motion ratio) • • • 2/3

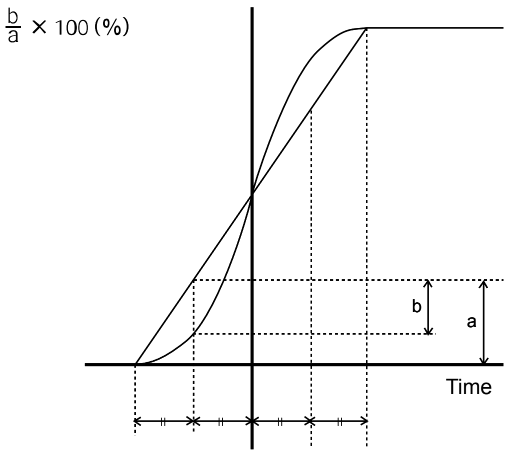
| Extension Condition (LD, A, O, AB, OB) | Input condition (I/O flag) | Command,declaration | | | Output (Output,flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand1 | Operand2 | |
| E | N, Cnd | Cmnd | Operand1 | Operand2 | Pst |
| Optional | Optional | SCRV | Ratio | (S-motion type) | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | ○ | × | × | ○ | ○ | × | × | TT: ×, TTA: ○ | ○ |

[Function] Set the ratio of sigmoid motion control of the actuator in the value specified in operand1.
The ratio is set as integer in a range from 0 to 50(%)
If the ratio is not set using this command or 0% is set, a trapezoid motion will be implemented.
A SCRV command can be used with the following commands：
MOVP, MOVL, MVPI, MVLI, JBWF, JBWN, JFWF, JFWN, TMPI, TMLI, RIGH, LEFT

| Value set in operand2 | Description |
|---|---|
| 0 or no specification | S-motionA |
| 1 | S-motionB (Recommended) |

- S-motion A (Operand 2 = Not specified or 0)




- S-motion B (Operand 2 = 1)
  If S-motionB is selected,the speed pattern becomes smoother(than the equivalent S-motion control ratio based on S-motionA).(The divergence peak relative to trapezoid motion becase smaller).

[Example] SCRV 30 Set the sigmoid motion ratio to 30%

## ● SCRV (Set sigmoid motion ratio) • • • 3/3

| Extension Condition (LD, A, O, AB, OB) | Input condition (I/O flag) | Command,declaration | | | Output (Output,flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand1 | Operand2 | |
| E | N, Cnd | Cmnd | Operand1 | Operand2 | Pst |
| Optional | Optional | SCRV | Ratio | (S-motion type) | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | × | ○ | × | × | × | ○ | × | × | × |

[Function] Set the ratio of sigmoid motion control of the actuator in the value specified in operand1.
The ratio is set as integer in a range from 0 to 50(%)
If the ratio is not set using this command or 0% is set,a trapezoid motion will be implemented.

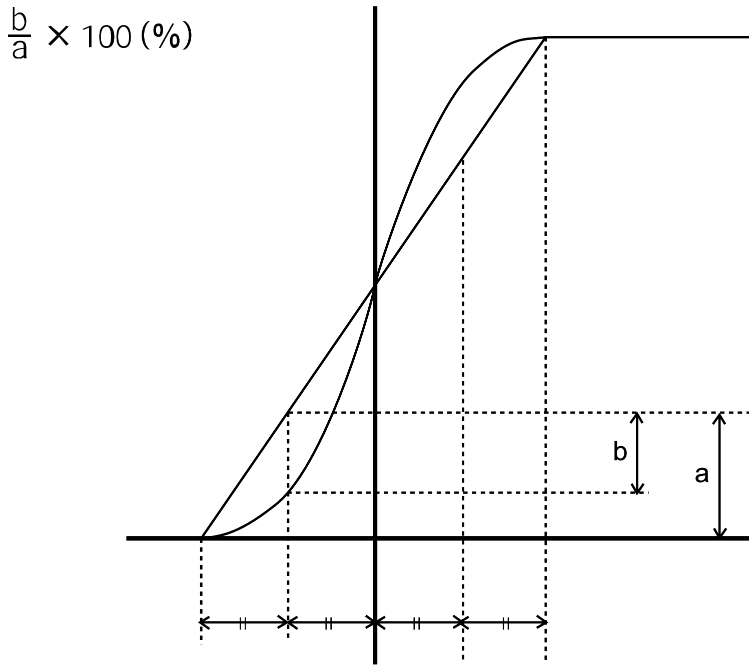
XSEL-P/Q/PCT/QCT are available to select operand2.(S-shaped type).
(XSEL-P/Q/PCT/QCT Main application Ver.1.25 and later)
Operand 2 can be inputted from IA-T-X(D):Ver.1.52 or subsequent ones later teaching box TB-02(D): first edition or later, TB-01(D): first edition later , SEL-T(D):Ver.1.12 later PC software:Ver.7.7.12.0 later.

| Model Name | Setting in Operand 2 (S-shaped Type) | S-shaped Motion Class | S-shaped Motion Effective Command Group (See the table below) |
|---|---|---|---|
| XSEL -P/Q XSEL-RA/ SA/RAX/ SAX/RAXD /SAXD | Not set, 0 | A | 1) |
| | 1 | B | 1) |
| | 2 | A | 2) (Note 2) |
| | 3 | B | 2) (Note 2) |
| XSEL -PCT/QCT | Not set, 0 | B (Note 1) | 1) |
| | 1 | | 1) |
| | 2 | | 2) (Note 2) |
| | 3 | | 2) (Note 2) |

Note 1 The class of S-shaped motion is compulsorily B.
Note 2 S-shaped Motion is effective also at the speed change point (position joint point) during PATH Command. If S-shaped Motion is activated, constant velocity or track could be lost. Use the unit with S-shaped Motion ineffective in such processes as applying paint or glue, in which the constant velocity and track are important.

| Effective Command Group | SCRV Effective Command |
|---|---|
| 1) | MOVP, MOVL, MVPI, MVLI, JBWF, JBWN, JFWF, JFWN |
| 2) | MOVP, MOVL, MVPI, MVLI, JBWF, JBWN, JFWF, JFWN, PATH, ARCH, PACH |

- S-motion A




- S-motion B
  In this class, operates with a speed pattern smoother than the control of S-shaped Motion Class A. (Estrangement peak with Trapezoid Motion becomes small.)

| | | | |
|---|---|---|---|
| [Example]SCRV | 30 | 1 | Set S-shaped motion ratio 30% and S-shaped motion class A. |

## ● OFST (Set offset)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OFST | Axis pattern | Offset value | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Add the offset in operand 2 to the target value for the axis pattern specified in operand 1 when the actuator moves, to reset the target value and operate the actuator accordingly.
The offset is set in mm, and the effective resolution is 0.001mm.
A negative offset may be specified as long as the operation range is not exceeded.

(Note) An OFST command cannot be used outside the applicable program. To use OFST in multiple programs, the command must be executed in each program.
An OFST command cannot be used with MVPI, MVTI, TMLI and TMPI commands.

[Example 1]

| | | | |
|---|---|---|---|
| OFST | 110 | 50 | Add 50mm to the specified positions of Y-axis and Z-axis. |

[Example 2] The axis pattern can be specified indirectly using a variable. When the command in [Example 1] is rephrased based on indirect specification using a variable:
110 (binary) → 6 (decimal)

| | | | |
|---|---|---|---|
| LET | 1 | 6 | Assign 6 to variable 1. |
| OFST | *1 | 50 | |

[Example 3]

| | | | |
|---|---|---|---|
| LET | 1 | 30 | Assign 30 to variable 1. |
| OFST | 1000 | *1 | Add the content of variable 1, or 30°, to the specified position of R-axis. |

## ● DEG (Set arc angle)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DEG | Angle | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set a division angle for the interpolation implemented by a CIR (move along circle) or ARC (move along arc) command.
When CIR or ARC is executed, a circle will be divided by the angle set here to calculate the passing points.
The angle is set in a range from 0 to 120°.
If the angle is set to “0”, an appropriate division angle will be calculated automatically so that the actuator will operate at the set speed (maximum 180°).
The angle is set in degrees and may include up to one decimal place.

(Note) If a CIR or ARC command is executed without setting an angle with this command, the default value registered in “All-axis parameter No. 30, Default division angle” will be used.

[Example 1] DEG 10 Set the division angle to 10°.

## ● BASE (Specify axis base)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | BASE | Datum axis number | Prohibited | CP |

| Applicable models |
|---|
| XSEL-JX/KX ×<br>Other than XSEL-JX/KX ○ |



[Function] Count the axes sequentially based on the axis number specified in operand 1 being the first axis.
BASE Command is available in PRED, PRDQ, AXST, actuator control commands, ARCH, PACH, PMVP, PMVL, zone commands, actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL, PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Note that each zone range is assigned to the actuator via parameter.

(Note 1) For XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD Commands are available in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL, PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Refer to the caution note for GRP and BASE Commands.

[Example 1]

```
BASE    5           Axis 5 is considered the first axis.
HOME    1           Axis 5 returns to the home.
HOME    10          Axis 6 returns to the home.
```

[Example 2]

```
LET     1     5     Assign 5 to variable 1.
BASE    *1          The content of variable 1 (axis 5) will be considered
                    as the first axis.
```

Thereafter, axes 5 and 6 move according to the specifications for axes 1 and 2.

## ● GRP (Set group axes)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GRP | Axis pattern | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Allow only the position data of the axis pattern specified in operand 1 to become valid.
The program assumes that there are no data for other axes not specified.
When multiple programs are run simultaneously, assigning axes will allow the same position data to be used effectively among the programs.
GRP Command is available in the operand axis pattern indication SEL commands except for OFST, DFTL, DFWK, DFIF, GTTL, GTWK and GTIF or the servo operation commands to use the position data, actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL, PTPE, PTPD, RIGH and LEFT, and the system information acquirement command GARM.
GRP Command activates in the condition before the axis number changed due to BASE Command.

(Note 1) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL, PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 "Axis Pattern Error" will occur if even one axis is set invalid by GRP and BASE Commands.

[Example] Express what axis is to be used by using either "1" or "0".

(Superior) (Inferior)

| Axis No. | 8-axis | 7-axis | 6-axis | 5-axis | 4-axis | 3-axis | 2-axis | 1-axis |
|---|---|---|---|---|---|---|---|---|
| Use | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| Unused | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

● When using 1st and 2nd axes;

2-axis
↓
000011… (0 in front are not necessary. Remove 0 and make it 11.)
↑
1-axis

● When using 1st and 4th axes;

4-axis
↓
1001… (In this case, 0 are necessary to express the position of the 4th axis.)
↑
1-axis

## ● HOLD (Hold: Declare axis port to pause)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | HOLD | (Input port, global flag) | (HOLD type) | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Declare an input port or global flag to pause while a servo command is being executed.
When operation is performed on the input port or global flag specified in operand 1, the current servo processing will pause. (If the axes are moving, they will decelerate to a stop.)
If nothing is specified in operand 1, the current pause declaration will become invalid.

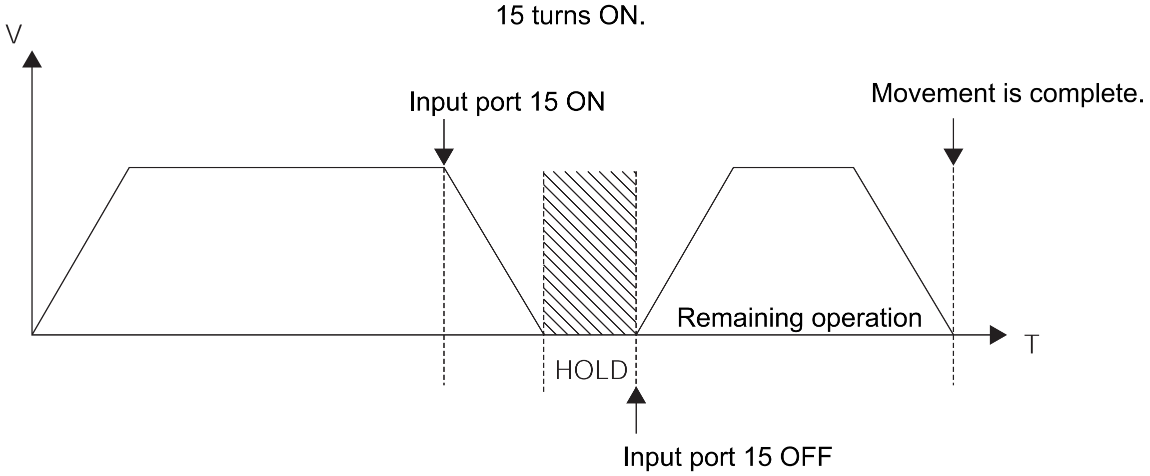
[HOLD type]
0 = Contact a (Deceleration stop)
1 = Contact b (Deceleration stop)
2 = Contact b (Deceleration stop → Servo OFF (The drive source will not be cut off))

The HOLD type is set to “0” (contact a) when the program is started.
If nothing is specified in operand 2, the current HOLD type will be used.
Using other task to issue a servo ON command to any axis currently stopped via a HOLD servo OFF will generate an “Error No. C66, Axis duplication error”. If the servo of that axis was ON prior to the HOLD stop, the system will automatically turn on the servo when the HOLD is cancelled. Therefore, do not issue a servo ON command to any axis currently stopped via a HOLD servo OFF.
If any axis currently stopped via a HOLD servo OFF is moved by external force, etc., from the stopped position, and when the servo of that axis was ON prior to the HOLD stop, the axis will move to the original stopped position when the HOLD is cancelled before resuming operation.

(Note 1) The input port or global flag specified by a HOLD declaration will only pause the axes used in the task (program) in which the HOLD is declared. The declaration will not be valid on axes used in different tasks (programs).

(Note 2) An input port or global flag to pause is valid for all active servo commands other than a SVOF command. (A deceleration stop will also be triggered in J☐W☐ and PATH operations.)

[Example] HOLD 15 0 The axes will decelerate to a stop when input port 15 turns ON.



## ● CANC (Cancel: Declare axis port to abort)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CANC | (Input port, global flag) | (CANC type) | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

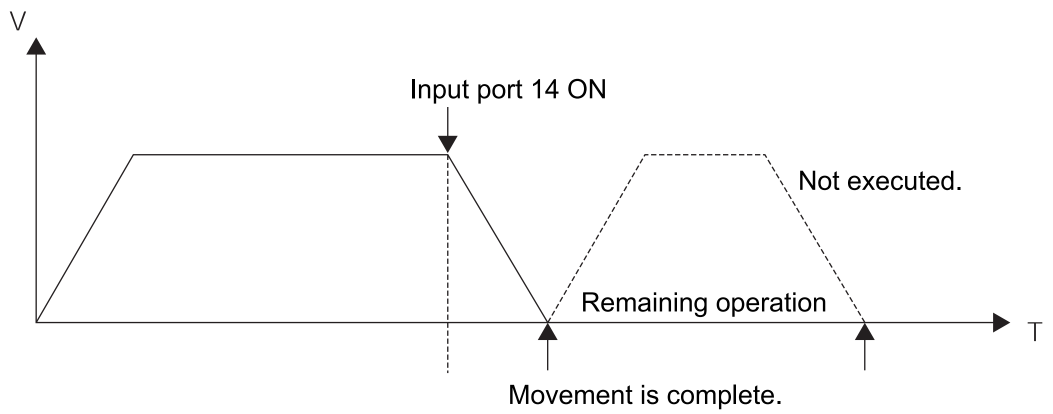
[Function] Declare an input port or global flag to abort while a servo command is being executed.
When operation is performed on the input port or global flag specified in operand 1, the current servo processing will be aborted. (If the axes are moving, they will decelerate to a stop before the processing is aborted.)
If nothing is specified in operand 1, the current abort declaration will become invalid.

[CANC type]
0 = Contact a (Deceleration stop)
1 = Contact b (Deceleration stop)
The CANC type is set to “0” (contact a) when the program is started.
If nothing is specified in operand 2, the current CANC type will be used.

(Note 1) The input port or global flag specified by a CANC command will only abort the axes used in the task (program) in which the CANC is declared. The declaration will not be valid on axes used in different tasks (programs).

(Note 2) An input port or global flag to pause is valid for all active servo commands other than a SVOF command. (A deceleration stop will also be triggered in JXWX and PATH operations.)

[Example] CANC 14 0 The axes will decelerate to a stop when input port 14 turns ON.



## ● CLLV (Collision Detection Level Setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CLLV | Axis pattern | Collision Detection Level | CP |

| Applicable models |
|---|
| XSEL-RAX/SAX/RAXD/SAXD(Main application V1.10 or later) |



[Function] The collision detection level [%] should be set for the axis pattern indicated in Operand 1. The range for detection level setting is from 1 to 400%.

(Note 1) At the controller startup, Axis-Specific Parameter No. 148 “Collision Detection Level Initial Value” should be the default.

(Note 2) After the collision detection level has been set with CLLV Command, it will be remained until the software reset is conducted or the power is turned OFF.

(Note 3) The sensitivity gets low as the detection level is set high while it gets high as the level is set low.

(Note 4) The risk of wrong detection gets high if the detection level is set low. Adjust the level considering it would not get too low.

(Note 5) The robot applicable for the collision detection feature is IXA SCARA Robot. Executing this command to a robot which is not applicable for the collision detection feature will not cause error but the command will be ignored.

(Note 6) There may be a case that it is necessary to change the setting of the collision detection level even for the same model robot depending on individual unit performance. Check the operation on each robot.

[Example]

| | | | |
|---|---|---|---|
| CLLV | 1111 | 100 | Set the collision detection level to 100% on axes from 1st to 4th |
| COL | 1 | 1111 | Activate the collision detection on axes from 1st to 4th |
| MOVP | 1 | | PTP Movement to Position No. 1 |
| COL | 0 | 1111 | Inactivate the collision detection on axes from 1st to 4th |

## ● COL (Collision Detection Feature Valid / Invalid Setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | COL | 0 or 1 | (Axis pattern) | CP |

| Applicable models |
|---|
| XSEL-RAX/SAX/RAXD/SAXD(Main application V1.10 or later) |

[Function] The collision detection feature should be set valid / invalid.
Indicate 0 (Collision Detection Invalid) / 1 (Collision Detection Valid) in Operand 1 and set the applicable axis pattern in Operand 2. All the axes on the connected SCARA Robot should be subject when Operand 2 is skipped.
When a collision is detected, Error No. 4C5 “Collision Detected” will be issued and the robot will get stopped. It is expected that the damage caused by a collision or interference between the robot and peripherals could get reduced. It is not a feature to protect the unit 100% from damage.

(Note 1) The robot applicable for the collision detection feature is IXA SCARA Robot. “C7A: Servo Unsupported Feature Error” will be issued when COL1 is executed to a robot axis which does not support the collision detection feature.
When COL0 is executed, it would not cause an error, but it will not be processed.

(Note 2) At the controller startup, settings in All Axes Common Parameter No. 134 and 232 “SCARA Collision Detection Feature Initial Value” should be the default.

(Note 3) After the collision detection is activated with COL1, it will be remained until it gets inactivated with COL0, the software reset is conducted or the power is turned OFF.

(Note 4) Even after the collision detection is unactivated with COL0, the collision detection level set with CLLV Command will be remained until the software reset is conducted or the power is turned OFF.

(Note 5) The collision detection cannot be activated while the compliance control is executed. When COL1 gets executed on an axis that the compliance control is in execution, Error No. 412 “Exclusion Mode Indicated Error” will be generated.

(Note 6) If the tip load condition setting (WGHT Command or WGT2 Command) is not established correctly, it could get high opportunity to have a wrong detection of collision. Make sure to establish the setting correctly before use.

[Example]

| | | | |
|---|---|---|---|
| CLLV | 1111 | 100 | Set the collision detection level to 100% on axes from 1st to 4th |
| COL | 1 | 1111 | Activate the collision detection on axes from 1st to 4th |
| MOVP | 1 | | PTP Movement to Position No. 1 |
| COL | 0 | 1111 | Inactivate the collision detection on axes from 1st to 4th |

## ● GCLX (Acquiring Max. Collision Level)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GCLX | Variable No. | Axis No. | CP |

| Applicable models |
|---|
| XSEL-RAX/SAX/RAXD/SAXD(Main application V1.10 or later) |

[Function] The maximum value [unit in %] of the collision level for the axis number indicated in Operand 2 should be stored in the variable number indicated in Operand 1. It is the maximum value of the collision level while the collision detection feature is activated.

(Note 1) While COL1 and CLLV Command are executed, the maximum value of the collision level gets initialized to 0 when the servo gets turned ON.

(Note 2) The robot applicable for the collision detection feature is IXA SCARA Robot. It should always be 0 acquired for a robot axis which does not support the collision detection feature.

[Example]

| | | | |
|---|---|---|---|
| COL | 1 | 1111 | Activate the collision detection on axes from 1st to 4th |
| MOVP | 1 | | PTP Movement to Position No. 1 |
| GCLX | 201 | 1 | Store collision level maximum value of 1st axis to variable 201 |
| GCLX | 202 | 2 | Store collision level maximum value of 2nd axis to variable 202 |
| GCLX | 203 | 3 | Store collision level maximum value of 3rd axis to variable 203 |
| GCLX | 204 | 4 | Store collision level maximum value of 4th axis to variable 204 |

## ● ACMX (Indicate ACMX acceleration) (Dedicated linear axis command)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ACMX | ACMX Acceleration No. | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | ○ | ○ | ○ (SSEL only | × | × |

[Function] Set the movement acceleration and deceleration of the actuator to the ACMX acceleration of the number indicated in Operand 1. Once ACMX Command is executed, the parameters registered in ACMX Acceleration No. 1 to 4 (Each Axis Parameters No. 2 to 5 and 194 to 197) are set as the movement acceleration and deceleration. It is necessary to set the parameters of ACMX acceleration in advance considering the conditions how to use the actuator (transportation weight, installation condition, etc.).

| ACMX Acceleration No. | Moving Direction | Acceleration | Deceleration |
|---|---|---|---|
| 1 | Positive | Each Axis Parameter No.2 Setting of “ACMX + Acceleration 1” | Each Axis Parameter No.3 Setting of “ACMX - Acceleration 1” |
| | Negative | Each Axis Parameter No.3 Setting of “ACMX - Acceleration 1” | Each Axis Parameter No.2 Setting of “ACMX + Acceleration 1” |
| 2 | Positive | Each Axis Parameter No.4 Setting of “ACMX + Acceleration 2” | Each Axis Parameter No.5 Setting of “ACMX - Acceleration 2” |
| | Negative | Each Axis Parameter No.5 Setting of “ACMX - Acceleration 1” | Each Axis Parameter No.4 Setting of “ACMX + Acceleration 2” |
| 3 | Positive | Each Axis Parameter No.194 Setting of “ACMX + Acceleration 3” | Each Axis Parameter No.195 Setting of “ACMX - Acceleration 3” |
| | Negative | Each Axis Parameter No.195 Setting of “ACMX - Acceleration 3” | Each Axis Parameter No.194 Setting of “ACMX + Acceleration 3” |
| 4 | Positive | Each Axis Parameter No.196 Setting of “ACMX + Acceleration 4” | Each Axis Parameter No.197 Setting of “ACMX - Acceleration 4” |
| | Negative | Each Axis Parameter No.197 Setting of “ACMX - Acceleration 4” | Each Axis Parameter No.196 Setting of “ACMX + Acceleration 4” |

(Note 1) It may generate an error is the acceleration or deceleration is set above the actuator specifications. Also, it cause a malfunction or drop of the production life.

(Note 2) The priority is put to the setting of acceleration and deceleration in the position data indicated with a movement command if there is any.

(Note 3) An operation is made within the range of the maximum acceleration and deceleration that would not exceed the ACMX acceleration/deceleration of each movement axis during the CP operation such as MOVL Command. In case constancy is required in the target acceleration/deceleration, indicate the acceleration and deceleration in ACC, DCL Command and the position data.

(Note 4) Do not attempt to indicate the ACMX acceleration/deceleration to the continuous movement related commands (PATH, PSPL, etc.).
It may cause a big speed drop depending on the direction of the movement position. Indicate the acceleration and deceleration in ACC, DCL Command and the position data.

(Note 5) Do not attempt to indicate the ACMX acceleration/deceleration to the extended motion control board axis movement commands. It would cause Error No. C89 "Acceleration/Deceleration Indication Error". Indicate the acceleration and deceleration in ACC, DCL Command and the position data.

(Note 6) ACMX Command is a command dedicated for the linear drive axes.

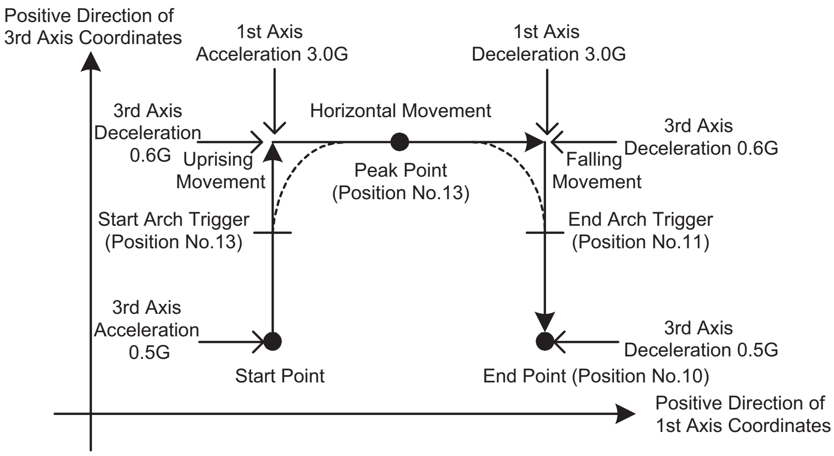
[Example 1] For arch motion movement (vertical axis to move up/down)

| | | | |
|---|---|---|---|
| VLMX | | | Set the speed setting in VLMX Speed. |
| ACMX | 1 | | Set the ACMX acceleration/deceleration of No. 1. |
| ACHZ | 3 | | Indicate the 3rd axis to Z-axis for arch motion. |
| ATRG | 13 | 11 | |
| ARCH | 10 | 12 | With Position No. 12 as the peak point, move with the arch motion to Position No. 10. |

- Setting for Example 1

| ACMX Acceleration No. | Each Axis Parameter No. | Parameter Name | Example for Setting | |
|---|---|---|---|---|
| | | | 1st Axis | 3rd Axis |
| 1 | 2 | ACMX + Acceleration 1 | 300 (3.0G) | 50 (0.5G) |
| | 3 | ACMX - Acceleration 1 | 300 (3.0G) | 60 (0.6G) |

- Operation of Example 1 (Acceleration/Deceleration in Arch Motion Movement)




5. SEL Commands

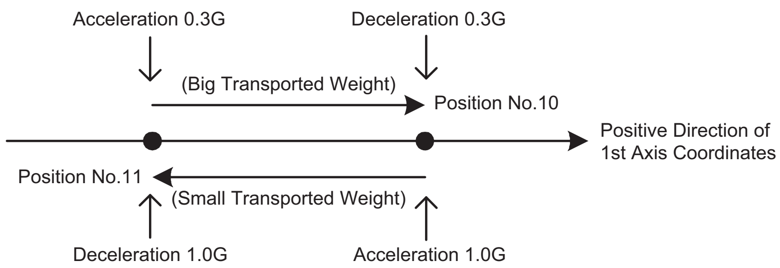
[Example 2] When the transported weight differs for going forward and backward

| | | |
|---|---|---|
| VLMX | | Set the speed setting in VLMX Speed. |
| ACMX | 1 | Set the ACMX acceleration/deceleration of No. 1. |
| MOVP | 10 | PTP movement is made to Position No. 10. |
| ACMX | 2 | Set the ACMX acceleration/deceleration of No. 2. |
| MOVP | 11 | PTP movement is made to Position No. 11. |

- Setting for Example 2

| ACMX Acceleration No. | Each Axis Parameter No. | Parameter Name | Example for Setting |
|---|---|---|---|
| | | | 1st Axis |
| 1 | 2 | ACMX + Acceleration 1 | 30 (0.3G) |
| | 3 | ACMX - Acceleration 1 | 30 (0.3G) |
| 2 | 4 | ACMX + Acceleration 2 | 100 (1.0G) |
| | 5 | ACMX - Acceleration 2 | 100 (1.0G) |

- Operation of Example 2



## ● VLMX (Dedicated linear axis command/Specify VLMX speed)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | VLMX | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| XSEL-JX/KX ×<br>Other than XSEL-JX/KX ○ |



[Function] Set the moving speed of a linear axis to the VLMX speed (normally maximum speed).
Executing a VLMX command will set the value registered in “Axis-specific parameter No. 29, VLMX speed” as the travel speed.

(Note 1) If the VLMX speed is specified for a continuous position movement command (PATH, PSPL), the target speed to each position becomes a composite speed based on the VLMX speed to the extent that each axis does not exceed the value set in axis-specific parameter No. 28, “Maximum PTP speed (SCARA axis)/ axis-specific maximum operating speed (linear axis)”. To keep the target speed constant, you must expressly specify the speed using a VEL command.

(Note 2) Error No. C88 “Velocity Specification Error” will occur if VLMX speed is indicated in case of CP operation held on the liner axes and SCARA axes at the same time. Indicate the speed with VEL Command.

[Example]

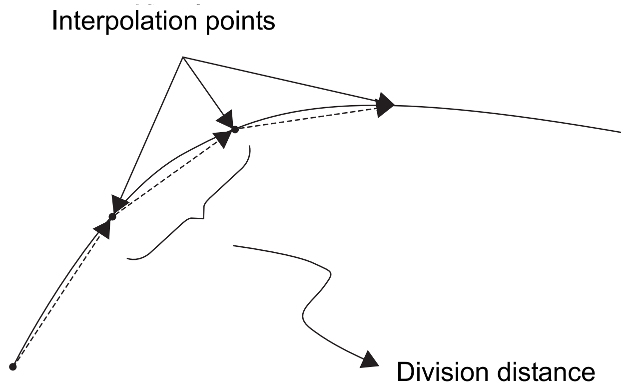
```
VEL     1000 ┐
MOVP    1    │ The speed becomes 1000mm/sec in this section.
MOVP    2    ┘
VLMX         ┐
MOVP    3    │ The speed becomes VLMXmm/sec in this section.
MOVP    4    ┘
```

## ● DIS (Set division distance at spline movement)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DIS | Distance | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set a division distance for the interpolation implemented by a PSPL (move along spline) command.
When a PSPL command is executed, a passing point will be calculated at each distance set here and the calculated passing points will be used as interpolation points.
If the distance is set to "0", an appropriate division distance will be calculated automatically so that the actuator will operate at the set speed.
The distance is input in mm.




(Note) If a PSPL command is executed without setting a distance with a DIS command, the default value registered in "All-axis parameter No. 31, Default division distance" will be used.

[Example] DIS 10 Set the division distance to 10mm.

## ● POTP (Set PATH output type)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | POTP | 0 or 1 | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

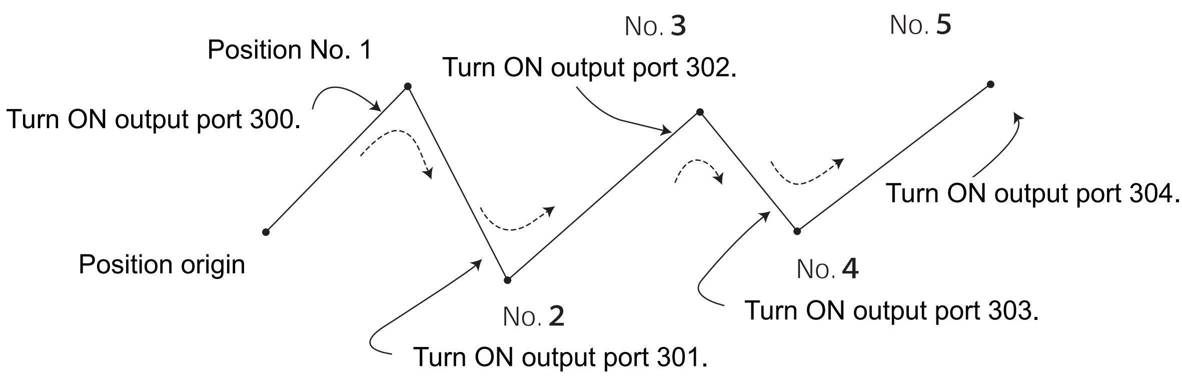
[Function] Set the output type in the output field to be used when a PATH and PSPL command is executed.
When a PATH and PSPL command is executed, the output will operate as follows in accordance with the setting of the POTP command.

1) POTP [Operand 1] = 0 (ON upon completion of operation)
The output port or flag will turn ON upon completion of operation.
2) POTP [Operand 1] = 1 (Increment and output on approaching each position; ON upon completion of operation for the last position)
During PATH or PSPL operation, the output port number or flag number specified in the output field will be incremented and turned ON when each specified position approaches.
At the last position, however, the output will turn ON upon completion of operation. This setting provides a rough guide for output in sequence control.

(Note 1) The default value of POTP, before it is set, is “0”.
(Note 2) If POTP = 1 and there is no valid data at the specified position, the output number will be incremented but the output will not turn ON. (The output number will be incremented regardless of the size of position numbers specified in operands 1 and 2 in a PATH or PSPL command.)

[Example]

| | | | | |
|---|---|---|---|---|
| POTP | 1 | | | |
| PATH | 1 | 5 | 300 | Turn ON output port No. 300 through 304 sequentially each time a specified position approaches during a pass movement from position No. 1 through 5, starting from the first position. |

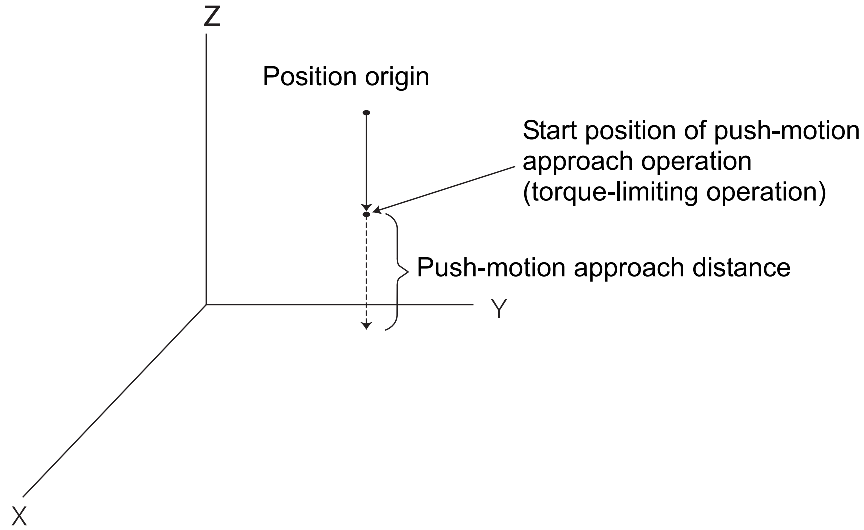


## ● PAPR (Set push-motion approach distance, speed)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PAPR | Distance | Speed | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the operation to be performed when a PUSH command is executed. Set the distance (push-motion approach distance) over which push-motion approach operation (torque-limiting operation) will be performed in operand 1 (in mm), and set the speed (push-motion approach speed) at which push-motion approach operation (torque-limiting operation) will be performed in operand 2 (in mm/sec).
The push-motion approach distance specified in operand 1 may contain up to three decimal places, while the speed specified in operand 2 cannot contain any decimal place.




[Example]

| | | | |
|---|---|---|---|
| PAPR | 100 | 30 | Set the push-motion approach distance in a PUSH command to 100mm and the push-motion approach speed to 30mm/sec. |
| MOVP | 10 | | |
| PUSH | 11 | | |

(Note) The push-motion approach speed in an OVRD command will be clamped by the minimum speed of 1mm/sec. (Correct push-motion operation is not guaranteed at the minimum speed. Operation at slow push-motion approach must be checked on the actual machine by considering the effects of mechanical characteristics, etc.)

## ● QRTN (Set quick-return mode)

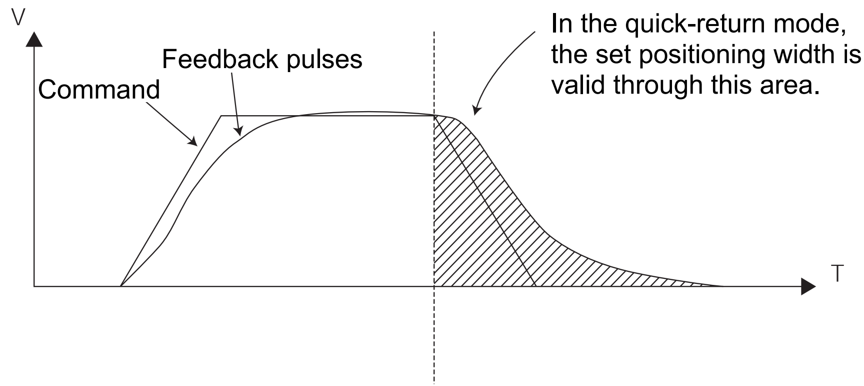
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | QRTN | 0 or 1 | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| ○ | ○ | ○ | ○ | × | × | × | ○ (V1.10 or later.) | ○ | ○ | ○ (PC/PG only) |



[Function] Set and cancel the quick-return mode.

1) QRTN [Operand 1] = 0 (Normal mode)
   Positioning is deemed complete when all command pulses have been output and the current position is inside the positioning width.
   * If a deceleration command is currently executed in the quick-return mode, the system will wait for all command pulses to be output.
2) QRTN [Operand 1] = 1 (Quick-return mode)
   Positioning is deemed complete when “a normal deceleration command is currently executed (excluding deceleration due to a stop command, etc.) or all command pulses have been output” and “the current position is inside the positioning width”. This setting is used to perform other processing during deceleration, in conjunction with a PBND command.




(Note 1) The quick-return mode will be cancelled when the program ends. (The positioning width set by a PBND command will not be cancelled.)

(Note 2) If a given axis is used even once in the quick-return mode, the program will not release the right to use the axis until the QRTN is set to “0” (normal mode) or the program ends. Any attempt to use the axis from other program will generate an “Error No. C66, Axis duplication error”.

(Note 3) Following a return from a normal deceleration command in the quick-return mode, the next positioning will start after all command pulses for the previous positioning have been output. Therefore, in the quick-return mode a simple reciprocating operation will require a longer tact time because of the extra completion check. In this sense, this setting should be used only if you wish to reduce the overall tact time by performing other processing during deceleration.

(Note 4) The quick-return mode represents very irregular processing. Therefore, be sure to revert to the normal mode when the overlay processing is completed in the necessary section.

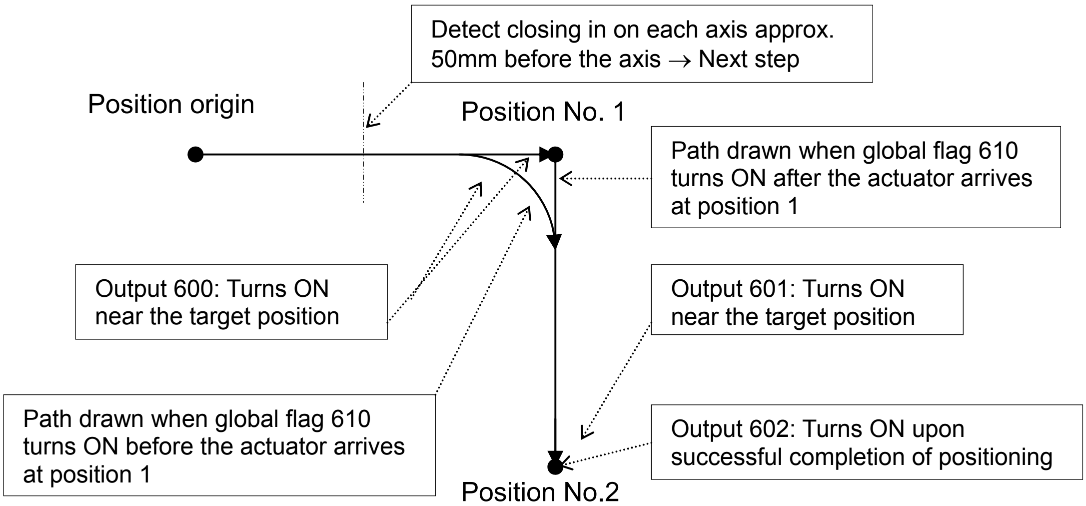
(Note 5) The quick-return mode cannot be used with a push-motion travel command or arc interpolation command.

(Note 6) It will be invalid if a SCARA axis is included in the operation axes to be operated with a CP operation command.

3) Quick return mode 2 (closeness-detection return target position addition mode)
* XSEL-J/K only
- When a MOVP, MOVL or PATH command (specifying the final moving position) is executed, closeness to the target position is detected when the close distance set by a NBND command is reached (or all command pulses are sent AND the positioning width is reached) while all used axes are positioning in steady state according to the applicable command, after which the command will be reset (quick return) and the SEL command in the next step will be executed.
  Set this mode if you want to perform other processing during positioning by using NBND and PEND commands together, or add a target position to operate the actuator continuously.
- If a MOVP, MOVL or PATH command is executed again while the actuator is moving in quick return mode 2, a target position will be added and the actuator will operate continuously.

[Example]

```
 :
QRTN   2                Set quick return mode to 2
NBND   11   50          Set close position for axes 1 and 2 to 50mm
MOVL   1          600   Move to position 1 (axes 1/2)
                        (Proceed to the next step when each axis reaches
                        approx. 50mm before the position.)
WTON   610              Wait for permission of movement to position 2 (610)
MOVL   2          601   Move to position 2 (axis 3)
PEND              602   Wait for all used axes to end operation
QRTN   0                Set quick return mode to 0
 :
```



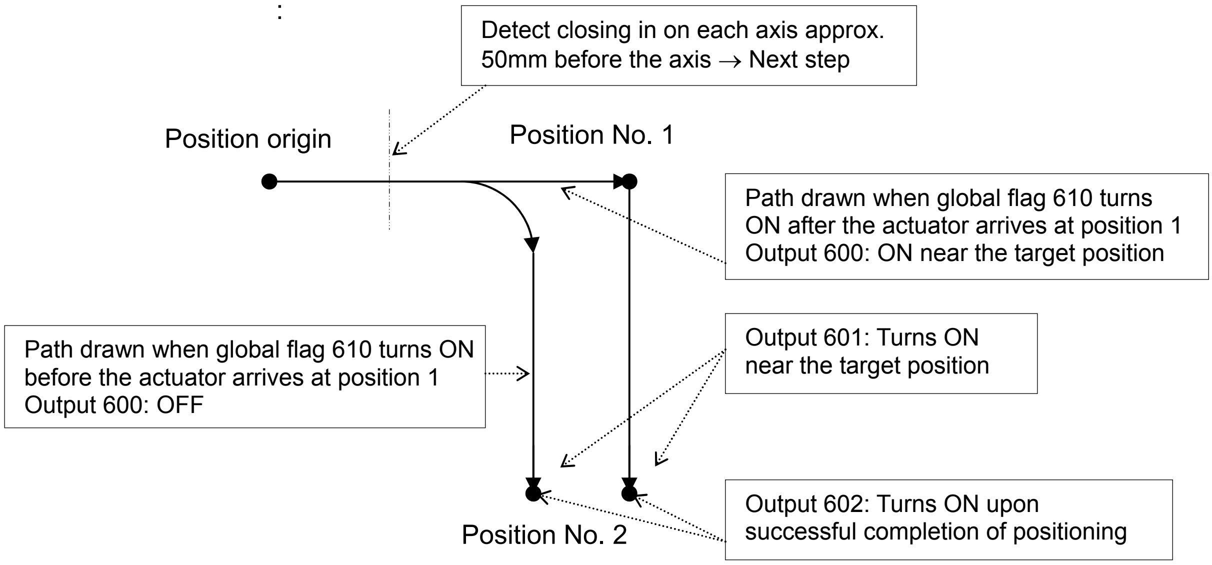

* This mode is invalid with respect to commands other than MOVP, MOVL and PATH.
  (With CIR2, ARC2, ARCC, ARCD, CIRS, ARCS, CIR, ARC, PSPL, MVPI and MVLI commands, “Error No. B24: Quick return mode error” occurs (= the command cannot be executed) because an unexpected path may be followed and a dangerous situation may result unless the start point is accurately understood.
* The close distance set by a NBND command must consider an allowance for the processing time in the next step onward following the quick return upon closeness detection (the specific processing time varies depending on the types of commands, number of steps, etc.) (this distance is not intended for use in precise processing).

* Behavior at the connection of movement commands when a new target position is added (when processing under the new movement command can be performed in time)
  If either the previous movement command (quick return) or new movement command is MOVP, the actuator starts moving to the target position under the new movement command simultaneously as the slowest axis starts decelerating under the previous movement command.
  If neither of the commands is MOVP (such as when MOVL and PATH commands are combined), the connection of operations is equivalent to what happens between normal PATH commands.
* During quick return mode 2, the output of a MOVP, MOVL or PATH command turns ON near the target position (regardless of the value set by the NBND command) (the operation is not yet complete). Use the output of a PEND command to check if the operation has completed (positioning has been successful).
* During quick return mode 2, the following tasks apply to all used axes for any operation with a MOVP command (they apply to all used axes even when specified for an individual axis):
  All stop processing including one by a STOP command
  Speed change by a CHVL command
* An attempt to switch from quick return mode 2 directly to quick return mode 1 generates “Error No. B24: Quick return mode error”.
* Software versions supporting quick return mode
  Controller main application: Ver.1.04 or later
  (excluding flash ROM 8Mbit versions)
  PC software: Ver.7.2.3.0 or later
  Teaching pendant:
  IA-T-X (D): Ver.1.44 or later
  SEL-T (D): Ver.1.02 or later
  TB-01 (D): First edition or later
  TB-02 (D): First edition or later

4) Quick return mode 3 (closeness-detection return target position addition mode)
   * XSEL-J/K only
   - When a MOVP, MOVL or PATH command (specifying the final moving position) is executed, closeness to the target position is detected when the close distance set by a NBND command is reached (or all command pulses are sent and the positioning width is reached) while all used axes are positioning in steady state according to the applicable command, after which the command will be reset (quick return) and the SEL command in the next step will be executed.
     Set this mode if you want to perform other processing during positioning by also using a NBND/PEND command or change the target position without stopping.
   - If the MOVP, MOVL or PATH command is executed again while the actuator is still moving as part of quick return in quick return mode 3, the actuator changes the target position (by decelerating to stop at the previous target position to cancel the position and then starting to move to the new target position) without stopping.

[Example]

```
   :
QRTN   3             Set quick return mode to 3
NBND   11   80       Set close position for axes 1 and 2 to 80mm
MOVL   1        600  Move to position 1 (axes 1/2)
                     (Proceed to the next step when each axis reaches
                     approx. 80mm before the position.)
WTON   610           Wait for permission of movement to position 2 (610)
MOVL   2        601  Move to position 2 (axis 3)
PEND            602  Wait for all used axes to end operation
QRTN   0             Set quick return mode to 0
   :
```




* This mode is invalid with respect to commands other than MOVP, MOVL and PATH.
  (With CIR2, ARC2, ARCC, ARCD, CIRS, ARCS, CIR, ARC, PSPL, MVPI and MVLI commands, “Error No. B24: Quick return mode error” occurs (= the command cannot be executed) because an unexpected path may be followed and a dangerous situation may result unless the start point is accurately understood.
* The close distance set by a NBND command must consider an allowance for the processing time in the next step onward following the quick return upon closeness detection (the specific processing time varies depending on the types of commands, number of steps, etc.) (this distance is not intended for use in precise processing).

* Transition between movement commands upon target position change
  The actuator starts moving to the target position under the new movement command roughly at the same time it starts cancelling the previous movement command via forced deceleration to a stop (there is a delay corresponding to the processing time to recalculate the target position).
* During quick return mode 3, the output of a MOVP, MOVL or PATH command turns ON near the target position (regardless of the value set by the NBND command) (the operation is not yet complete). Use the output of a PEND command to check if the operation has completed (positioning has been successful).
  However, the output is invalid if the target position was changed (cancelled via forced deceleration to a stop) before the start of normal deceleration (during acceleration or constant-speed operation), and so is the S-motion mode during forced deceleration after the target position has been changed.
* During quick return mode 3, the following tasks apply to all used axes for any operation with a MOVP command (they apply to all used axes even when specified for an individual axis):
  All stop processing including one by a STOP command
  Speed change by a CHVL command
* An attempt to switch from quick return mode 3 directly to quick return mode 1 generates "Error No. B24: Quick return mode error".
* Software versions supporting quick return mode 3
  Controller main application: Ver.1.04 or later
  (excluding flash ROM 8Mbit versions)
  PC software: Ver.7.2.3.0 or later
  Teaching pendant:
  IA-T-X (D): Ver.1.44 or later
  SEL-T (D): Ver.1.02 or later
  TB-01 (D): First edition or later
  TB-02 (D): First edition or later

(Note 1) Following a quick return from a SEL movement command, the right to use the applicable axis is not released in the program even after the command has been reset. Accordingly, an attempt to use that axis from other program generates "Error No. C66: Multiple axis use error". To release the right to use the applicable axis, set quick return mode 0 (Normal mode = Quick return mode cancelled).

(Note 2) Quick return modes 1 to 3 are cancelled when the program ends (the close distance set by the NBND command and positioning width set by the PBND command are not cancelled).

(Note 3) At the end of combined processing requiring a quick return, be sure to reset the quick return mode to 0 (Normal mode = Quick return mode cancelled).

(Note 4) Always refer to the pages explaining the NBND and PEND commands.

## ● DFTL (Define tool coordinate system)

(Note) When using this command in MSEL-PC/PG or TTA, set All Axes Parameter No. 55 and No. 56. (MSEL-PC/PG or TTA main application V2.00 or later)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DFTL | Tool coordinate system number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | ○ (TTA only) | ○ |

[Function] Set the position data in operand 2 as the tool coordinate system offset data specified in operand 1.
The position data for all the SCARA axes go into the tool coordinate system offset data, however, 0 will be set for an axis that the position data is invalid. In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD if all the position data for the SCARA axes in one unit are invalid, data cannot be established in the tool coordinate system offset, and the data before executing DFTL Command is saved.
In MSEL-PCX/PGX, position data for four axes needs to be set in the tool coordinate system offset data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the tool coordinate system data is set from is used as the dedicated data for the tool coordinate set, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, Axis 4 in the position data set to the tool coordinate system in DFTL is not a target position for the additional linear axis.
In MSEL-PC/PG and TTA, tool coordinate system offset X, Y, Z and R is set in the work coordinate system offset on Coordinate System Definition 1 Constructing Axes Setting Value in order from Axis 1 data (All Axes Parameter No. 56).

(Note 1) The tool/work coordinate systems are functions available for SCARA Robot. Also, this is a feature for the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56).

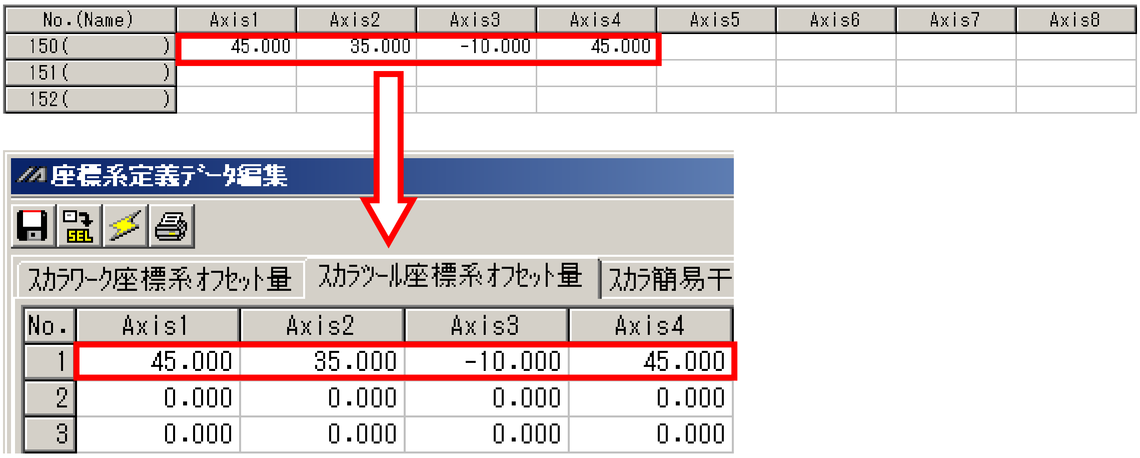
(Note 2) Since tool coordinate system No. 0 is reserved by the system as a condition specifying no tool offset, selecting this number generates “Error No. B71: Coordinate system number error”.

(Note 3) The GRP command is invalid with respect to this command.

(Note 4) “Error No. B71 Coordinate System Number Error” will occur if this command is executed when Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 0 invalid in MSEL-PC/PG or TTA.
MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-02 (D) of first edition later, TB-01 (D) of Ver 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D))

(Note 5) Ordinary when using the PC software in MSEL-PC/PG or TTA and opened the position edit window, only the axis data set valid in All Axes Parameter No. 1 “Valid Axis Pattern” will be displayed.
However, it is possible that the position data for all the four axes can always be displayed and available for editing by setting in the environment setting window in the PC software.
[Refer to the instruction manuals of XSEL PC software Instruction Manual for details]
By conducting the setting above in the environment setting window, referring to and editing the position data in the position edit window become available for the 4th axis applicable to R-axis offset.
(Also, it becomes always available to refer to and edit data for all the four axes for the operation of position data by PPUT Command and PGET Command regardless of the parameter settings or the setting in the PC software.)

[XSEL-JX/KX/PX/QX/RX/SX/RAX/SAX: 1 unit of SCARA connected]
[Example] DFTL 1 150



| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 150( ) | 45.000 | 35.000 | -10.000 | 45.000 | | | | |
| 151( ) | | | | | | | | |
| 152( ) | | | | | | | | |



| No. | Axis1 | Axis2 | Axis3 | Axis4 |
|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 |
| 2 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 |

[XSEL-RXD/SXD/RAXD/SAXD: 2 unit of SCARA connected]

[Example 1]    DFTL    1    150

In case that the command shown above is executed with the position data as shown below, the data is set to Axis 1 to 4 in Tool Coordinate System No. 1 as the position data in Axis 1 to 4 for the SCARA axes (1st to 4th axes) are set effective. There will be no change to Axis 5 to 8 in Tool Coordinate System No. 1 as the position data in Axis 5 to 8 for the SCARA axes (5th to 8th axes) are all set ineffective.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 150( ) | 45.000 | 35.000 | -10.000 | 45.000 | | | | |
| 151( ) | | | | | | | | |
| 152( ) | | | | | | | | |

座標系定義ﾃﾞｰﾀ編集

ｽｶﾗﾜｰｸ座標系ｵﾌｾｯﾄ量 | ｽｶﾗﾂｰﾙ座標系ｵﾌｾｯﾄ量 | ｽｶﾗ簡易干渉ﾁｪｯｸｿﾞｰﾝ定義座標

| No. | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 | 45.000 | 35.000 | -10.000 | 45.000 |
| 2 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

[Example 2]    DFTL    2    152

In case that the command shown above is executed with the position data as shown below, the data is set to Axis 1 to 8 in Tool Coordinate System No. 2 as the position data in either of Axis 1 to 4 or Axis 5 to 8 for the SCARA axes (1st to 4th axes) or SCARA axes (5th to 8th axes) is set effective.

However, 0 will be set to Axis 1 to 2 and 7 to 8 that the position data is the invalid axes.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 150( ) | 45.000 | 35.000 | -10.000 | 45.000 | | | | |
| 151( ) | | | | | | | | |
| 152( ) | | | -10.000 | 45.000 | 45.000 | 35.000 | | |

座標系定義ﾃﾞｰﾀ編集

ｽｶﾗﾜｰｸ座標系ｵﾌｾｯﾄ量 | ｽｶﾗﾂｰﾙ座標系ｵﾌｾｯﾄ量 | ｽｶﾗ簡易干渉ﾁｪｯｸｿﾞｰﾝ定義座標

| No. | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2 | 0.000 | 0.000 | -10.000 | 45.000 | 45.000 | 35.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

## ● SLTL (Select tool coordinate system)

(Note) When using this command in MSEL-PC/PG or TTA, set All Axes Parameter No. 55 and No. 56. (MSEL-PC/PG or TTA main application V2.00 or later)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SLTL | Tool coordinate system number | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | ○ (TTA only) | ○ |

[Function] Set the tool coordinate system selection number in operand 1.

(Note 1) The tool/work coordinate systems are functions available for SCARA Robot. Also, this is a feature for the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56).

(Note 2) The selected number last declared in the system becomes effective. The selected tool coordinate system number will remain effective even after the program ends, and also after the power is reconnected if the system-memory backup battery is installed (Note 6).

(Note 3) Only one tool coordinate system selection number is present within the system.

(Note 4) Expressly declare SLTL in the program to prevent unwanted problems resulting from forgetting to reset the coordinate system selection number after changing it in the PC software or on the teaching pendant.
(Execute SLTL 0, if the tool coordinate system is not used.)

(Note 5) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

(Note 6) XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD, MSEL controller or TTA save the tool coordinate system numbers without using a battery.

(Note 7) “Error No. B71 Coordinate System Number Error” will occur if this command is executed when Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 0 invalid in MSEL-PC/PG or TTA.

| | | | |
|---|---|---|---|
| [Example 1] | GRP | 1111 | It makes the 1st to 4th axes effective. |
| | SLTL | 1 | Selected tool coordinate system of the SCARA axes (1st to 4th axes) is changed to No. 1. |
| [Example 2] | GRP | 11111111 | It makes the 1st to 8th axes effective. |
| | SLTL | 2 | Selected tool coordinate system of the SCARA axes (1st to 4th axes) and the SCARA axes (5th to 8th axes) is changed to No. 2. |

MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-02 (D) of first edition later, TB-01 (D) of Ver 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D))

## ● GTTL (Get tool coordinate system definition data)

(Note) When using this command in MSEL-PC/PG or TTA, set All Axes Parameter No. 55 and No. 56. (MSEL-PC/PG or TTA main application V2.00 or later)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GTTL | Tool coordinate system number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | ○ (TTA only) | ○ |

[Function] Set the tool coordinate system offset data specified in operand 1 for the position data specified in operand 2. Tool coordinate system offset data for all SCARA axes is set for the position data.
In MSEL-PCX/PGX, tool coordinate system offset data for four axes is set in the position data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the tool coordinate system is acquired from is used as the dedicated data for the tool coordinate acquirement, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, the tool coordinate system R-axis offset in the position data is written by execution of GTTL Command.
In MSEL-PC/PG and TTA, tool coordinate system offset X, Y, Z and R is set in the work coordinate system offset on Coordinate System Definition 1 Constructing Axes Setting Value in order from Axis 1 data (All Axes Parameter No. 56).

(Note 1) The tool/work coordinate systems are functions available for SCARA Robot. Also, this is a feature for the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56).

(Note 2) The position data for the liner axes (5th to 8th axes) are cleared when the command is executed.

(Note 3) Since tool coordinate system No. 0 is reserved by the system as a condition specifying no tool offset, selecting this number generates “Error No. B71: Coordinate system number error”.

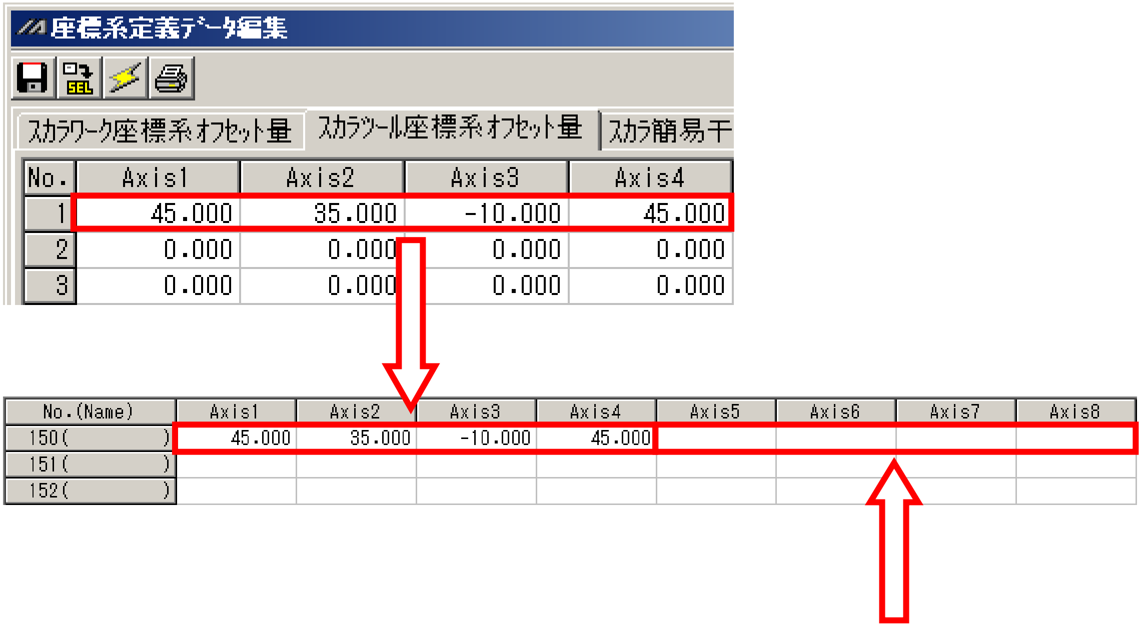
(Note 4) The GRP command is invalid with respect to this command.

(Note 5) “Error No. B71 Coordinate System Number Error” will occur if this command is executed when Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 0 invalid in MSEL-PC/PG or TTA.

(Note 6) Ordinary when using the PC software in MSEL-PC/PG or TTA and opened the position edit window, only the axis data set valid in All Axes Parameter No. 1 “Valid Axis Pattern” will be displayed.
However, it is possible that the position data for all the four axes can always be displayed and available for editing by setting in the environment setting window in the PC software.
[Refer to the instruction manuals of XSEL PC software Instruction Manual for details]

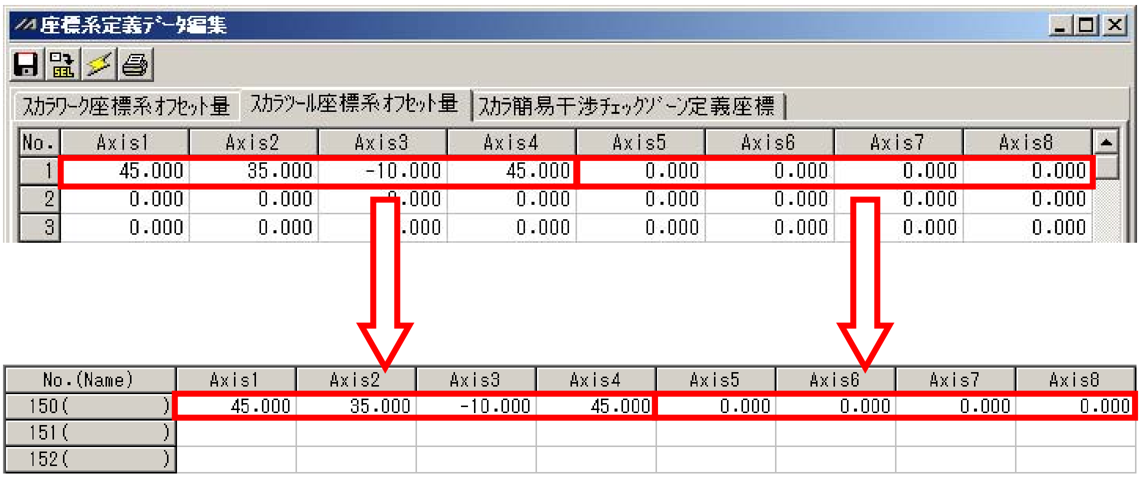
By conducting the setting above in the environment setting window, referring to and editing the position data in the position edit window become available for the 4th axis applicable to R-axis offset.
(Also, it becomes always available to refer to and edit data for all the four axes for the operation of position data by PPUT Command and PGET Command regardless of the parameter settings or the setting in the PC software.)
MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-02 (D) of first edition later, TB-01 (D) of Ver 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D))

[XSEL-JX/KX/PX/QX/RX/SX/RAX/SAX: 1 unit of SCARA connected]
[Example] GTTL 1 150
After the command shown above is executed, the position data for the liner axes (5th to 8th axes) are cleared.




The data before GTTL Command was executed gets cleared.

[XSEL-RXD/SXD/RAXD/SAXD: 2 unit of SCARA connected]
[Example] GTTL 1 150



## ● DFWK (Define work coordinate system)

(Note) When using this command in MSEL-PC/PG or TTA, set All Axes Parameter No. 55 and No. 56. (MSEL-PC/PG or TTA main application V2.00 or later)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DFWK | Work coordinate system number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | ○ (TTA only) | ○ |

[Function] Set the position data in operand 2 for the work coordinate system offset data specified in operand 1. The position data for all the axes go into the work coordinate system offset data, however, 0 will be set for an axis that the position data is invalid.
In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD, if all the position data for the SCARA axes in one unit are invalid, data cannot be established in the tool coordinate system offset, and the data before executing DFWK Command is saved
In MSEL-PCX/PGX, position data for four axes is set in the work coordinate system offset data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the work coordinate system data is set from is used as the dedicated data for the work coordinate set, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, Axis 4 in the position data set to the tool coordinate system in DFWK is not a target position for the additional linear axis.
In MSEL-PC/PG and TTA, tool coordinate system offset X, Y, Z and R is set in the work coordinate system offset on Coordinate System Definition 1 Constructing Axes Setting Value in order from Axis 1 data (All Axes Parameter No. 56).

(Note 1) The tool/work coordinate systems are functions available for SCARA Robot. Also, this is a feature for the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56).

(Note 2) Since work coordinate system No. 0 is reserved by the system as the base coordinate system, selecting this number generates “Error No. B71: Coordinate system number error”.

(Note 3) The GRP command is invalid with respect to this command.

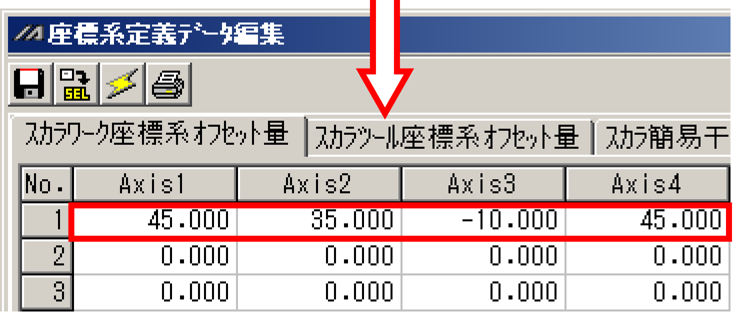
(Note 4) “Error No. B71 Coordinate System Number Error” will occur if this command is executed when Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 0 invalid in MSEL-PC/PG or TTA.

MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-02 (D) of first edition later, TB-01 (D) of Ver 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D))

(Note 5) Ordinary when using the PC software in MSEL-PC/PG or TTA and opened the position edit window, only the axis data set valid in All Axes Parameter No. 1 “Valid Axis Pattern” will be displayed.
However, it is possible that the position data for all the four axes can always be displayed and available for editing by setting in the environment setting window in the PC software.
[Refer to the instruction manuals of XSEL PC software Instruction Manual for details]
By conducting the setting above in the environment setting window, referring to and editing the position data in the position edit window become available for the 4th axis applicable to R-axis offset.
(Also, it becomes always available to refer to and edit data for all the four axes for the operation of position data by PPUT Command and PGET Command regardless of the parameter settings or the setting in the PC software.)

[XSEL-JX/KX/PX/QX/RX/SX/RAX/SAX: 1 unit of SCARA connected]
[Example] DFWK 1 150

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 150 ( ) | 45.000 | 35.000 | -10.000 | 45.000 | | | | |
| 151 ( ) | | | | | | | | |
| 152 ( ) | | | | | | | | |




| No. | Axis1 | Axis2 | Axis3 | Axis4 |
|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 |
| 2 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 |

[XSEL-RXD/SXD/RAXD/SAXD: 2 unit of SCARA connected]

[Example 1]    DFWK    1    150

In case that the command shown above is executed with the position data as shown below, the data is set to Axis 1 to 4 in Work Coordinate System No. 1 as the position data in Axis 1 to 4 for the SCARA axes (1st to 4th axes) are set effective. There will be no change to Axis 5 to 8 in Work Coordinate System No. 1 as the position data in Axis 5 to 8 for the SCARA axes (5th to 8th axes) are all set ineffective.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 150( ) | 45.000 | 35.000 | -10.000 | 45.000 | | | | |
| 151( ) | | | | | | | | |
| 152( ) | | | | | | | | |

座標系定義ﾃﾞｰﾀ編集

ｽｶﾗﾜｰｸ座標系ｵﾌｾｯﾄ量 | ｽｶﾗﾂｰﾙ座標系ｵﾌｾｯﾄ量 | ｽｶﾗ簡易干渉ﾁｪｯｸｿﾞｰﾝ定義座標

| No. | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 | 45.000 | 35.000 | -10.000 | 45.000 |
| 2 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

[Example 2]    DFWK    2    152

In case that the command shown above is executed with the position data as shown below, the data is set to Axis 1 to 8 in Work Coordinate System No. 2 as the position data in either of Axis 1 to 4 or Axis 5 to 8 for the SCARA axes (1st to 4th axes) or SCARA axes (5th to 8th axes) is set effective.

However, 0 will be set to Axis 1 to 2 and 7 to 8 that the position data is the invalid axes.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 150( ) | 45.000 | 35.000 | -10.000 | 45.000 | | | | |
| 151( ) | | | | | | | | |
| 152( ) | | | -10.000 | 45.000 | 45.000 | 35.000 | | |

座標系定義ﾃﾞｰﾀ編集

ｽｶﾗﾜｰｸ座標系ｵﾌｾｯﾄ量 | ｽｶﾗﾂｰﾙ座標系ｵﾌｾｯﾄ量 | ｽｶﾗ簡易干渉ﾁｪｯｸｿﾞｰﾝ定義座標

| No. | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2 | 0.000 | 0.000 | -10.000 | 45.000 | 45.000 | 35.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

## ● SLWK (Select work coordinate system)

(Note) When using this command in MSEL-PC/PG or TTA, set All Axes Parameter No. 55 and No. 56. (MSEL-PC/PG or TTA main application V2.00 or later)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SLWK | Work coordinate system number | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | ○ (TTA only) | ○ |

[Function] Set the work coordinate system selection number in operand 1.

(Note 1) The tool/work coordinate systems are functions available for SCARA Robot. Also, this is a feature for the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56).

(Note 2) The selected number last declared in the system becomes effective. The selected work coordinate system number will remain effective even after the program ends, and also after the power is reconnected if the system-memory backup battery is installed (Note 6).

(Note 3) Only one work coordinate system selection number is present within the system.

(Note 4) Expressly declare SLWK in the program to prevent unwanted problems resulting from forgetting to reset the coordinate system selection number after changing it in the PC software or on the teaching pendant.
(Execute SLWK 0, if the work coordinate system is not used.)

(Note 5) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

(Note 6) XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD, MSEL Controller or TTA save the tool coordinate system numbers without using a battery.

(Note 7) “Error No. B71 Coordinate System Number Error” will occur if this command is executed when Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 0 invalid in MSEL-PC/PG or TTA.

| | | | |
|---|---|---|---|
| [Example 1] | GRP | 1111 | It makes the 1st to 4th axes effective. |
| | SLWK | 1 | Selected work coordinate system of the SCARA axes (1st to 4th axes) is changed to No. 1. |
| [Example 2] | GRP | 11111111 | It makes the 1st to 8th axes effective. |
| | SLWK | 2 | Selected work coordinate system of the SCARA axes (1st to 4th axes) and the SCARA axes (5th to 8th axes) is changed to No. 1. |

MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-02 (D) of first edition later, TB-01 (D) of Ver 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D)) (Not applicable for SEL-T (D) and IA-T-X (D))

## ● GTWK (Get work coordinate system definition number)

(Note) When using this command in MSEL-PC/PG or TTA, set All Axes Parameter No. 55 and No. 56. (MSEL-PC/PG or TTA main application V2.00 or later)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GTWK | Work coordinate system number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | ○ (TTA only) | ○ |

[Function] Set the work coordinate system offset data specified in operand 1 for the position data specified in operand 2. Work coordinate system offset data for all axes is set for the position data.
In MSEL-PCX/PGX, work coordinate system offset data for four axes is set in the position data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the work coordinate system is acquired from is used as the dedicated data for the work coordinate acquirement, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, the work coordinate system R-axis offset in the position data is written by execution of GTWK Command.
In MSEL-PC/PG and TTA, tool coordinate system offset X, Y, Z and R is set in the work coordinate system offset on Coordinate System Definition 1 Constructing Axes Setting Value in order from Axis 1 data (All Axes Parameter No. 56).

(Note 1) The tool/work coordinate systems are functions available for SCARA Robot. Also, this is a feature for the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56).

(Note 2) The position data for the liner axes (5th to 8th axes) are cleared when the command is executed.

(Note 3) Since work coordinate system No. 0 is reserved by the system as the base coordinate system, selecting this number generates "Error No. B71: Coordinate system number error".

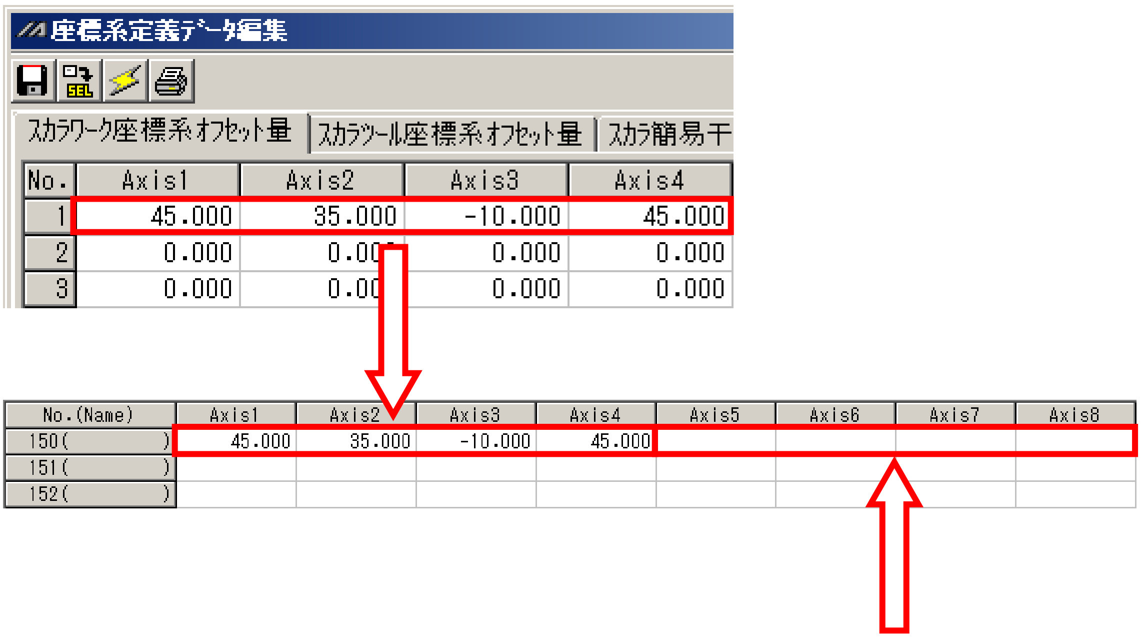
(Note 4) The GRP command is invalid with respect to this command.

(Note 5) "Error No. B71 Coordinate System Number Error" will occur if this command is executed when Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 0 invalid in MSEL-PC/PG or TTA.

(Note 6) Ordinary when using the PC software in MSEL-PC/PG or TTA and opened the position edit window, only the axis data set valid in All Axes Parameter No. 1 "Valid Axis Pattern" will be displayed.
However, it is possible that the position data for all the four axes can always be displayed and available for editing by setting in the environment setting window in the PC software.
[Refer to the instruction manuals of XSEL PC software Instruction Manual for details]
By conducting the setting above in the environment setting window, referring to and editing the position data in the position edit window become available for the 4th axis applicable to R-axis offset.
(Also, it becomes always available to refer to and edit data for all the four axes for the operation of position data by PPUT Command and PGET Command regardless of the parameter settings or the setting in the PC software.)

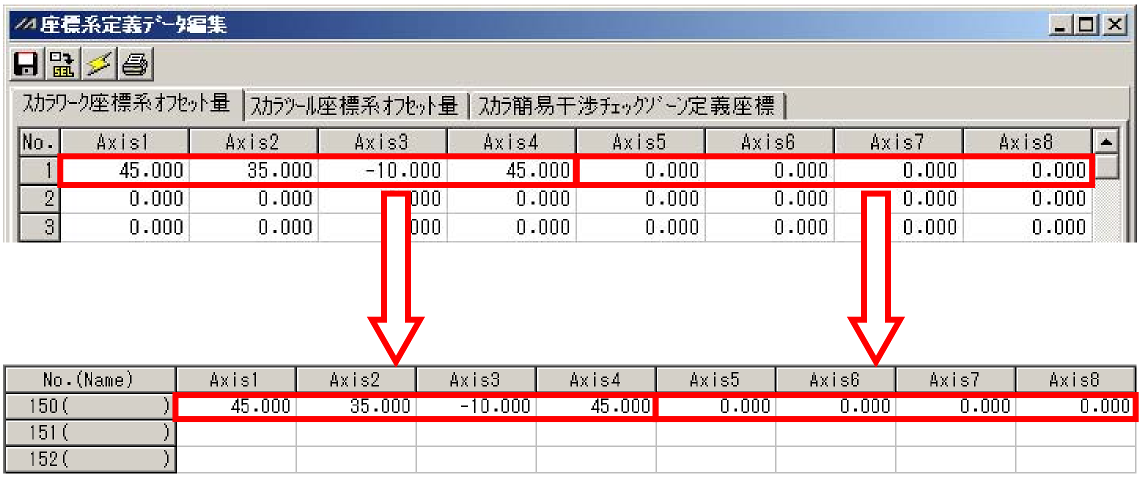
MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-02 (D) of first edition later, TB-01 (D) of Ver 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D)) (Not applicable for SEL-T (D) and IA-T-X (D))

[XSEL-JX/KX/PX/QX/RX/SX/RAX/SAX: 1 unit of SCARA connected]
[Example] GTWK 1 150




The data before GTWK Command was executed gets cleared.

[XSEL-RXD/SXD/RAXD/SAXD: 2 unit of SCARA connected]
[Example]　　GTWK　1　　150



座標系定義データ編集

スカラワーク座標系オフセット量 | スカラツール座標系オフセット量 | スカラ簡易干渉チェックゾーン定義座標

| No. | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 1 | 45.000 | 35.000 | -10.000 | 45.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2 | 0.000 | 0.000 | 000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Axis5 | Axis6 | Axis7 | Axis8 |
|---|---|---|---|---|---|---|---|---|
| 150( ) | 45.000 | 35.000 | -10.000 | 45.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 151( ) | | | | | | | | |
| 152( ) | | | | | | | | |

## ● RIGH (Dedicated SCARA command/Change current arm system to right arm (arm 2 operation involved if current arm system is opposite))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RIGH | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |



[Function] Change the current SCARA arm system to the right arm system. If the current arm system is the left arm system, arm 2 is moved to change it to the right arm system. After the operation, arms 1 and 2 form a straight line. No arm operation is performed if the current arm system is the right arm system.

(Note 1) To use a RIGH or LEFT command, the speed must be set with VELS even when a SCARA PTP operation command is not used.

(Note 2) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

| | | | |
|---|---|---|---|
| [Example 1] | GRP | 1111 | It makes the 1st to 4th axes effective. |
| | RIGH | | The current arm system of the SCARA axes (1st to 4th axes) is changed to the right arm system. |
| [Example 2] | GRP | 11111111 | It makes the 1st to 8th axes effective. |
| | RIGH | | The current arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is changed to the right arm system. |

## ● LEFT (Dedicated SCARA command/Change current arm system to left arm (arm 2 operation involved if current arm system is opposite))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | LEFT | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |



[Function] Change the current SCARA arm system to the left arm system. If the current arm system is the right arm system, arm 2 is moved to change it to the left arm system. After the operation, arms 1 and 2 form a straight line. No arm operation is performed if the current arm system is the left arm system.

(Note 1) To use a RIGH or LEFT command, the speed must be set with VELS even when a SCARA PTP operation command is not used.

(Note 2) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

| | | | |
|---|---|---|---|
| [Example 1] | GRP | 1111 | It makes the 1st to 4th axes effective |
| | LEFT | | The current arm system of the SCARA axes (1st to 4th axes) is changed to the right arm system. |
| [Example 2] | GRP | 11111111 | It makes the 1st to 8th axes effective. |
| | LEFT | | The current arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is changed to the left arm system. |

## ● PTPR (Dedicated SCARA command/Specify PTP target arm system to right arm (Movement of opposite arm system prohibited (no operation is performed) if target unachievable)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTPR | Prohibited | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Specify the target arm system for SCARA PTP operation commands to the right arm system. After the PTPR command is executed, the target arm system for SCARA PTP operation commands becomes the right arm system and an error occurs if the target value cannot be achieved by operating on the right arm system. Executing this command does not initiate any arm operation.

(Note 1) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]

| | | |
|---|---|---|
| GRP | 1111 | It makes the 1st to 4th axes effective. |
| PTPR | | PTP target arm system of the SCARA axes (1st to 4th axes) is indicated to the right arm system. |
| MOVP | 1 | Move to Position No. 1 to become the right arm system. |

[Example 2]

| | | |
|---|---|---|
| GRP | 11111111 | It makes the 1st to 8th axes effective. |
| PTPR | | PTP target arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is indicated to the right arm system. |
| MOVP | 2 | Move to Position No. 2 to become the right arm system. |

## ● PTPL (Dedicated SCARA command/Specify PTP target arm system to left arm (Movement of opposite arm system prohibited (no operation is performed) if target unachievable)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTPL | Prohibited | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Specify the target arm system for SCARA PTP operation commands to the left arm system. After the PTPL command is executed, the target arm system for SCARA PTP operation commands becomes the left arm system and an error occurs if the target value cannot be achieved by operating on the left arm system. Executing this command does not initiate any arm operation.

(Note 1) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]

| | | |
|---|---|---|
| GRP | 1111 | It makes the 1st to 4th axes effective. |
| PTPL | | PTP target arm system of the SCARA axes (1st to 4th axes) is indicated to the left arm system. |
| MOVP | 1 | Move to Position No. 1 to become the left arm system. |

[Example 2]

| | | |
|---|---|---|
| GRP | 11111111 | It makes the 1st to 8th axes effective. |
| PTPL | | PTP target arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is indicated to the left arm system. |
| MOVP | 2 | Move to Position No. 2 to become the left arm system. |

## ● PTPD (Dedicated SCARA command/Specify PTP target arm system to current arm (Movement of opposite arm system permitted (no operation is performed) if target unachievable)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTPD | Prohibited | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Specify the target arm system for SCARA PTP operation commands to the current arm system. After the PTPD command is executed, the target arm system for SCARA PTP operation commands becomes the current arm system and an error occurs if the target value cannot be achieved by operating on this arm system. Executing this command does not initiate any arm operation.

(Note 1) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 "Axis Pattern Error" will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]

| | | |
|---|---|---|
| GRP | 1111 | It makes the 1st to 4th axes effective. |
| PTPD | | PTP target arm system of the SCARA axes (1st to 4th axes) is indicated to the reversed arm system move prohibited when it is not possible. |
| MOVP | 1 | Movement is made to Position No. 1 with the current arm system. ("C73: Target Track Software Limit Excess Error" will occur when positioning cannot be performed without changing to the reversed arm system.) |

[Example 2]

| | | |
|---|---|---|
| GRP | 11111111 | It makes the 1st to 8th axes effective. |
| PTPR | | PTP target arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is indicated to the reversed arm system move prohibited when it is not possible. |
| MOVP | 2 | Movement is made to Position No. 2 with the current arm system. ("C73: Target Track Software Limit Excess Error" will occur when positioning cannot be performed without changing to the reversed arm system.) |

## ● PTPE (Dedicated SCARA command/Specify PTP target arm system to current arm (Movement of opposite arm system prohibited (no operation is performed) if target unachievable)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTPE | Prohibited | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Specify the target arm system for SCARA PTP operation commands to the current arm system. After the PTPE command is executed, the target arm system for SCARA PTP operation commands becomes the current arm system and if the target value cannot be achieved by operating on this arm system, the target arm system is changed to the one opposite the current arm system. An error occurs if the target value cannot be achieved by operating on either the right arm system or left arm system. Executing this command does not initiate any arm operation.

(Note 1) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, WGT2, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

| | | | |
|---|---|---|---|
| [Example 1] | GRP | 1111 | It makes the 1st to 4th axes effective. |
| | PTPE | | PTP target arm system of the SCARA axes (1st to 4th axes) is indicated to the reversed arm system move permitted when it is not possible. |
| | MOVP | 1 | Movement is made to Position No. 1 with the current arm system.<br>(Positioning is performed with the reversed arm system when positioning cannot be performed without changing to the reversed arm system.) |
| [Example 2] | GRP | 11111111 | It makes the 1st to 8th axes effective. |
| | PTPE | | PTP target arm system of the SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) is indicated to the reversed arm system move permitted when it is not possible. |
| | MOVP | 2 | Movement is made to Position No. 2 with the current arm system.<br>(Positioning is performed with the reversed arm system when positioning cannot be performed without changing to the reversed arm system.) |

## ● DFIF (Dedicated SCARA command/Define simple contact check zone coordinate)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DFIF | Contact check zone number | Position number (2 successive positions are used) | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Set the data of two successive positions starting from the position number specified in operand 2, for the simple contact check zone definition coordinate data in operand 1.
The position data specified in operand 2 is set for simple contact check zone definition coordinate 1, while the data of the next position is set for definition coordinate 2. If the axis pattern does not match between the data of the two successive positions, "Error No. C30: Axis pattern error" occurs.
In MSEL-PCX/PGX, position data for four axes is set in the simple contact check zone coordinate data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the simple contact check zone coordinate data is set from is used as the dedicated data for the simple contact check zone coordinate set, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, Axis 4 in the position data set to the simple contact check zone coordinate in DFIF is not a target position for the additional linear axis.

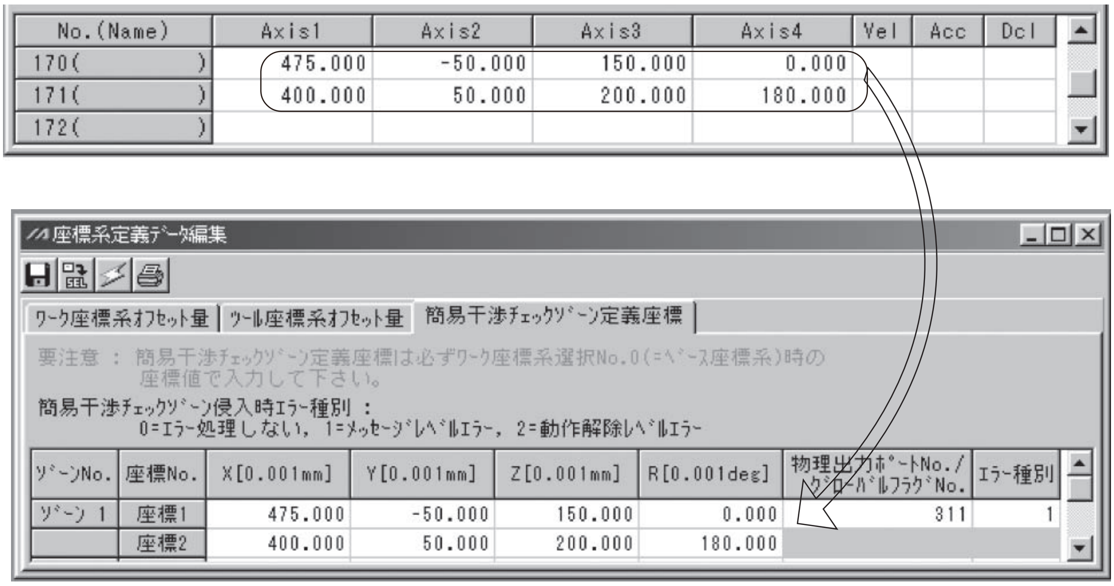
(Note 1) Simple contact check zone definition coordinates are always recognized as data on the base coordinate system (work coordinate system selection No. 0). If you are setting aside position data for use as effective definition coordinates for the DFIF command, you must set the data on the base coordinate system.

(Note 2) When the simple contact check zone definition coordinates are changed, it takes 5msec for the check result based on the new settings to be reflected.

(Note 3) The GRP command is invalid with respect to this command.

(Note 4) Indicate the position data effective either on SCARA axes (1st to 4th axes) or SCARA axes (5th to 8th axes) for the valid axes of the position data. "Error No. C30: Axis Pattern Error" will be issued when both of SCARA axes (1st to 4th axes) and SCARA axes (5th to 8th axes) are set effective for the valid axes of the position data.

[Example] DFIF 1 170




5. SEL Commands

## ● SOIF (Dedicated SCARA command/Specify output for simple contact check zone)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SOIF | Contact check zone number | Output/ global flag number | CP |

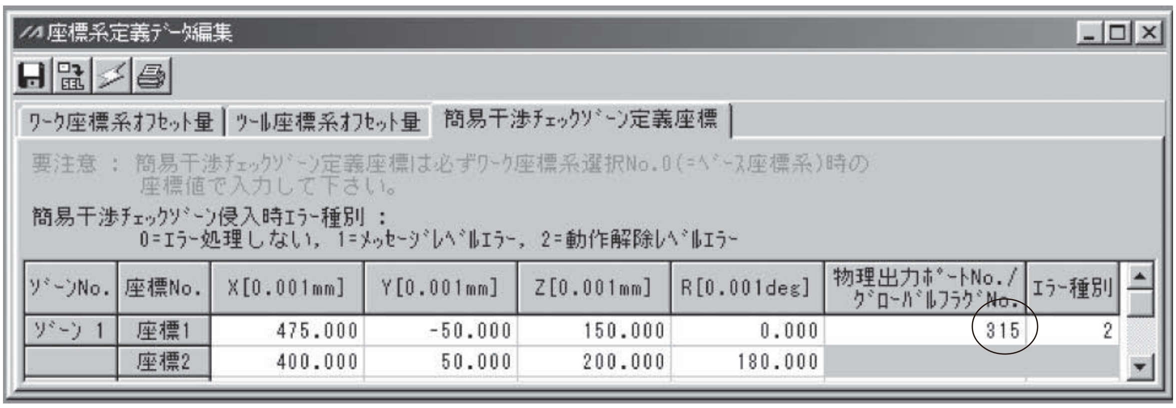
| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Set the output number/global flag number in operand 2 as the output specification to be applied upon entry into the simple contact check zone specified in operand 1.

(Note 1) The simple contact check zone is a function available for SCARA.

(Note 2) If duplicate physical output port numbers/global flag numbers are specified, chattering occurs and operation results become indeterminable.

[Example] SOIF 1 315



## ● SEIF (Dedicated SCARA command/Specify type of simple contact check zone)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SEIF | Contact check zone number | 0 or 1 or 2 (error type) | CP |

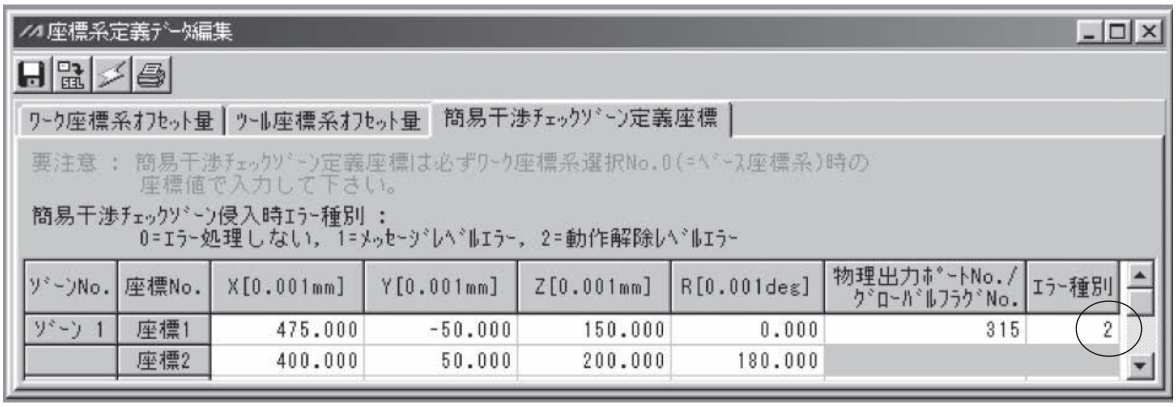
| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Specify the error type in operand 2 (see below) as the error type to be applied upon entry into the simple contact check zone specified in operand 1.

Error types applicable upon entry into simple contact check zone
0: No error
1: Message level error
2: Operation-cancellation level error

(Note 1) The simple contact check zone is a function available for SCARA.

[Example] SEIF 1 2



## ● GTIF (Dedicated SCARA command/Get simple contact check zone definition coordinate)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GTIF | Contact check zone number | Position number (2 successive positions are used) | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Set the simple contact check zone definition coordinate data in operand 1 for the data of two successive positions starting from the position number specified in operand 2.
Simple contact check zone definition coordinate 1 is set for the position data specified in operand 2, while definition coordinate 2 is set for the data of the next position. At this time, coordinate data in the position data becomes invalid for all axes, and then the simple contact check zone definition coordinate data is set.
In MSEL-PCX/PGX, simple contact check zone coordinate data for four axes is set in the position data no mater of the number of axes on SCARA Robot.
It is recommended that the position that the simple contact check zone coordinate is acquired from is used as the dedicated data for the simple contact check zone coordinate acquirement, not to be shared with the movement target position.
In case there is an additional linear axis is connected on Axis 4 on 3-axis type SCARA Robot, the simple contact check zone coordinate R-axis offset in the position data is written by execution of GTIF Command.

(Note 1) The position data of the invalid SCARA axes in the liner axes or the simple interference check zone definition coordinate data is cleared when the command is executed.

(Note 2) Simple contact check zone definition coordinate is always recognized as data on the base coordinate system (work coordinate system selection No. 0). Accordingly, the position data set by a GTIF command must be handled on the base coordinate system.

(Note 3) The GRP command is invalid with respect to this command.

## ● WGHT (Dedicated SCARA command/Set tip load mass, inertial moment)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | WGHT | Mass | (Inertial moment) | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | × | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

This command is supported by XSEL-PX/QX controller main application Ver.0.45 or later.
It is supported by PC software of Ver.7.5.0.0 or later and teaching pendants TB-02 (D): first edition or later, TB-01 (D): first edition or later SEL-T (D), of Ver.1.11 or later.
(Not applicable for IA-T-X(0))

(Note) Conventional models such as IX-NNN5020 cannot use this command. (A “D8A: Internal parameter error of acceleration/deceleration optimization or horizontal movement Z-position optimization function” occurs.)

[Function] Set the mass and inertial moment of the tip load (tool + work).
Set the mass in operand 1, and inertial moment in operand 2. The unit of operand 1 is [g], while the unit of operand 2 is [kg•mm$^2$].
The tip load mass/inertial moment set by a WGHT command will be retained until a new WGHT command is set again (= the set values will be retained even after the program ends). However, they are cleared when the power is turned OFF or a software reset is performed, after which you must set the applicable values again expressly in the program.

(Note 1) For the inertial moment in operand 2, set a composite inertial moment covering the tool and work relating to the center of rotation of the R-axis.

(Note 2) Although entry of inertial moment in operand 2 is optional, if no inertial moment is set the maximum allowable inertial moment of the robot is set automatically.

(Note 3) If the tip load mass exceeds the maximum loading capacity of the robot, a “B44: Load mass setting error” occurs.

(Note 4) Executing a WGHT command updates the information of both the tip load mass and inertial moment. You cannot change only the mass or only the inertial moment.

(Note 5) Although both the tip load mass and inertial moment can be approximate values, set values slightly larger than necessary. Before setting the values, round them up to the nearest multiple of 1g or 1kg•mm$^2$, respectively.

(Note 6) If a WGHT command has not yet been executed, the load mass and inertial moment have been initialized to the maximum loading capacity and maximum allowable inertial moment of the robot. Set an appropriate load mass and inertial moment according to the use conditions.

(Note 7) The load mass and inertial moment set by a WGHT command are used in the SCARA PTP acceleration/deceleration optimization function, SCARA horizontal movement Z-position optimization function, etc.

(Note 8) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

| | | | | |
|---|---|---|---|---|
| [Example 1] | GRP | 1111 | | It makes the 1st to 4th axes effective. |
| | WGHT | 2000 | 5000 | Set a tip load with 2000g of weight and 5000kgmm2 of the moment of inertia to the SCARA axes (1st to 4th axes). |

| | | | | |
|---|---|---|---|---|
| [Example 2] | GRP | 11111111 | | It makes the 1st to 8th axes effective. |
| | WGHT | 1000 | 30000 | Set SCARA axes (1st to 4th axes9 and SCARA axes (5th to 8th axes) so the movement is made to Position No. 2 to make it become the right arm system. |

## ● WGT2 (Dedicated SCARA command/Tip load condition setting)

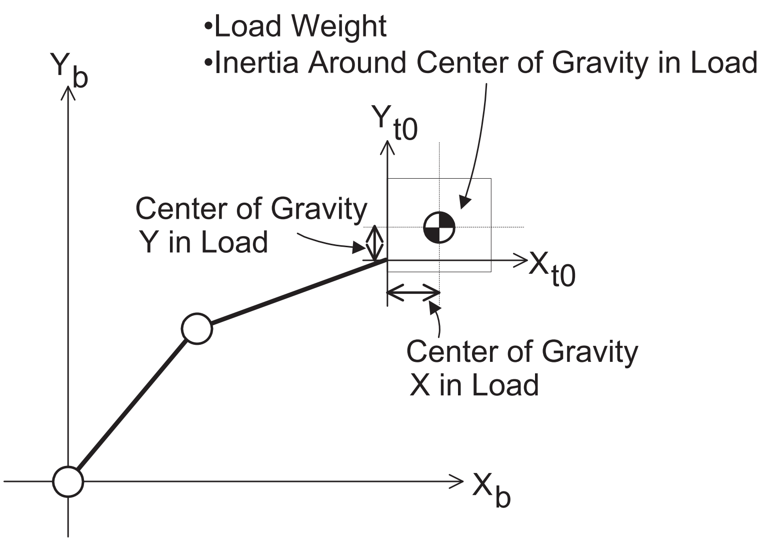
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | WGT2 | Mass | (Variable No.) | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | × | × | ○ | ○ | × | × | × |

[Function] Set the weight [gr] of the load on the tip (tool + work piece) in Operation 1, and the center of gravity in the load, inertia around the center of gravity and other related parameters to the six variables in a row from the indicated variable in Operation 2.
The tip load mass/inertial moment set by a WGT2 command will be retained until a new WGT2 command is set again. However, they are cleared when the power is turned OFF or a software reset is performed, after which you must set the applicable values again expressly in the program.

• Indicated Variable in Operation 2

| Variable No. | Contents of Setting | Remarks |
|---|---|---|
| n | Gravity Center X for Tip Load [1/1000mm unit] | Input the position at Tool Coordinate No. 0. |
| n+1 | Gravity Center Y for Tip Load [1/1000mm unit] | |
| n+2 | Inertia around Center of Gravity in Top Load [kgmm$^2$] | |
| n+3 | Set to 0 | Reservation (∗Possibility of use in future) |
| n+4 | Set to 0 | Reservation (∗Possibility of use in future) |
| n+5 | Set to 0 | Reservation (∗Possibility of use in future) |




5. SEL Commands

(Note 1) Inputting in Operation 2 is optional. When the setting in Operation 2 is not established, the parameters are the center of gravity in tip load X-Y = 0 and the maximum allowable moment of inertia.

(Note 2) An error will be issued when the tip load weight exceeds the maximum transportable weight of the robot.

(Note 3) When WGT2 Command is executed, the information for both the tip load weight and the moment of inertia is updated.
A change to individuals such as the weight only or center of gravity in tip load and inertia around the center of gravity only is not available.

(Note 4) Inappropriate setting of the robot tip load condition may cause vibration (abnormal noise) or error, and also may give an impact that shortens the mechanical life.
Establish the setting that reflects the actual mounted load.

(Note 5) For XSEL-RX/SX/RXD/SDX, XSEL-RAX/SAX/RAXD/SAXD Commands become effective even in WGT2 Command. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands. When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

[Example 1]

| | | | |
|---|---|---|---|
| GRP | 1111 | | Indicates SCARA of 1st to 4th axes |
| LET | 1001 | 50000 | Indicates Center of gravity X in tip load = 50.000mm |
| LET | 1002 | 0 | Indicates Center of gravity Y in tip load = 0.000mm |
| LET | 1003 | 2000 | Indicates inertia around center of gravity in load = 2000kg•mm$^2$ |
| WGT2 | 1000 | 1001 | For SCARA of 1st to 4th axes sets weight of 1000g and conditions of the tip load for Variable No. 1001 to 1003 |

[Example 2]

| | | | |
|---|---|---|---|
| GRP | 11110000 | | Indicates SCARA of 5th to 8th axes |
| LET | 1001 | 20000 | Indicates Center of gravity X in tip load = 20.000mm |
| LET | 1002 | 20000 | Indicates Center of gravity Y in tip load = 20.000mm |
| LET | 1003 | 500 | Indicates inertia around center of gravity in load = 500kg•mm$^2$ |
| WGT2 | 500 | 1001 | SCARA of 5th to 8th axes sets weight of 500g and conditions of the tip load for Variable No. 1001 to 1003 |

## ● NBND (Dedicated linear axis command/Set close distance)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | NBND | Axis pattern | Close distance | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| ○ | × | × | × | × | × | × | × | × | × | × |

[Function] Set in operand 2 the close distance (mm) from the target position based on the axis pattern specified in operand 1.
This command is valid only with respect to MOVP, MOVL and PATH commands in quick return mode 2 (closeness-detection return target position addition mode) or quick return mode 3 (closeness-detection return target position change mode).
A different value can be set for each axis.

(Note 1) The default value of 0 is applied if the close distance is not set with a NBND command.

(Note 2) In the case of PATH commands involving successive movements to multiple positions, the close distance becomes effective after the movement to the last position in the last movement is started and also after the processing of the previous position movement is completed. Accordingly, a dead width is created between (= at the overlap of) the movement to the last position in the PATH commands and the movement to the position immediately before it.

(Note 3) The close distance set here will remain effective even after the program ends. When building a system using NBND commands, therefore, specify the close distance expressly with a NBND command in all programs before any operation is started in each program. If you assume that the close distance will be reset after the end of operation in other programs, an unexpected close distance may be applied should the program abort due to an error, etc., in which case unforeseen problems may result.

(Note 4) Be sure to also refer to the pages that explain the QRTN command and PEND command.

(Note 5) Software versions supporting NBND
XSEL-J/K Controller main application: Ver.1.04 or later
(excluding flash ROM 8Mbit versions)
PC software: Ver.7.2.3.0 or later
Teaching pendant:
IA-T-X (D): Ver.1.44 or later
SEL-T (D): Ver.1.02 or later
TB-01 (D): First edition or later
TB-02 (D): First edition or later

| | | | | |
|---|---|---|---|---|
| [Example 1] | NBND | 11 | 50 | Set the close distance for axes 1 and 2 to 50mm after this command. |

[Example 2] The axis pattern can be specified indirectly using a variable. [Example 1] can be rephrased using indirect specification by variable as follows:

11 (binary) → 3 (decimal)

| | | | |
|---|---|---|---|
| LET | 1 | 3 | Assign 3 to variable 1. |
| NBND | *1 | 50 | Set the close distance for axes 1 and 2 to 50mm after this command. |

## [12] Actuator Control Command

### ● SV☐☐ (Turn ON/OFF servo)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SV☐☐ | Axis pattern | Prohibited | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Turn ON/OFF the servos of the axes specified by the axis pattern in operand 1.

SV☐☐
- ON ..................... Turn ON the servo.
- OF ..................... Turn OFF the servo.

(Other than SCARA robot)

[Example 1] SVON 11 — Turn ON the servos of axes 1 and 2. Nothing will occur if the axis servos are already ON.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET   1   3     Assign 3 to variable 1.
SVON  *1
```

(SCARA robots)

The arm system of SCARA axes (1st to 4th axes or 1st to 3rd axes) is set to Local Variable No. 99 when complete in normal condition.

Right arm system = 1
Left arm system = -1
Indeterminable = 0

The angle of arm 2 is used to make judgment.
The arm system effective immediately after the servo ON is set. The arm system is not monitored continuously.

(Note) The arm system data set in Local Variable No. 99 is the arm system for SCARA axes (1st to 4th axes or 1st to 3rd axes). To acquire the arm system data for SCARA axes (5th to 8th axes), use GARM Command.

## ● HOME (Dedicated linear axis command/Home return)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | HOME | Axis pattern | Prohibited | PE |

| Applicable models |
|---|
| XSEL-JX/KX × <br> Other than XSEL-JX/KX ○ |

[Function] Perform home return of the axes specified by the axis pattern in operand 1.
The servo of each home-return axis will turn ON automatically.
The output will turn OFF at the start of home return, and turn ON when the home return is completed.

(Note 1) This is a dedicated command for linear axes. If a SCARA axis (except for IXP Type Incremental specification) is specified, "Error No. B80: Specification-prohibited axis error" or "Error No. 421: SCARA/linear-axis simultaneous specification error" occurs.

(Note 2) Following a pause of home return, the operation will resume from the beginning of the home-return sequence.

(Note 3) Home-return operation for the axis using an ABS encoder makes a movement to the multi-rotation data reset position, thus it does not always make a movement to the home preset coordinate (including 0).
Use a MOVP command, instead of a HOME command, if you want to turn ON output 304 when I/O parameter No. 50, "Output function selection 304" is set to 1 (Output when all effective linear axes are home (= 0)) or 3 (Output when all effective linear axes are at home preset coordinate).

(Note 4) If an operation pause or cancel is performed during the HOME Command is executed for the axis using an ABS encoder other than the absolute reset mode provided by the PC software or teaching pendant, it may cause the "actual-position soft limit error" due to the position.
It is not recommended to perform home return other than for the purpose of adjusting an absolute-encoder axis.

[Example 1] HOME 11 Axes 1 and 2 return to the home.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET     1    3      Assign 3 to variable 1.
HOME    *1
```

## ● MOVP (Move PTP by specifying position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | MOVP | Position number | Prohibited | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Move the actuator to the position corresponding to the position number specified in operand 1, without interpolation (PTP stands for "Point-to-Point").
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

(Note) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 "SCARA/Linear Drive Axes Double Indication Error")
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Other than SCARA robots)

[Example 1]

```
VEL   100        Set the speed to 100mm/s.
MOVP  1          Move the axes to the position corresponding to
                 position No. 1 (200, 100).
```

[Example 2]

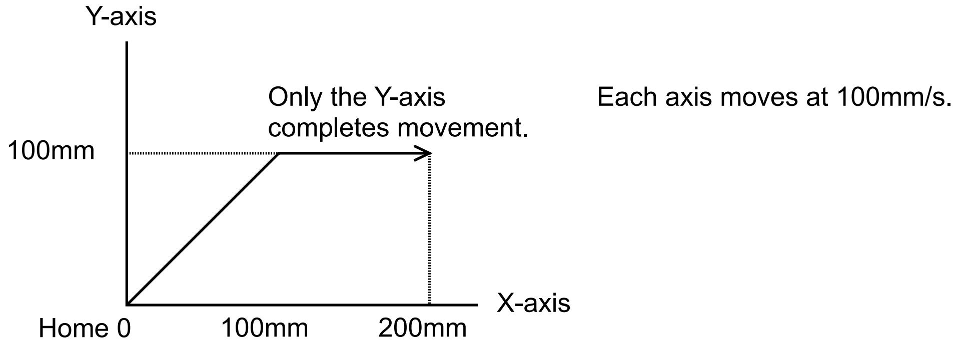
```
VEL   100        Set the speed to 100mm/s.
LET   1    2     Assign 2 to variable 1.
MOVP  *1         Move the axes to the position corresponding to the
                 content of variable 1 (position No. 2, or (100,
                 100)).
```

Position Data Display in PC Software

| No. | Axis 1 (X-axis) | Axis 2 (Y-axis) | Vel | Acc | Dcl |
|---|---|---|---|---|---|
| 1 | 200.000 | 100.000 | | | |
| 2 | 100.000 | 100.000 | | | |

(Note) If acceleration and deceleration are not specified by position data or ACC (DCL) commands, the actuator operates at the default values set in all-axis parameter No. 11, "Default acceleration" and all-axis parameter No. 12, "Default deceleration".

Travel path from the home to the position corresponding to position No. 1 (200, 100)



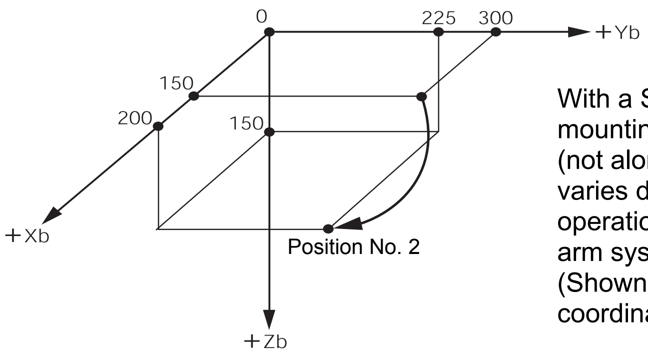
(SCARA robots)

[Example 1] MOVP 2 — Move the axes to the positions set under position No. 2 (200, 225, 150, 30).

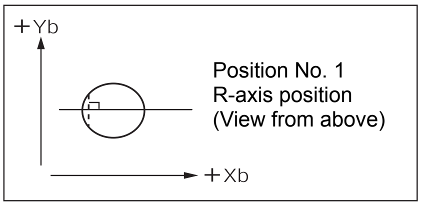
| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 1( ) | 150.000 | 300.000 | 0.000 | 0.000 | | | |
| 2( ) | 200.000 | 225.000 | 150.000 | 30.000 | | | |
| 3( ) | | | | | | | |
| 4( ) | | | | | | | |

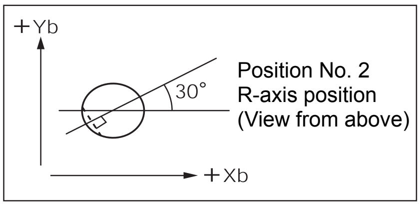
(Note) In the case of a SCARA axis, the axis operates according to all-axis parameter No. 47, “Default PTP acceleration for SCARA axis” or all-axis parameter No. 48, “Default PTP deceleration for SCARA axis” if the acceleration/deceleration is not specified using an ACCS (DCLS) command. In the case of a linear axis, the axis operates according to all-axis parameter No. 200, “Default acceleration for linear axis” or all-axis parameter No. 201, “Default deceleration for linear axis” if the acceleration/deceleration is not specified in the position data table or using an ACC (DCL) command.

Path of moving from position No. 1 to position No. 2




With a SCARA axis, the center of the tool mounting surface or tool tip moves via PTP (not along a straight line). The moving path varies depending on the start position of operation, completion position of operation, arm system, etc.
(Shown to the left are positions on the base coordinate system.)



## ● MOVL (Move by specifying position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | MOVL | Position number | Prohibited | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Move the actuator to the position corresponding to the position number specified in operand 1, with interpolation.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

(Note) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 "SCARA/Linear Drive Axes Double Indication Error")
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Other than SCARA robots)

[Example 1] VEL 100 — Set the speed to 100mm/s.
MOVL 1 — Move the axes to the position corresponding to position No. 1 (200, 100), with interpolation.

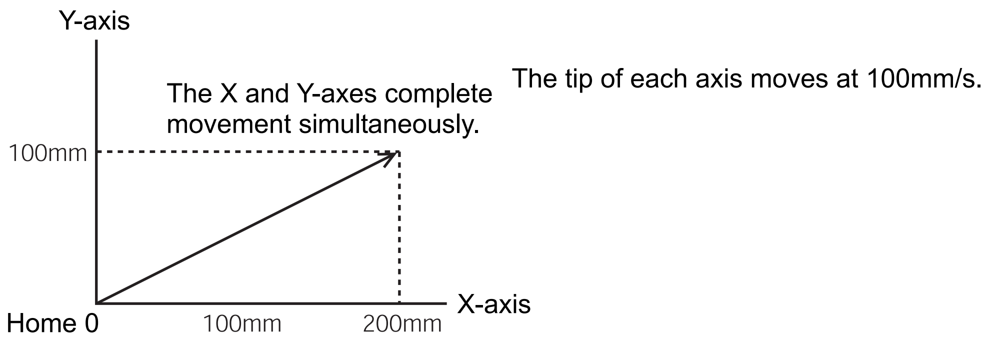
[Example 2] VEL 100 — Set the speed to 100mm/s.
LET 1 2 — Assign 2 to variable 1.
MOVL *1 — Move the axes to the position corresponding to the content of variable 1 (position No. 2, or (100, 100)), with interpolation.

Position Data Display in PC Software

| No. | Axis 1 (X-axis) | Axis 2 (Y-axis) | Vel | Acc | Dcl |
|---|---|---|---|---|---|
| 1 | 200.000 | 100.000 | | | |
| 2 | 100.000 | 100.000 | | | |

(Note) If acceleration and deceleration are not specified by position data or ACC (DCL) commands, the actuator operates at the default values set in all-axis parameter No. 11, "Default acceleration" and all-axis parameter No. 12, "Default deceleration".

Travel path from the home to the position corresponding to position No. 1 (200, 100)

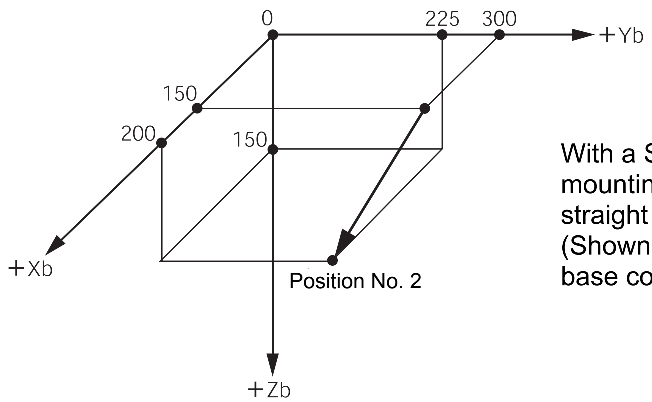


(SCARA robots)
[Example 1] MOVL 2 Move the axes to the positions set under position No. 2 (200, 225, 150, 30) via interpolation.

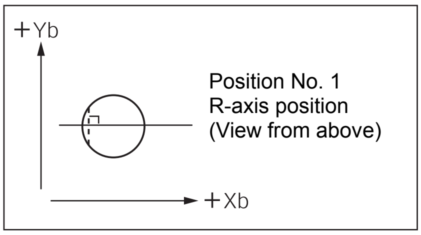
Path of moving from position No. 1 to position No. 2

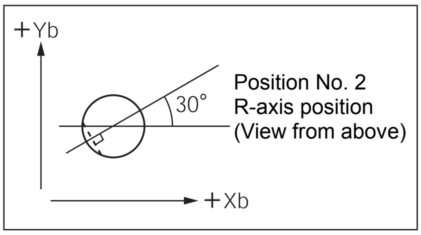
| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 1( ) | 150.000 | 300.000 | 0.000 | 0.000 | | | |
| 2( ) | 200.000 | 225.000 | 150.000 | 30.000 | | | |
| 3( ) | | | | | | | |
| 4( ) | | | | | | | |

(Note) In the case of a SCARA axis, the axis operates according to all-axis parameter No. 11, “Default CP acceleration for SCARA axis” or all-axis parameter No. 12, “Default CP deceleration for SCARA axis” if the acceleration/deceleration is not specified in the position data table or using an ACC (DCL) command.
In the case of a linear axis, the axis operates according to all-axis parameter No. 200, “Default acceleration for linear axis” or all-axis parameter No. 201, “Default deceleration for linear axis” if the acceleration/deceleration is not specified in the position data table or using an ACC (DCL) command.




With a SCARA axis, the center of the tool mounting surface or tool tip moves along a straight line.
(Shown to the left are positions on the base coordinate system.)



## ● MVPI (Move via incremental PTP)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | MVPI | Position number | Prohibited | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Move the actuator, without interpolation, from the current position by the travel distance corresponding to the position number specified in operand 1.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

(Note) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 “SCARA/Linear Drive Axes Double Indication Error”)
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Other than SCARA robots)

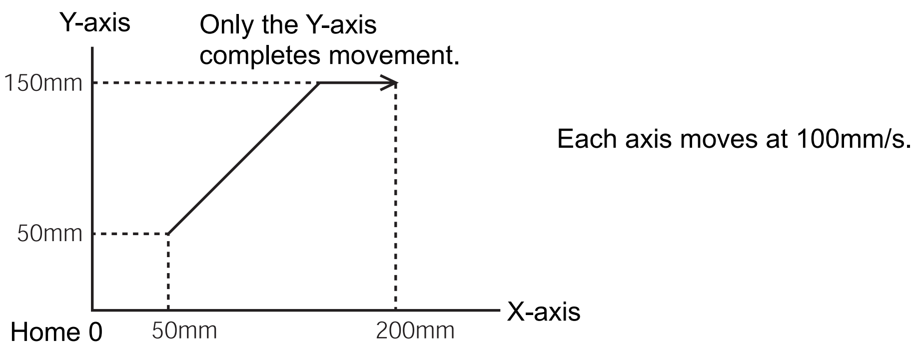
| | | | | |
|---|---|---|---|---|
| [Example 1] | VEL | 100 | | Set the speed to 100mm/s. |
| | MVPI | 1 | | If the current position is (50, 50) and position No. 1 is set to (150, 100), the axes will move 150 in the X direction and 100 in the Y direction (200, 150) from the current position. |
| [Example 2] | VEL | 100 | | Set the speed to 100mm/s. |
| | LET | 1 | 2 | Assign 2 to variable 1. |
| | MVPI | *1 | | Move from the current position by the travel distance corresponding to the content of variable 1 (position No. 2, or (100, 100)). |

Position Data Display in PC Software

| No. | Axis 1 (X-axis) | Axis 2 (Y-axis) | Vel | Acc | Dcl |
|---|---|---|---|---|---|
| 1 | 150.000 | 100.000 | | | |
| 2 | 100.000 | 100.000 | | | |

(Note) If acceleration and deceleration are not specified by position data or ACC (DCL) commands, the actuator operates at the default values set in all-axis parameter No. 11, “Default acceleration” and all-axis parameter No. 12, “Default deceleration”.

Travel path from (50, 50) by the travel distance corresponding to position No. 1 (150, 100)



(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse]
= (Screw lead [0.001mm] × Gear ratio numerator)
/ (Encoder resolution [pulses/rev] × Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse]
= Encoder resolution (0.001μm/pulse) × 1000
/ (2 ^ Encoder division ratio)
(Reference) Use the values of the following parameters for the above calculation formulas:

| | |
|---|---|
| Encoder resolution | : Axis-specific parameter No. 42 |
| Encoder division ratio | : Axis-specific parameter No. 43 |
| Screw lead | : Axis-specific parameter No. 47 |
| Gear ratio numerator | : Axis-specific parameter No. 50 |
| Gear ratio denominator | : Axis-specific parameter No. 51 |

(SCARA robots)
(Note 1) If an incremental movement command (MVPI, MVLI, TMPI or TMLI) is used repeatedly, coordinate conversion rounding errors, etc., will accumulate. To eliminate these errors, etc., execute an absolute movement command (MOVP, MOVL, etc.) once.

[Example 1] MVPI 6 Move from the current position by the travel according to position No. 6.
If the current positions of the axes are specified by position No. 5 (200, 150, 50, 45) and travels are specified by position No. 6 (15, 30, 20, 30), the axes move to the positions (215, 180, 70, 75).

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 5( ) | 200.000 | 150.000 | 50.000 | 45.000 | | | |
| 6( ) | 15.000 | 30.000 | 20.000 | 30.000 | | | |
| 7( ) | | | | | | | |

(Note) In the case of a SCARA axis, the axis operates according to all-axis parameter No. 47, “Default PTP acceleration for SCARA axis” or all-axis parameter No. 48, “Default PTP deceleration for SCARA axis” if the acceleration/deceleration is not specified using an ACCS (DCLS) command.
In the case of a linear axis, the axis operates according to all-axis parameter No. 200, “Default acceleration for linear axis” or all-axis parameter No. 201, “Default deceleration for linear axis” if the acceleration/deceleration is not specified in the position data table or using an ACC (DCL) command.

> ⚠ Caution
> A margin of error could accumulate between each pitch if the incremental (relative position indication) movement commands are repeated continuously.
> To avoid accumulation of errors, utilize the movement command to indicate the absolute position (MOVP Command).

## ● MVLI (Move via incremental interpolation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | MVLI | Position number | Prohibited | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Move the actuator, with interpolation, from the current position by the travel distance corresponding to the position number specified in operand 1.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.

(Note) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 "SCARA/Linear Drive Axes Double Indication Error")
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Other than SCARA robots)

[Example 1]

```
VEL   100      Set the speed to 100mm/s.
MVLI  1        If the current position is (50, 50) and position No. 1
               is set to (150, 100), the axes will move 150 in the X
               direction and 100 in the Y direction (200, 150) from
               the current position, with interpolation.
```

[Example 2]

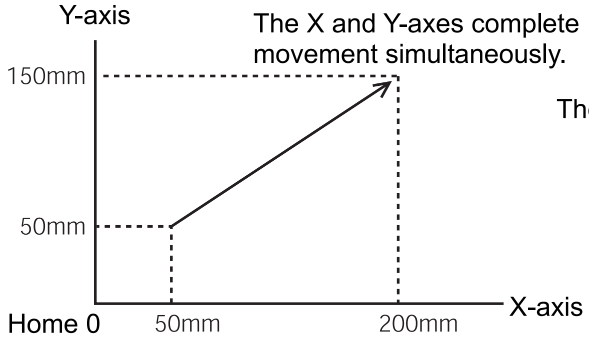
```
VEL   100      Set the speed to 100mm/s.
LET   1    2   Assign 2 to variable 1.
MVLI  *1       Move from the current position by the travel distance
               corresponding to the content of variable 1 (position
               No. 2, or (100, 100)).
```

Position Data Display in PC Software

| No. | Axis 1 (X-axis) | Axis 2 (Y-axis) | Vel | Acc | Dcl |
|---|---|---|---|---|---|
| 1 | 150.000 | 100.000 | | | |
| 2 | 100.000 | 100.000 | | | |

(Note) If acceleration and deceleration are not specified by position data or ACC (DCL) commands, the actuator operates at the default values set in all-axis parameter No. 11, "Default acceleration" and all-axis parameter No. 12, "Default deceleration".

Travel path from (50, 50) by the travel distance corresponding to position No. 1 (150, 100)



(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse]
= (Screw lead [0.001mm] × Gear ratio numerator)
/ (Encoder resolution [pulses/rev] × Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse]
= Encoder resolution [0.001μm/pulse] × 1000
/ (2 ^ Encoder division ratio)
(Reference) Use the values of the following parameters for the above calculation formulas:
Encoder resolution : Axis-specific parameter No. 42
Encoder division ratio : Axis-specific parameter No. 43
Screw lead : Axis-specific parameter No. 47
Gear ratio numerator : Axis-specific parameter No. 50
Gear ratio denominator : Axis-specific parameter No. 51

(SCARA robots)
(Note 1) If an incremental movement command (MVPI, MVLI, TMPI or TMLI) is used repeatedly, coordinate conversion rounding errors, etc., will accumulate. To eliminate these errors, etc., execute an absolute movement command (MOVP, MOVL, etc.) once.

[Example 1] MVLI 6 Move from the current position by the travel according to position No. 6.
If the current positions of the axes are specified by position No. 5 (200, 150, 50, 45) and travels are specified by position No. 6 (15, 30, 20, 30), the axes move to the positions (215, 180, 70, 75).

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 5( ) | 200.000 | 150.000 | 50.000 | 45.000 | | | |
| 6( ) | 15.000 | 30.000 | 20.000 | 30.000 | | | |
| 7( ) | | | | | | | |

(Note) In the case of a SCARA axis, the axis operates according to all-axis parameter No. 11, “Default CP acceleration for SCARA axis” or all-axis parameter No. 12, “Default CP deceleration for SCARA axis” if the acceleration/deceleration is not specified in the position data or using an ACC (DCL) command.
In the case of a linear axis, the axis operates according to all-axis parameter No. 200, “Default acceleration for linear axis” or all-axis parameter No. 201, “Default deceleration for linear axis” if the acceleration/deceleration is not specified in the position data or using an ACC (DCL) command.

⚠ Caution
A margin of error could accumulate between each pitch if the incremental (relative position indication) movement commands are repeated continuously.
To avoid accumulation of errors, utilize the movement command to indicate the absolute position (MOVL Command).

## ● MOVD (Move via direct value specification)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | MOVD | Target position | (Axis pattern) | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | × | × | × | × | ○ | × | ○ (PC/PG only) |



[Function] Move the axis specified by the axis pattern in operand 2, to the target position corresponding to the value specified in operand 1. If operand 2 is not specified, all axes will be moved.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.
The target position is set in mm, and the set value is valid to the third decimal place.

[Example 1] MOVD 100 10 Move axis 2 to position 100.

[Example 2] LET 1 100 Assign 100 to variable 1.
MOVD *1 11 Move all axes to the content of variable 1 (100).

## ● MVDI (Move relatively via direct value specification)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | MVDI | Travel distance | (Axis pattern) | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | × | × | × | × | ○ | × | ○ (PC/PG only) |

[Function] Move the axis specified by the axis pattern in operand 2 from its current position by the travel distance corresponding to the value specified in operand 1. If operand 2 is not specified, all axes will be moved.
The output will turn OFF at the start of axis movement, and turn ON when the movement is complete.
The travel distance is set in mm, and the set value is valid to the third decimal place.

(Note) If the specified travel distance is equal to or less than the travel distance per encoder pulse [mm/pulse], the axis may not move.
[Calculation formula of travel distance per encoder pulse]
Rotary encoder
Travel distance per encoder pulse [mm/pulse]
= (Screw lead [0.001mm] × Gear ratio numerator)
/ (Encoder resolution [pulses/rev] × Gear ratio denominator
/ (2 ^ Encoder division ratio)
Linear encoder
Travel distance per encoder pulse [mm/pulse]
= Encoder resolution [0.001μm/pulse] × 1000
/ (2 ^ Encoder division ratio)
(Reference) Use the values of the following parameters for the above calculation formulas:
Encoder resolution : Axis-specific parameter No. 42
Encoder division ratio : Axis-specific parameter No. 43
Screw lead : Axis-specific parameter No. 47
Gear ratio numerator : Axis-specific parameter No. 50
Gear ratio denominator : Axis-specific parameter No. 51

[Example 1]

| | | | |
|---|---|---|---|
| MVDI | 30 | 11 | Move all axes from the current position by 30mm in the positive direction. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | -100 | Assign -100 to variable 1. |
| MVDI | *1 | 1 | Move axis 1 from the current position in accordance with the content of variable 1 (-100), or by 100mm in the negative direction. |

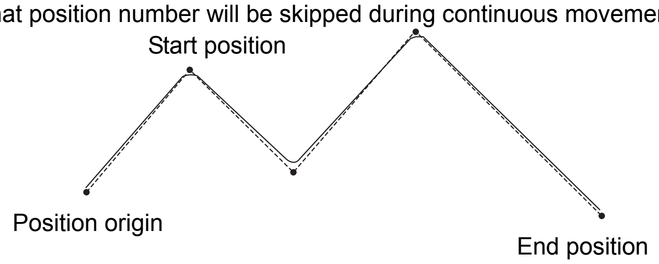
## ● PATH (Move along path via CP operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PATH | Start position number | End position number | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Move continuously from the position specified in operand 1 to the position specified in operand 2.
The output type in the output field can be set using an actuator-declaration command POTP.
Increasing the acceleration will make the passing points closer to the specified positions.
If invalid data is set for any position number between the start and end position numbers, that position number will be skipped during continuous movement.

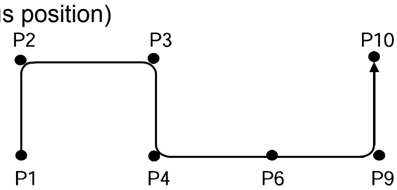



(Note 1) Multi-dimensional movement can be performed using a PATH command.
In this case, input in operand 1 the point number of the next target, instead of the predicted current position upon execution of the applicable command.
(Inputting a point number corresponding to the predicted current position will trigger movement to the same point during continuous movement, thereby causing the speed to drop.)

(Note 2) It is possible to move through discontinuous positions or move continuously by passing the same position.
As shown in the example, specify the number corresponding to the discontinuous position for both the start position number and end position number in the PATH command. In the example, this position is No. 6.

[Example] The actuator moves continuously in the sequence of position No. 1 → 2 → 3 → 4 → 6 → 9 → 10.

```
PATH   1   4
PATH   6   6 (discontinuous position)
PATH   9   10
```




(Note 3) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 “SCARA/Linear Drive Axes Double Indication Error”)
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example 1] PATH 100 120 Move continuously from position No. 100 to 120.

## ● CNTP (PTP Continuous Operation Mode Setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | CNTP | Other numbers | Prohibited | CP |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later.) |



[Function] It is the command to set PTP Continuous Operation Mode. By having PTP Continuous Operation Mode activated, the next PTP operation command can be started during deceleration of PTP operation command, which is expected to contribute to shorten time for movement. However, different from normal operation, continuous operation performs movement to pass through the vicinity of the target point. Also, the setting by this command would not give an impact to the continuous operation of CP movement command (Refer to [5.4.1 Continuous Movement Commands]).

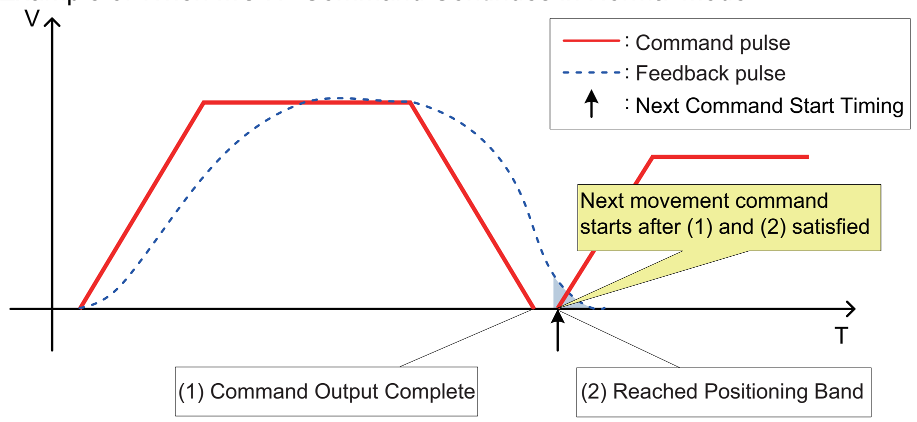
Continuous operation mode should be indicated by the value in Operation 1. The action of the case when PTP operation command is input to the continuous program steps should vary depending on the indicated mode.

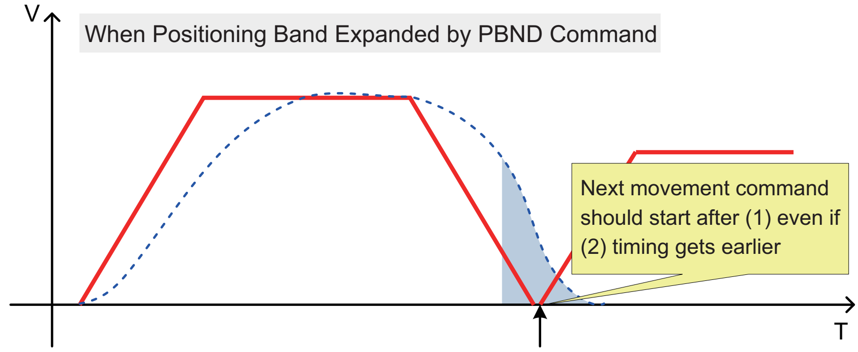
| Operation 1 | Mode Name | Summary |
|---|---|---|
| 0 | Normal mode | • PTP Continuous Operation: Inactivated<br>• Initial mode when CNTP Command not executed |
| 1 | PTP Continuous Operation Mode 1 | • PTP Continuous Operation: Activated<br>• Continuous operation while checking arrival to vicinity of target point (with positioning band width check) |
| 2 | PTP Continuous Operation Mode 2 | • PTP Continuous Operation: Activated<br>• Time saving highly expected (with no positioning band width check) |

The types of PTP operation command subject to continuous operation should be as shown below.
MOVP･MVPI･TMPI･PMVP･ARCH(*)･PACH(*)
(*: Arch Motion Command should be subject only when SCARA axes are operated.)

[0] Nomal mode (PTP Continuous Operation Inactivated) (Operation 1 = “0”)
PTP movement command would not perform continuous movement in a normal mode. The next PTP operation command should be output only after Condition (1) and (2) below are satisfied.
1) Command outputs to all the operation axes are complete
2) All the operation axes have got into the position band
Also, when this command is not executed in a program, the mode will be set to this mode automatically.

Example of When MOVP Command Continues in Normal Mode



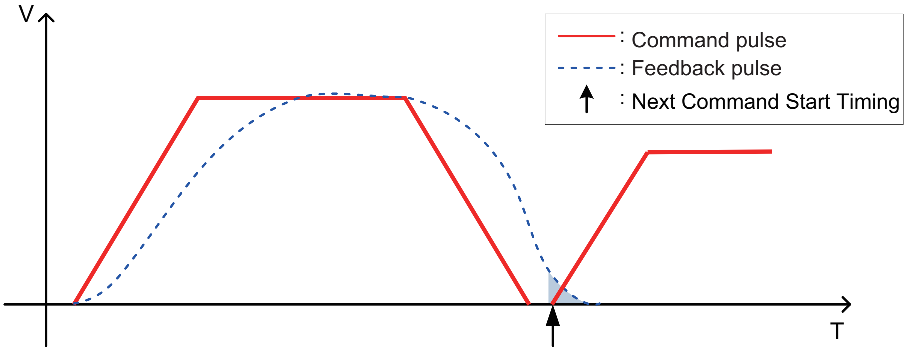



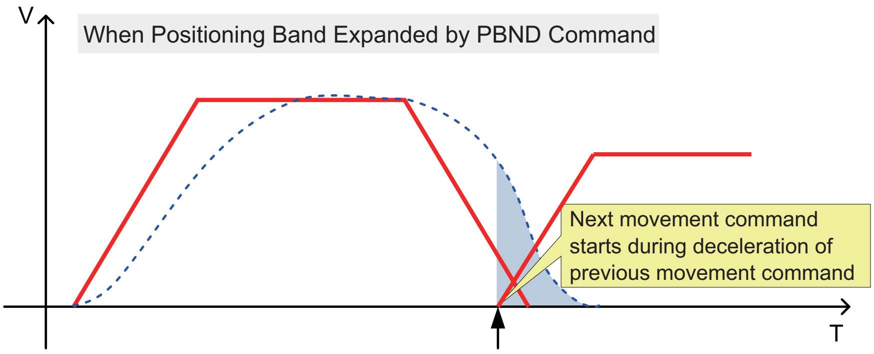

[1] PTP Continuous Operation Mode 1 (Operation 1 = “1”)
[2] PTP Continuous Operation Mode 2 (Operation 1 = “2”)
If PTP Continuous Operation Mode is indicated, the PTP operation command that is subject to should perform continuous operation.
The start condition of the next PTP operation command should differ depending on the mode that was indicated.

In Mode 1, the next movement command starts after all the axes are confirmed in the positioning band.
By changing the positioning band width by using PBND Command together, it is available to check if reached near the target position at the same time as attempting to save time. Also, even if having the positioning band width wide, the expectation of time shortening efficiency should not be as high as Mode 2. It is a mode suitable for time saving of such operations as a rough positioning operation, which requires a certain level of accuracy for positioning.

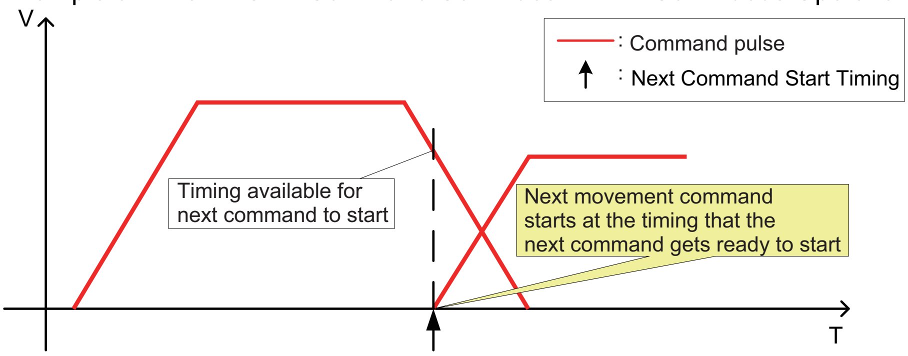
In Mode 2, the movement command in action and the next movement command get compared, and the next operation command should be started once it gets ready to start. Commands are added on top based on the information of movement commands, thus higher efficiency in time saving can be expected than Mode 1. However, there is a concern that the actuator may not reach the vicinity of the target point for each operation in some cases. It is a mode suitable for time saving in such operations as interference prevention that does not require accurate positioning.

Example of When MOVP Command Continues in PTP Continuous Operation Mode 1




Example of When MOVP Command Continues in PTP Continuous Operation Mode 2




(Note 1) PTP continuous operation mode indicated in this command is valid only in the program that the command was executed.
Also, the setting of the PTP continuous operation mode will be cancelled when the program ends.

(Note 2) When PTP continuous operation mode gets activated, the motion path should get different from that in a normal mode.

(Note 3) In case a command which is not subject to PTP continuous operation such as CP operation command (eg. PATH Command), output operation command (eg. BTON Command) and palletizing position number calculation command (eg. PINC Command) during a command subject to PTP continuous operation, PTP continuous operation will not be performed while this command is executed.

(Note 4) If PBND Command is to be used together, be aware that PBND Command setting will still be valid after the program finishes.
It is recommended basically that the value of the positioning band that has been changed is set back after the PTP continuous operation mode is finished when building a program. Also, for the positioning band of SCARA Robot J1/J2/R axes, the setting should be established in joint angles ([deg]). (For details, refer to the explanation for PBND Command.)

(Note 5) There is a tendency that high efficiency in time saving by continuous operation cannot be expected in such motions as stated below.
❑ Duration of acceleration and deceleration is short
- Acceleration and deceleration are high considering velocity
- An axis that operates only for a short distance is included in the operation axes
❑ An axis with a large deviation in operation is included in the operation axes

(Note 6) For a controller for SCARA Robot, it is not available to execute an arch motion command (ARCH/PACH) to a linear axis while PTP continuous operation mode activated. Error No. B80 "Indication Prohibited Axis Error" will be issued if attempted to execute. In the case above, it is necessary to get the setting back to the normal mode in order to execute an arch motion command.

(Note 7) Check also in the sections in "5.4.1 Continuous Movement Commands" for the details of the specifications related to the continuous operation.

[Example 1]

| | | | | |
|---|---|---|---|---|
| CNTP | 2 | | Indicate PTP Continuous Operation Mode 2. | |
| MOVP | 1 | | Perform PTP operation to position of Position No. 1. | PTP Continuous Operation |
| MOVP | 2 | | Perform PTP operation to position of Position No. 2. | |
| MOVP | 3 | | Perform PTP operation to position of Position No. 3. | |
| CNTP | 0 | | Put the setting back to normal mode. | |
| MOVP | 4 | | Perform PTP operation to position of Position No. 4. | |
| MOVP | 5 | | Perform PTP operation to position of Position No. 5. | |

[Example 2]

| | | | | |
|---|---|---|---|---|
| CNTP | 2 | | Indicate PTP Continuous Operation Mode 2. | |
| MOVP | 1 | | Perform PTP operation to position of Position No. 1. | |
| CIR2 | 10 | 11 | CP arc motion going through Position No. 10 and 11 should be performed. | |
| MOVP | 2 | | Perform PTP operation to position of Position No. 2. | PTP Continuous Operation |
| MOVP | 3 | | Perform PTP operation to position of Position No. 3. | |
| CNTP | 0 | | Put the setting back to normal mode. | |

[Example 3]

| | | | | |
|---|---|---|---|---|
| PBND | 11 | 5 | Set the width of the positioning band for 1st and 2nd axes to 5 [mm]. | |
| CNTP | 1 | | Indicate PTP Continuous Operation Mode 1. | |
| MOVP | 1 | | Perform PTP operation to position of Position No. 1. | PTP Continuous Operation |
| MOVP | 2 | | Perform PTP operation to position of Position No. 2. | |
| MOVP | 3 | | Perform PTP operation to position of Position No. 3.<br>* The next command will start after it is confirmed that the axis gets into the range within 5 [mm] from the target position. | |
| CNTP | 0 | | Put the setting back to normal mode. | |
| PBND | 11 | 0.1 | Set the width of the positioning band for 1st and 2nd axes to 0.1 [mm]. | |

## ● J☐W☐ (Jog)

(Note) When this command is to be used in MSEL-PC/PG or TTA, whether All Axes Parameter No. 55 (Coordinate System Definition 1 Control) and No. 56 (Coordinate System Definition 1 Construction Axis Setting) are set or not makes the operation different.

◎With No Setting

The operation will be the same as the existing linear axis specifications.

- There is no restriction to the number of indicated axes
- JOG operation is made on each axis system
- Execution is available with the home-return operation incomplete

◎With Setting Done

- Only one axis is available to be indicated.
- JOG operation is made on the work coordinate system.
- Execution is not available unless the home-return operation is completed.

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | J☐W☐ | Axis pattern | Input/output/ flag number | PE |

| Applicable models |
|---|
| XSEL-JX/KX ×<br>Other than XSEL-JX/KX ○ |

[Function] The axes in the axis pattern specified in operand 1 will move forward or backward while the input or output port or flag specified in operand 2 is ON or OFF.

JBWF .......... Move backward while the specified port is OFF.
JBWN.......... Move backward while the specified port is ON.
JFWF .......... Move forward while the specified port is OFF.
JFWN .......... Move forward while the specified port is ON.

With SCARA axes, coordinate system definition unit axis (refer to 1.4.6) only one axis (X, Y, Z or R-axis) can be specified.
If a SCARA axes, coordinate system definition unit axis is specified, operation (CP operation) based on the currently selected work coordinate system is applied. With linear axis, multiple axes can be specified.

(Note 1) In main application XSEL-PX/QX Ver.0.33 or older, “Error No. B80: Specification-prohibited axis error” or “Error No. 421: SCARA/linear-axis simultaneous specification error” occurs if a SCARA axis is specified.

(Note 2) With linear axes, this command is also effective on axes that have not yet performed home return. In this case, however, the maximum speed is limited to all-axis parameter No. 15, “Maximum jog speed before coordinate confirmation/home return”. Since the coordinate values have no meaning in this condition, pay due attention to prevent contact with the stroke end. With SCARA axes, operation by this command is disabled only before the confirmation of ABS coordinates.

(Note 3) The jog speed of a SCARA axis is limited by all-axis parameter No. 37, “Maximum SCARA-axis speed under J☐W☐ command” (default: 250mm/sec). This parameter can be edited using PC software Ver.7.0.11.0 or later, teaching pendant TB-02 (D): first edition or later, TB-01 (D): first edition or later, SEL-T (D): Ver.1.01, IA-T-X (D): Ver.1.44 or later. (Although the value set in the above parameter can also be changed using an older PC software or teaching pendant version, the parameter name is not displayed and the set value is indicated in hexadecimal notation.)

(Note 4) Axes other than the SCARA axes, coordinate system definition unit axis (refer to 1.4.6) Z-axis cannot be operated from other tasks while the X, Y or R-axis is jogging.

(Note 5) If the start position of operation of a SCARA axis is near the point at which arms 1 and 2 form a straight line (singular point), operation is performed at low acceleration to prevent sudden movement.

(Note 6) If the start position of operation of a SCARA axis is outside the work envelope (within the soft limit overt points of each axis, CP operation limit band, tool-reference-point entry prohibition circle (if tool offset is enabled) or back entry prohibition area), select an appropriate axis and direction and move the axis to inside the work envelope. Jogging out of the work envelope is not permitted.

(Note 7) If the R-axis generates “Error No. C74: Actual-position soft limit over error” due to a posture control component, etc., during SCARA-axis jog operation, take an appropriate action, such as bringing the R-axis position closer to the center of the R-axis stroke, using the jog function for each axis in the PC software or on the teaching pendant.

(Note 8) The R-axis JOG operation will be the rotational operation of the tool tip when it is set to tool offset valid (tool coordinate system select number ≠ 0) for SCARA axes, coordinate system definition unit axis (refer to 1.4.6).
Therefore, Arm 1 and 2 for SCARA axes and X and Y-axes for coordinate system definition unit axis will operate. Pay attention to this difference.

(Note 9) If the axis that moves according to J☐W☐ is a linear axis, and also if axis-specific parameter No. 1, “Axis operation type” is set to 0 (Linear movement axis) while axis-specific parameter No. 68, “Linear-axis linear movement mode selection” is set to 1 (Infinite stroke mode*), infinite stroke operation is performed.
During infinite stroke operation, the current position circulates within a range of approx. -10m to 10m.
Any positioning command to a position outside a coordinate range of approx. -9999 to +9990 generates “Error No. CBE: Target-value data boundary over error”.
If a positioning command not meeting the above condition is executed outside a coordinate range of approx. -9990 to +9990, “Error No. CC5: Positioning boundary breakout error” occurs.
(These errors are generated intentionally because the user cannot recognize the operating direction precisely around the boundary. If any of these errors occurs, axis-specific parameter No. 10, “ABS reset position movement/home return method” must be set to 1 (Current position 0 home) and, if necessary, the current value may also have to be reset with a HOME command.)
During infinite stroke operation, be sure to implement a timeout check using other task or external system.

| |
|---|
| The infinite-stroke mode can be specified only when an incremental encoder is used.<br>If you wish to use the infinite-stroke mode, contact IAI’s Sales Engineering. |

(Note 10) JOG operation in positive direction when the current position is approximately in the range from 9990 to 9999 and JOG operation in negative direction when the current position is approximately in the range from -9999 to -9990.
Move the actuator once to the opposite direction with JOG operation and get it out of the ranges described above, and then perform the JOG operation again.

(Note 11) JOG operation by this command cannot be performed for coordinate system definition unit axis (refer to 1.4.6) while in home-return incomplete condition. Also, number of axes available for operation indication is only one.

| | | | | |
|---|---|---|---|---|
| [Example 1] | VEL | 100 | | Set the speed to 100mm/s. |
| | JBWF | 10000 | 10 | Move axis 5 backward while input 10 is OFF. |

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
10000 (binary) → 16 (decimal)

| | | | |
|---|---|---|---|
| VEL | 100 | | Set the speed to 100mm/s. |
| LET | 1 | 16 | Assign 12 to variable 1. |
| JBWF | *1 | 10 | |

| | | | | |
|---|---|---|---|---|
| [Example 3] | VEL | 100 | | Set the speed to 100mm/s. |
| | LET | 5 | 20 | Assign 20 to variable 5. |
| | JFWN | 10000 | *5 | Move axis 5 forward while the content of variable 5 (input 20), is ON. |

## ● STOP (Stop movement)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | STOP | Axis pattern | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Decelerate and stop the axes specified by the axis pattern in operand 1.

(Note 1) A STOP command can be used with all active servo commands other than a SVOF command.

(Note 2) With a SCARA robots, all axes are decelerated to a stop regardless of the axis pattern.

(Note 3) The STOP command only issues a deceleration stop (operation cancellation) command and the program does not wait for completion of stopping. If other servo command is issued while the axes are stopping, the command becomes invalid or an "axis multiple-use" or other error occurs.
Set a timer, etc., in the program so that the next servo command will be issued after a sufficient deceleration-stop processing time elapses.
Even when a STOP command is to be issued to an axis currently stopped, provide a minimum interval of 0.1sec before the next servo command is issued.

(Other than SCARA robots)

[Example 1] STOP 11 Decelerate and stop axes 1 and 2.

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET   1   3      Assign 3 to variable 1.
STOP  *1
```

(SCARA robot)

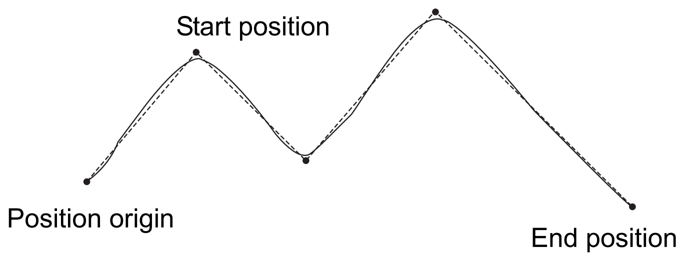
[Example 1] STOP 1 Decelerate the SCARA axes to a stop.

## ● PSPL (Move along spline via CP operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PSPL | Start position number | End position number | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Continuously move from the specified start position to end position via interpolation along a spline-interpolation curve.
The output type in the output field can be set using an actuator-declaration command POTP.
If invalid data is set for any position number between the start and end position numbers, that position number will be skipped during continuous movement.




(The diagram above is an image.)

(Note 1) If the acceleration and deceleration are different between points, the speeds will not be connected smoothly.

In this case, input in operand 1 the point number of the next target, instead of the predicted current position upon execution of the applicable command.
(Inputting a point number corresponding to the predicted current position will trigger movement to the same point during continuous movement, thereby causing the speed to drop.)

(Note 2) In XSEL-PX/QX, a movement to a position that indicates the target for SCARA axis and linear drive axis at the same time cannot be made. (421 "SCARA/Linear Drive Axes Double Indication Error")
Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

[Example] PSPL 100 120 Continuously move from position Nos. 100 to 120 along a spline-interpolation curve.

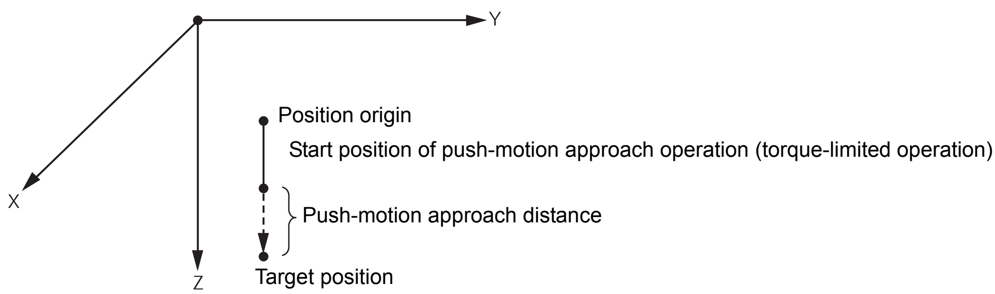
## ● PUSH (Move by push motion)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PUSH | Target position number | Prohibited | PE |

| Applicable models |
|---|
| MSEL-PCX/PGX × (Axes other than the power Con SCARA, such as the gripper is excluded.)<br>Other than above ○ |



[Function] Perform push-motion operation until the target position specified in operand 1 is reached.
The axes move in a normal mode from the position origin to the push-motion approach start position as determined by a PAPR command, after which push-motion approach operation (toque-limiting operation) will be performed. The speed of push-motion approach operation (toque-limiting operation) is determined by the push-motion approach speed specified by a PAPR command. If the output field is specified, the output will turn ON when a contact is confirmed, and turn OFF when a missed contact is detected.
Movement from the position origin to start position of push-motion approach conforms to the speed and acceleration/deceleration specified by VEL/ACC/DCL commands or in the position data table.




The pressing force can be adjusted in Driver Card Parameter No. 38 Limitation for pressing torque in positioning process (default value = 70%) or PTRQ Command for models other than XSEL-J/K, MSEL and TTA.
For XSEL-J/K, MSEL and TTA, the pressing force can be adjusted in Driver Card Parameter No. 33 Limitation for Pressing Torque in Positioning Process (default value = 70%).
For MSEL and TTA, it can also be adjusted with PTRQ Command.

(Note 1) A PUSH command only moves a single axis. PUSH Command moves only the Z-axis for SCARA Coordinates. PUSH Command is not available for IXP PowerCON SCARA. If multiple axes are specified, an “Error No. C91, Multiple push-axes specification error” will generate.
Even though indication of Z-axis is available for coordinate system definition unit axis (refer to 1.4.6), “Error No. C91 Indication of 2 or More Pressing Axes Error” could occur and operation could not be made depending on the offset of the selected coordinate system in X, Y and R-axes.
Indication is available only when R-axis offset of the selected work coordinate system for X and Y-axes is 0 for coordinate system definition unit axis (refer to 1.4.5). Also, indication is available only when both of X-axis offset and Y-axis offset in the selected tool coordinate system for R-axis is 0 for coordinate system definition unit axis.

(Note 2) A push-motion approach speed exceeding the maximum speed permitted by the system will be clamped at the maximum speed.
(The maximum system speed is not the maximum practical speed. Determine a practical speed by considering the impact upon contact, etc.)

(Note 3) When the pressing approaching distance is longer than the distance from the position start point to the target position, the pressing operation starts from the start at the position start point.

[Example]

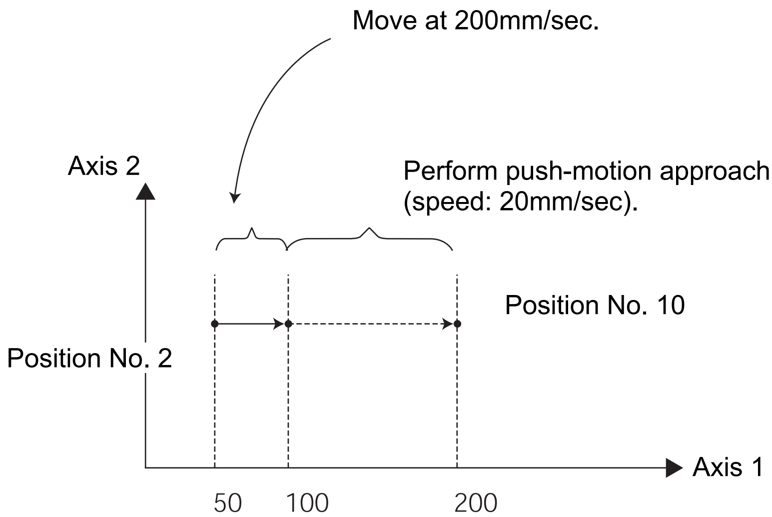
```
PAPR    100    20
MOVP    2
PUSH    10
```

Set the push-motion approach distance to 100mm and push-motion approach speed to 20mm/sec.
Move from the current position to position No. 2.
Perform push-motion movement from position No. 2 to 10.

The diagram below describes a push-motion movement based on the position data shown in the table below:

Position Data Display in PC Software

| Position No. | Axis 1 | Axis 2 | Vel | Acc | Dcl |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | 50.000 | 100.000 | | | |
| • | | | | | |
| • | | | | | |
| • | | | | | |
| • | | | | | |
| 10 | 200.000 | | 200 | 0.30 | 0.30 |
| • | | | | | |
| • | | | | | |



## ● PTRQ (Change push torque limit parameter)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTRQ | Axis pattern | Ratio | CC |

| Applicable models |
|---|
| XSEL-J/K × <br> Other than XSEL-J/K ○ |



[Function] Change the push torque limit parameter of the axis pattern specified in operand 1 to the value in operand 2. Operand 2 is set as an integer (unit: %).
A PTRQ command temporarily rewrites “Driver parameter No. 38: Push torque limit at positioning”.

(Note 1) If the pressing torque limit is not set by PTRQ Command, the value set in “Driver Card “Parameter No. 38 Limitation for Pressing Torque in Positioning Process” will be used for models other than XSEL-PC/PG ad TTA.
For XSEL-PC/PG and TTA, the value should be that in “Driver Card Parameter No. 33 Limitation for Pressing Torque in Positioning Process”.

(Note 2) The new push torque limit will remain effective even after the program ends. Therefore, when building a system using the PTRQ command, in every program explicitly specify a push torque limit using a PTRQ command before each push-motion operation. Assuming that the push torque limit will be reset to the original value when push-motion operation ends in one program can cause an unexpected problem in another program, because a different push torque limit will be used if the program is aborted due to an error, etc.

(Note 3) The new value set by a PTRQ command will become ineffective after a power ON reset or software reset.

(Note 4) A PTRQ command does not rewrite “Driver parameter No. 38: Push torque limit at positioning” (main CPU flash memory (non-volatile memory)).

[Example]

| | | | |
|---|---|---|---|
| PTRQ | 1 | 50 | Change the push torque limit parameter for axis 1 to 50%. |
| PAPR | 100 | 20 | Set the push-motion approach distance to 100mm and the push-motion approach speed to 20mm/sec. |
| MOVP | 2 | | Move to position No. 2. |
| PUSH | 10 | | Move by push motion from position No. 2 to position No. 10. |

## ● CIR2 (Move along circle via CP operation 2 (Arc interpolation))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CIR2 | Passing position 1 number | Passing position 2 number | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Move along a circle originating from the current position and passing positions 1 and 2, via arc interpolation.
The rotating direction of the circle is determined by the given position data.
The diagram below describes a CW (clockwise) movement. Reversing passing positions 1 and 2 will change the direction of movement to CCW (counterclockwise).

(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

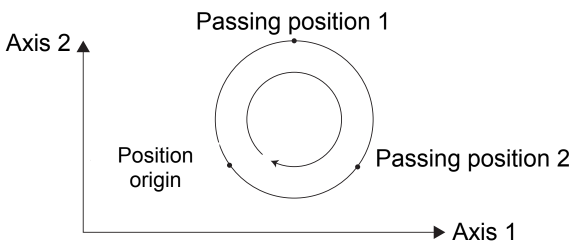
| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12) |

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | All-axis parameter No. 11, “Default acceleration for SCARA axis”<br>(All-axis parameter No. 12, “Default deceleration for SCARA axis”)<br>All-axis parameter No. 200, “Default acceleration for linear axis”<br>(All-axis parameter No. 201, “Default deceleration for linear axis”) |

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



(Note 1) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a CIR2 command is executed based on the position data of axes 2 and 3.

(Note 2) SCARA axes are available only on XY plane.

(Note 3) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, "Error No. C73: Target-path soft limit over error" may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

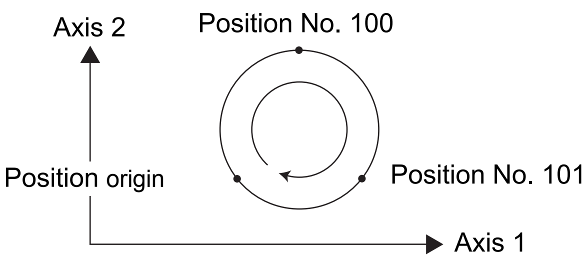
(Note 4) XSEL-PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 "Indication Prohibited Axes Error" or 421 "SCARA/Linear Drive Axes Double Indication Error" will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Note 5) The coordinate system definition unit axes set in ALL Individual Parameter No. 56 (Coordinate System Definition 1 Constructing Axes Setting) in TTA are valid on XY plane.
When Z-axis is indicated (XZ plane and YZ plane), "Error No. B80 Indication of Forbidden Axis Error" will occur unless the R-axis offset in the selected work coordinate system is 0.

(Note 6) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, such an operation is not available as to draw an arc by indicating axes other than the coordinate system definition unit axes at the same time. (B80 "Indication of Forbidden Axis Error" will occur.)
Either get rid of the axes other than the coordinate system definition unit axes by using GRP Command, or separate the position data for the coordinate system definition unit axes and other axes in order to make operation.

[Example]

| | | | |
|---|---|---|---|
| VEL | 100 | | Set the speed to 100mm/s. |
| CIR2 | 100 | 101 | Move along a circle (circular interpolation) passing position No. 100 and 101. |



## ● ARC2 (Move along circle via CP operation 2 (Arc interpolation))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ARC2 | Passing position number | End position number | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Move along an arc originating from the current position, passing the specified position and terminating at the end position, via arc interpolation.

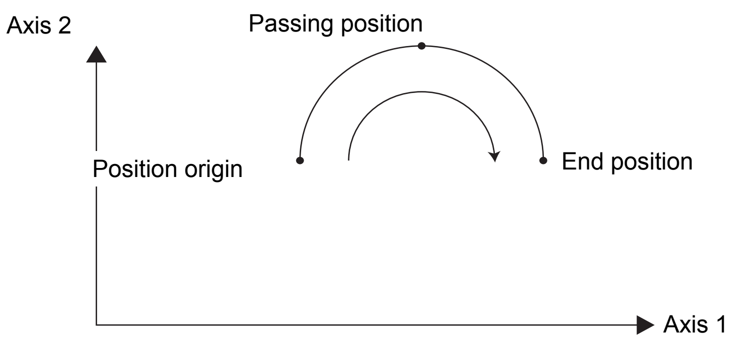
(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12) |

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”) |

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



(Note 1) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a ARC2 command is executed based on the position data of axes 2 and 3.

(Note 2) SCARA axes are available only on XY plane.

(Note 3) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, "Error No. C73: Target-path soft limit over error" may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

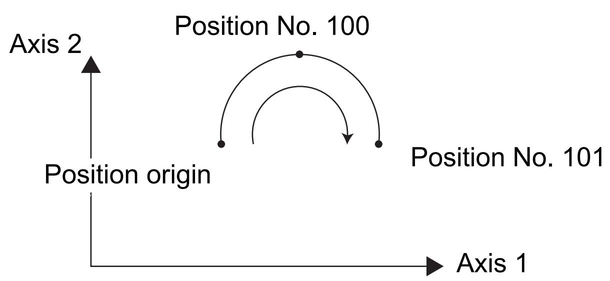
(Note 4) XSEL-PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 "Indication Prohibited Axes Error" or 421 "SCARA/Linear Drive Axes Double Indication Error" will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Note 5) The coordinate system definition unit axes set in ALL Individual Parameter No. 56 (Coordinate System Definition 1 Constructing Axes Setting) in TTA are valid on XY plane.
When Z-axis is indicated (XZ plane and YZ plane), "Error No. B80 Indication of Forbidden Axis Error" will occur unless the R-axis offset in the selected work coordinate system is 0.

(Note 6) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, such an operation is not available as to draw an arc by indicating axes other than the coordinate system definition unit axes at the same time. (B80 "Indication of Forbidden Axis Error" will occur.)
Either get rid of the axes other than the coordinate system definition unit axes by using GRP Command, or separate the position data for the coordinate system definition unit axes and other axes in order to make operation.

[Example]

| | | | |
|---|---|---|---|
| VEL | 100 | | Set the speed to 100mm/s. |
| ARC2 | 100 | 101 | Move along an arc (circular interpolation) from the current position to position No. 101 by passing position No. 100. |

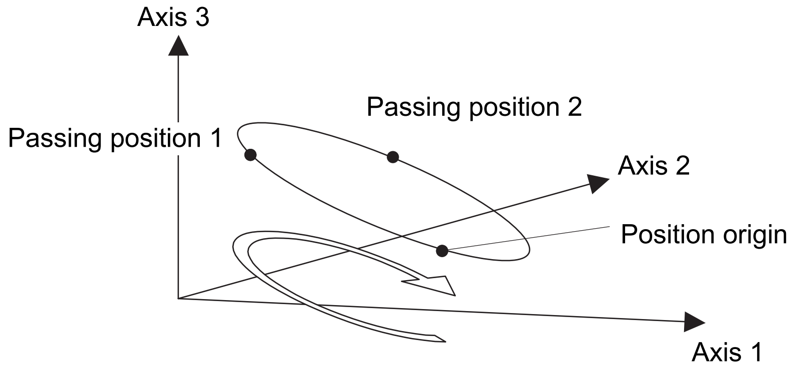


## ● CIRS (Move along circle three-dimensionally via CP operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CIRS | Passing position 1 number | Passing position 2 number | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Move along a circle by passing the passing positions 1 and 2 in this order, starting from the current position being the origin (three-dimensional movement). The direction in which to go around the circle is determined by the position data given. In the figure below, the rotating direction is reversed if passing positions 1 and 2 are swapped.




(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration | Deceleration |
|---|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 | Same as the effective value of acceleration |
| 2 | Setting value by VEL command | Setting value by ACC command | |
| 3 | | Default acceleration in all-axis parameter No. 11 | |

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”) |

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.

(Note 1) This command is valid on any planes in three-dimensional space. If four or more axes are set in the position data, three axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 5, for example, a CIRS command is executed based on the position data of axes 2 to 4.

(Note 2) The path tends to shift inward as the speed rises. Minor correction such as setting the position data slightly outward may be required.

(Note 3) If the diameter of the circle is smaller relative to the set speed, the speed may be limited.
(Although the extent to which the speed is limited can be reduced by raising the acceleration/acceleration, make sure the acceleration and deceleration do not exceed the range permitted by the actuator.)

(Note 4) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 5) XSEL-PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

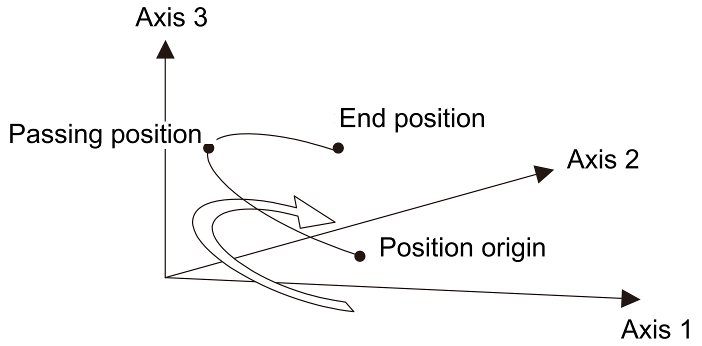
(Note 6) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, such an operation is not available as to draw an arc by indicating axes other than the coordinate system definition unit axes at the same time. (B80 “Indication of Forbidden Axis Error” will occur.)
Either get rid of the axes other than the coordinate system definition unit axes by using GRP Command, or separate the position data for the coordinate system definition unit axes and other axes in order to make operation.

## ● ARCS (Move along arc three-dimensionally via CP operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ARCS | Passing position number | End position number | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Move to the end position along an arc by passing the passing position, starting from the current position being the origin (three-dimensional movement).




(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration | Deceleration |
|---|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 | Same as the effective value of acceleration |
| 2 | Setting value by VEL command | Setting value by ACC command | |
| 3 | | Default acceleration in all-axis parameter No. 11 | |

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”) |

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.

(Note 1) This command is valid on any planes in three-dimensional space. If four or more axes are set in the position data, three axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 5, for example, a ARCS command is executed based on the position data of axes 2 to 4.

(Note 2) The path tends to shift inward as the speed rises. Minor correction such as setting the position data slightly outward may be required.

(Note 3) If the diameter of the circle is smaller relative to the set speed, the speed may be limited.
(Although the extent to which the speed is limited can be reduced by raising the acceleration/acceleration, make sure the acceleration and deceleration do not exceed the range permitted by the actuator.)

(Note 4) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 5) XSEL-PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Note 6) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, such an operation is not available as to draw an arc by indicating axes other than the coordinate system definition unit axes at the same time. (B80 “Indication of Forbidden Axis Error” will occur.)
Either get rid of the axes other than the coordinate system definition unit axes by using GRP Command, or separate the position data for the coordinate system definition unit axes and other axes in order to make operation.

## ● CHVL (Dedicated command for linear axis: Change speed)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CHVL | Axis pattern | Speed | CP |

| Applicable models |
|---|
| XSEL-JX/KX × <br> Other than XSEL-JX/KX ○ |

[Function] Change the speed of the axes operating in other task.
When a CHVL command is executed, the speed of the axes specified in operand 1 will change to the value specified in operand 2.

(Note 1) This command is not valid on an axis operated by a CIR, ARC, PSPL, PUSH, or ARCH command.

(Note 2) Executing a CHVL command for an axis operating in sigmoid motion (SCRV command) will generate an “Error No. CC1, Speed-change condition error”.

(Note 3) This is a temporary speed-change command issued from other task to the active packet (point). It is not affected by the data declared by VEL.

| Program 1 | Program 2 |
|---|---|
| | VEL 300 <br> • <br> • |
| | MOVP 1 |
| CHVL 11 100 | MOVP 2 |
| | MOVP 3 <br> • <br> • |

If CHVL is executed in program 1 while MOVP 2 is executed in program 2, the travel speed of MOVP 2 will become 100mm/sec.
The speeds of other move commands will remain 300mm/sec.

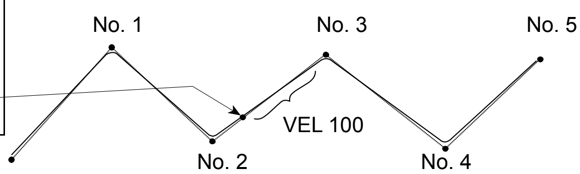
The axis pattern can be specified indirectly using a variable.
When program 1 is rephrased based on indirect specification using a variable:

11 (binary) → 3 (decimal)

| | | | |
|---|---|---|---|
| LET | 1 | 3 | Assign 3 to variable 1. |
| CHVL | *1 | 100 | |

(Note 4) Since this command is valid only for the packet that is active at the time of execution of the command for an axis subject to continuous motion in a PATH command, etc., caution must be exercised against the timing shift. The packet handling will be put on hold during speed-change processing, so caution must also be exercised against the locus shift.

| Program 1 | Program 2 |
|---|---|
| • <br> • <br> • | VEL 300 <br> • <br> • |
| CHVL 11 100 → | PATH 1 5 |
| • <br> • <br> • | |




If CHVL is executed in program 1 while PATH is executed in program 2, or specifically during the PATH movement from point No. 2 to point No. 3, the speed specified by CHVL (100mm/sec in the above example) will become valid only during the PATH movement to point No. 3. Other travel speeds will remain at the speed specified by VEL (300mm/sec in the above example).

(Note 5) Override of the CHVL call task will be applied, so caution must be exercised.

(Note 6) The maximum speed of the specified axis completing home return will be clamped by the minimum value set in “Axis-specific parameter No. 28, Maximum operating speed of each axis” or “Axis-specific parameter No. 27, Maximum speed limited by maximum motor speed” with respect to the specified axis and related interpolation axes currently operating. To prevent the maximum speed from being limited due to the effect of other axis whose maximum speed is lower than the speed specified in the CHVL command, issue a CHVL command in multiple steps corresponding to the respective axes having different maximum speeds. In particular, specification of a CHVL command in a separate step is recommended for a rotating axis.

(Note 7) This command is dedicated for the liner axes only. “Error No. 80 Indication Prohibited Axis Error” will be issued if the SCARA axes are indicated, or SCARA axes and the liner axes are indicated at the same time.

(Note 8) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, and satisfies the following conditions, the upper limit of the change velocity will be clamped at the velocity at the operation start.

- The subjected axis of the coordinate system definition unit is in CP operation together with R-axis and the tool offset of the coordinate system definition unit is valid.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| [Example] | CHVL | 11 | 500 | ⇒ | CHVL | 1 | 500 |
| | | | | | CHVL | 10 | 500 |

## ● ARCD (Move along arc via CP operation by specifying end position and center angle (Arc interpolation))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ARCD | End position number | Center angle | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Move along a circle originating from the current position and passing positions 1 and 2, via arc interpolation.
The rotating direction of the circle is determined by the given position data.
The diagram below describes a CW (clockwise) movement. Reversing passing positions 1 and 2 will change the direction of movement to CCW (counterclockwise).
The setting unit of the center angle is degree and the set value is effective to three decimal points.

(Note) The rotating direction of the actual operation locus may vary from the specified direction depending on how each axis is installed, how the two axes are combined, and so on. Perform test operation to check the rotating direction.
The setting unit of the center angle is degree and the set value is effective to three decimal points.

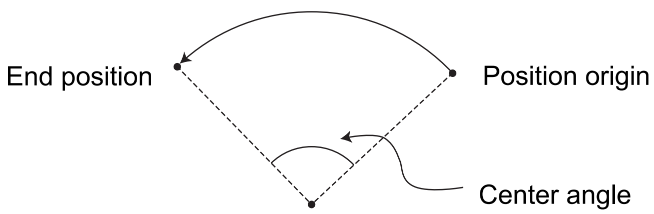
(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12) |

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | All-axis parameter No. 11, “Default acceleration for SCARA axis” (All-axis parameter No. 12, “Default deceleration for SCARA axis”) All-axis parameter No. 200, “Default acceleration for linear axis” (All-axis parameter No. 201, “Default deceleration for linear axis”) |

If speed is not set, a “C88 speed specification error” will generate.
If acceleration/deceleration is not valid, a “C89 acceleration/deceleration specification error” will generate.



(Note 1) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a ARCD command is executed based on the position data of axes 2 and 3.

(Note 2) SCARA axes are available only on XY plane.

(Note 3) If the center angle is small and the path is near a soft limit, "Error No. C73: Target-path soft limit over error" may occur.
In this case, move the path slightly inward from the soft limit boundary or make other appropriate correction. Also note that the larger the center angle, the smaller the path error becomes.

(Note 4) XSEL-PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 "Indication Prohibited Axes Error" or 421 "SCARA/Linear Drive Axes Double Indication Error" will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Note 5) The coordinate system definition unit axes set in ALL Individual Parameter No. 56 (Coordinate System Definition 1 Constructing Axes Setting) in TTA are valid on XY plane.
When Z-axis is indicated (XZ plane and YZ plane), "Error No. B80 Indication of Forbidden Axis Error" will occur unless the R-axis offset in the selected work coordinate system is 0.

(Note 6) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, such an operation is not available as to draw an arc by indicating axes other than the coordinate system definition unit axes at the same time. (B80 "Indication of Forbidden Axis Error" will occur.)
Either get rid of the axes other than the coordinate system definition unit axes by using GRP Command, or separate the position data for the coordinate system definition unit axes and other axes in order to make operation.

[Example]

| | | | | |
|---|---|---|---|---|
| | VEL | 100 | | Set the speed to 100mm/s. |
| | ARCD | 100 | 120 | Move along an arc from the position origin to position No. 100 for a center angle of 120° (CCW direction). |

## ● ARCC (Move along arc via CP operation by specifying center position and center angle (Arc interpolation))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ARCC | Center position number | Center angle | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Move along an arc originating from the current position by keeping a specified radius from the center position, via arc interpolation.
Specify the center position in operand 1, and the center angle formed by the position origin and end position in operand 2. The center angle is set in a range from -3600 to 3600° (±10 revolutions). A positive value indicates CCW (counterclockwise-direction) movement, while a negative value indicates CW (clockwise-direction) movement (setting unit: °(degree)).
The setting unit of the center angle is degree and the set value is effective to three decimal points.

(Note) The rotating direction of the actual operation locus may vary from the specified direction depending on how each axis is installed, how the two axes are combined, and so on. Perform test operation to check the rotating direction.
The setting unit of the center angle is degree and the set value is effective to three decimal points.

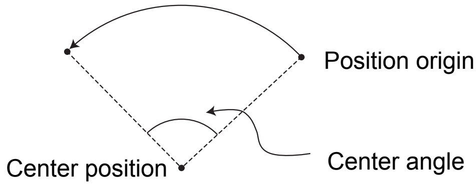
(Other than XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | Default acceleration in all-axis parameter No. 11 (Default deceleration in all-axis parameter No. 12) |

(XSEL-JX/KX/PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD and MSEL-PCX/PGX)
The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting value in the position data specified in operand 1 | Setting value in the position data specified in operand 1 |
| 2 | Setting value by VEL command | Setting value by ACC (DCL) command |
| 3 | | All-axis parameter No. 11, "Default acceleration for SCARA axis" (All-axis parameter No. 12, "Default deceleration for SCARA axis") All-axis parameter No. 200, "Default acceleration for linear axis" (All-axis parameter No. 201, "Default deceleration for linear axis") |

If speed is not set, a "C88 speed specification error" will generate.
If acceleration/deceleration is not valid, a "C89 acceleration/deceleration specification error" will generate.



(Note 1) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a ARCC command is executed based on the position data of axes 2 and 3.

(Note 2) SCARA axes are available only on XY plane.

(Note 3) If the center angle is small and the path is near a soft limit, “Error No. C73: Target-path soft limit over error” may occur.
In this case, move the path slightly inward from the soft limit boundary or make other appropriate correction. Also note that the larger the center angle, the smaller the path error becomes.

(Note 4) XSEL-PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 “Indication Prohibited Axes Error” or 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Note 5) The coordinate system definition unit axes set in ALL Individual Parameter No. 56 (Coordinate System Definition 1 Constructing Axes Setting) in TTA are valid on XY plane.
When Z-axis is indicated (XZ plane and YZ plane), “Error No. B80 Indication of Forbidden Axis Error” will occur unless the R-axis offset in the selected work coordinate system is 0.

(Note 6) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, such an operation is not available as to draw an arc by indicating axes other than the coordinate system definition unit axes at the same time. (B80 “Indication of Forbidden Axis Error” will occur.)
Either get rid of the axes other than the coordinate system definition unit axes by using GRP Command, or separate the position data for the coordinate system definition unit axes and other axes in order to make operation.

| [Example] | | | | |
|---|---|---|---|---|
| | VEL | 100 | | Set the speed to 100mm/s. |
| | ARCC | 100 | 120 | Move along an arc from the position origin for a center angle of 120° around position No. 100 being the center (CCW direction). |

## ● PBND (Set positioning width)

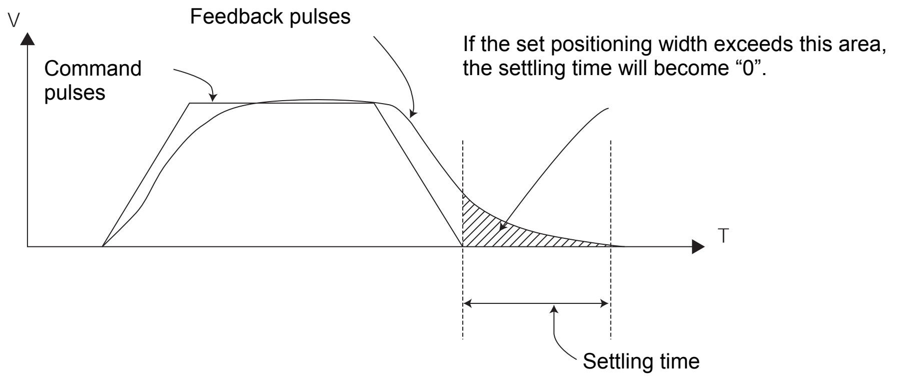
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PBND | Axis pattern | Distance | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the positioning complete width for the axes that correspond to the axis pattern specified in operand 2. The unit of operand 2 is as follows.

| | Unit of operand 2 |
|---|---|
| SCARA | X, Y, R: deg / Z: mm |
| Linear | mm / RS: deg |

[Function] As a rule, positioning is deemed complete when all command pulses have been sent and the current position is within the positioning complete width. Accordingly, this command provides an effective way to shorten the tact time by shortening the settling time after rough positioning. (Normally a desired effect can be achieved with approx. 3 to 5°, but you must check on the actual equipment.)




(Note 1) If positioning width is not set with a PBND command, the value set in “Axis-specific parameter No. 58, Positioning width” will be used.

(Note 2) If the positioning width is changed, the new setting will remain valid even after the program ends. Therefore, to build a system using PBND commands, a positioning band must be expressly specified with a PBND command before operation of each program. An assumption that the positioning width will be reset to the original value when the operation ends in other program may lead to an unexpected problem, because the positioning width will become different from what is anticipated in case the applicable program is aborted due to error, etc.

(Note 3) The value set in “Axis-specific parameter No. 58, Positioning width” will not be written by a PBND command.

[Example 1]

```
PBND    11    5        Set the positioning width for X-axis and Y-axis to 5°
                       after this command.
```

[Example 2] The axis pattern can be specified indirectly using a variable. When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET     1     3        Assign 3 to variable 1.
PBND    *1    5
```

## ● TMPI (Move incrementally to position on tool coordinate system via PTP operation)

(Note) When using this command in MSEL-PC/PG or TTA, set All Axes Parameter No. 55 and No. 56. (MSEL-PC/PG or TTA main application V2.00 or later)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TMPI | Position number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | ○ (TTA only) | ○ |

[Function] Move incrementally on the tool coordinate system without interpolation (= via PTP operation), by the travel from the current position corresponding to the position data in operand 1.

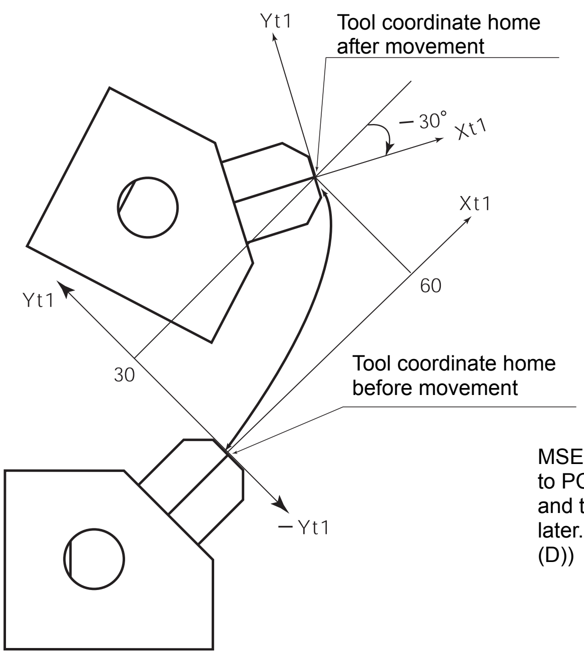
(Note 1) The tool/work coordinate systems are functions available for SCARA Robot. Also, this is a feature for the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56). Indicating other axes will cause “Error No. B80 Indication of Forbidden Axis Error”.

(Note 2) If an incremental movement command is used repeatedly, coordinate conversion rounding errors, etc., will accumulate.

[Example] TMPI 120

Position data

| No. | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 120 | 60.000 | 30.000 | 0.000 | -30.000 | | | |
| 121 | | | | | | | |
| 122 | | | | | | | |




MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-01 of Ver. 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D))

5. SEL Commands

## ● TMLI (Move incrementally to position on tool coordinate system via CP operation)

(Note) When using this command in MSEL-PC/PG or TTA, set All Axes Parameter No. 55 and No. 56. (MSEL-PC/PG or TTA main application V2.00 or later)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TMLI | Position number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | ○ (TTA only) | ○ |

[Function] Move incrementally on the tool coordinate system without interpolation (= via CP operation), by the travel from the current position corresponding to the position data in operand 1.

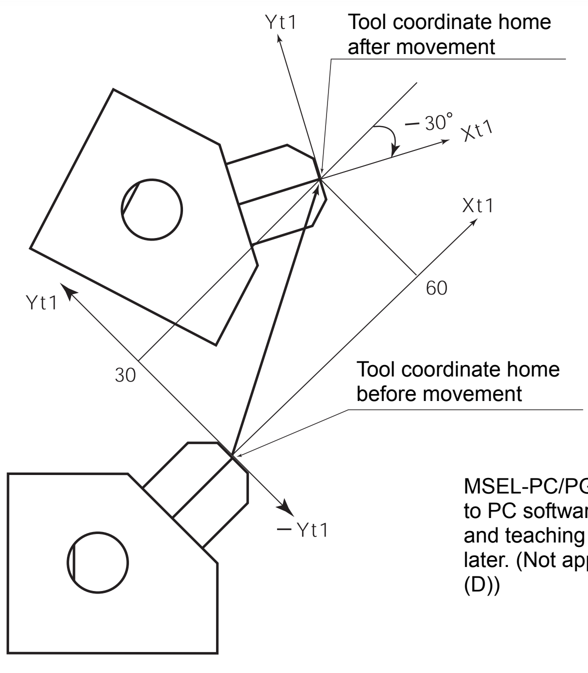
(Note 1) The tool/work coordinate systems are functions available for SCARA Robot. Also, this is a feature for the axes of the coordinate system definition unit of MSEL-PC/PG and TTA set in Coordinate System Definition 1 Constructing Axes Setting (All Axes Parameter No. 56). Indicating other axes will cause “Error No. B80 Indication of Forbidden Axis Error”.

(Note 2) If an incremental movement command is used repeatedly, coordinate conversion rounding errors, etc., will accumulate.

[Example] TMLI 120

Position data

| No. | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl |
|---|---|---|---|---|---|---|---|
| 120 | 60.000 | 30.000 | 0.000 | -30.000 | | | |
| 121 | | | | | | | |
| 122 | | | | | | | |




MSEL-PC/PG and TTA are available for input to PC software of Ver. 12.03.00.00 and later and teaching pendant TB-01 of Ver. 1.50 and later. (Not applicable for SEL-T (D) and IA-T-X (D))

## ● CIR (Move along circle via CP operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CIR | Passing position 1 number | Passing position 2 number | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Move along a circle originating from the current position and passing the positions specified in operands 1 and 2.
Therefore, reversing the settings of operands 1 and 2 will implement a circular movement in the reverse direction.
The output will turn OFF at the start of circular movement, and turn ON when the movement is complete.
Difference from CIR2:
CIR processing resembles moving along a polygon with a PATH command, while CIR2 actually performs arc interpolation.
Select an applicable command by considering the characteristics of each command. (Normally CIR2 is used.)

(Note 1) If the division angle is set to “0” with a DEG command (division angle is calculated automatically based on priority speed setting), the speed set in the data at passing position 1 or speed set by a VEL command will be used (former is given priority). The speed set in the data at passing position 2 will have no meaning.

(Note 2) If the division angle is set to a value other than “0” with a DEG command (normal division angle), the speed specified in the target position data will be used. (The speed set by a VEL command will become valid if position data is not specified.)
In the case of circular movement, the axes will return from passing position 2 to the start position at the speed declared by a VEL command.
Therefore, a VEL command must always be used with a CIR command.

(Note 3) The acceleration is selected in the order of the acceleration in the data at passing position 1, followed by the value in “All-axis parameter No. 11, Default acceleration”.
The deceleration will become the same value as the valid acceleration selected above. Therefore, the deceleration in the data at passing position 1 and the acceleration/deceleration in the data at passing position 2 will not have any meaning.

(Note 4) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a CIR command is executed based on the position data of axes 2 and 3.

(Note 5) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, "Error No. C73: Target-path soft limit over error" may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 6) XSEL-PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 "Indication Prohibited Axes Error" or 421 "SCARA/Linear Drive Axes Double Indication Error" will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Note 7) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, such an operation is not available as to draw an arc by indicating axes other than the coordinate system definition unit axes at the same time. (B80 "Indication of Forbidden Axis Error" will occur.)
Either get rid of the axes other than the coordinate system definition unit axes by using GRP Command, or separate the position data for the coordinate system definition unit axes and other axes in order to make operation.

| | | | | |
|---|---|---|---|---|
| [Example 1] | VEL | 100 | | Set the speed to 100mm/s. |
| | CIR | 100 | 101 | Move along a circle from the current position by passing positions 100 and 101 sequentially. |

| | | | | |
|---|---|---|---|---|
| [Example 2] | VEL | 100 | | Set the speed to 100mm/s. |
| | LET | 1 | 5 | Assign 5 to variable 1. |
| | LET | 2 | 6 | Assign 6 to variable 2. |
| | CIR | *1 | *2 | Move along a circle from the current position by passing the contents of variables 1 and 2 (positions 5 and 6) sequentially. |

## ● ARC (Move along arc via CP operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ARC | Passing position number | End position number | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Move along an arc from the current position to the position specified in operand 2, by passing the position specified in operand 1.
The output will turn OFF at the start of arc movement, and turn ON when the movement is complete.
Difference from ARC2:
ARC processing resembles moving along a polygon with a PATH command, while ARC2 actually performs arc interpolation.
Select an applicable command by considering the characteristics of each command. (Normally ARC2 is used.)

(Note 1) If the division angle is set to "0" with a DEG command (division angle is calculated automatically based on priority speed setting), the speed set in the data at passing position 1 or speed set by a VEL command will be used (former is given priority). The speed set in the data at passing position 2 will have no meaning.

(Note 2) If the division angle is set to a value other than "0" with a DEG command (normal division angle), the speed specified in the target position data will be used. (The speed set by a VEL command will become valid if position data is not specified.)

(Note 3) The acceleration is selected in the order of passing position 1 data, ACC command, and all-axis parameter No. 11, "Default acceleration for SCARA axis" or all-axis parameter No. 200, "Default acceleration for linear axis".
The deceleration will become the same value as the valid acceleration selected above. Therefore, the deceleration in the data at passing position 1 and the acceleration/deceleration in the data at passing position 2 will not have any meaning.

(Note 4) With rectangular actuators, this command is valid on any rectangular planes. If three or more axes are set in the position data, two axes are selected automatically from the axes that have been set, starting from the axis of the youngest number.
If position data is set for axes 2 to 4, for example, a ARC command is executed based on the position data of axes 2 and 3.

(Note 5) If the distance between the position origin and passing position 1 or between passing position 1 and passing position 2 is small and the path is near a soft limit, "Error No. C73: Target-path soft limit over error" may occur.
In this case, increase the distance between the adjacent positions as much as possible, move the path slightly inward from the soft limit boundary, or make other appropriate correction.

(Note 6) XSEL-PX/QX/RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD cannot make a movement to draw an arch using the SCARA axes and liner axes, or using the SCARA axes (axes 1 to 4) and SCARA axes (axes 5 to 8). Either B80 "Indication Prohibited Axes Error" or 421 "SCARA/Linear Drive Axes Double Indication Error" will occur. Use GRP Command, or operate the position data of SCARA axis and linear drive axis separately.

(Note 7) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, such an operation is not available as to draw an arc by indicating axes other than the coordinate system definition unit axes at the same time. (B80 "Indication of Forbidden Axis Error" will occur.)
Either get rid of the axes other than the coordinate system definition unit axes by using GRP Command, or separate the position data for the coordinate system definition unit axes and other axes in order to make operation.

| | | | | |
|---|---|---|---|---|
| [Example 1] | VEL | 100 | | Set the speed to 100mm/s. |
| | ARC | 100 | 101 | Move along an arc from the current position to position 101 by passing position 100. |

| | | | | |
|---|---|---|---|---|
| [Example 2] | VEL | 100 | | Set the speed to 100mm/s. |
| | LET | 1 | 5 | Assign 5 to variable 1. |
| | LET | 2 | 6 | Assign 6 to variable 2. |
| | ARC | *1 | *2 | Move along an arc from the current position to the content of variable 2 (position 6) by passing the content of variable 1 (position 5). |

## ● PEND (Wait for end of operation by axes currently used by program)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PEND | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| ○ | × | × | × | × | × | × | × | × | × | × |



[Function] When a PEND command is executed, the program waits for the end of operation by the axes it is currently using. The output turns ON only when a MOVP, MOVL or PATH command has been successfully executed (positioning has been successful) in quick return mode 2 (closeness-detection return target position addition mode) or quick return mode 3 (closeness-detection return target position change mode) (the output does not turn ON if any other servo command is executed).

(Note 1) To check if the operation has been successful (positioning has been successful), execute a PEND command before the quick return mode is cancelled.

(Note 2) Be sure to also refer to the pages that explain the QRTN command and NBND command.

(Note 3) Software versions supporting PEND
XSEL-J/K controller main application: Ver.1.04 or later
(excluding flash ROM 8Mbit versions)

PC software: Ver.7.2.3.0 or later
Teaching pendant:
IA-T-X (D): Ver.1.44 or later
SEL-T (D): Ver.1.02 or later
TB-01 (D): First edition or later
TB-02 (D): First edition or later

## [13] IF structure

### ● IF☐☐ (Structural IF)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | IF☐☐ | Variable number | Data | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Compare the content of the variable specified in operand 1 with the value specified in operand 2, and proceed to the next step if the condition is satisfied.
If the condition is not satisfied, the program will proceed to the step next to the corresponding ELSE command, if any, or to the step next to the corresponding EDIF command.
If the input condition is not satisfied and the IF☐☐ command is not executed, the program will proceed to the step next to the corresponding EDIF.
A maximum of 15 nests are supported when IS☐☐ and DW☐☐ are combined.

IF☐☐

- EQ .......... Operand 1 = Operand 2
- NE .......... Operand 1 ≠ Operand 2
- GT .......... Operand 1 > Operand 2
- GE .......... Operand 1 ≥ Operand 2
- LT .......... Operand 1 < Operand 2
- LE .......... Operand 1 ≤ Operand 2

[Example 1]

```
        SVON   1111                Set the current arm system in variable 99.
        PRDQ   1     100           Read the current X coordinate value into
                                   variable 100.
        CPNE   99    0     600     Turn OFF flag 600 if the arm system is
                                   indeterminable.
  600   IFEQ   99    1             Determine the arm system. The
                                   processing ends if the arm system is
                                   indeterminable.
        IFGE   100   0             Move to position No. 1 via PTP if the X
        MOVP   1                   coordinate value is 0 or greater.
        ELSE
        MOVP   2                   Move to position No. 2 via PTP.
        EDIF
        ELSE
        IFGE   100   0             Move to position No. 3 via PTP if the X
        MOVP   3                   coordinate value is 0 or greater.
        ELSE
        MOVP   4                   Move to position No. 4 via PTP.
        EDIF
        EDIF
        EXIT
```

If the current arm system is the right arm and X coordinate is 0 or greater, the axis moves to position No. 1. If the X coordinate is smaller than 0, it moves to position No. 2. If the left arm system is currently used and X coordinate is 0 or greater, the axis moves to position No. 3. If the X coordinate is smaller than 0, it moves to position No. 4.

(Note) Using a GOTO command to branch out of or into an IF☐☐-EDIF syntax is prohibited.

## ● IS□□ (Compare strings)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | IS□□ | Column number | Column number, character literal | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Compare the character strings in the columns specified in operands 1 and 2, and proceed to the next step if the condition is satisfied.
If the condition is not satisfied, the program will proceed to the step next to the corresponding ELSE command, if any, or to the step next to the corresponding EDIF command.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.
If the input condition is not satisfied and the IS□□ command is not executed, the program will proceed to the step next to the EDIF.
A maximum of 15 nests are supported when IF□□ and DW□□ are combined.

```
IS□□
  ├ EQ  .............  Operand 1 = Operand 2
  └ NE  .............  Operand 1 ≠ Operand 2
```

[Example 1]

```
        SCPY   10   'GOFD'
                    (Move forward)
        SCPY   14   'GOBK'
                    (Move backward)
        SLEN   4                      Set the number of comparing
                                      characters to 4.
  600   ISEQ   1    'XAXS' (X-axis)   Select an axis.
        ISEQ   5    10                Select a moving direction.
        MOVL   1                      Move to position 1 via CP.
        ELSE
        MOVL   2                      Move to position 2 via CP.
        EDIF
        ELSE
        ISNE   5    14                Select a moving direction.
        MOVL   3                      Move to position 3 via CP.
        ELSE
        MOVL   4                      Move to position 4 via CP.
        EDIF
        EDIF
```

CP operation is performed based on position No. 1 and 2 selected in columns 1 to 4, or position No. 3 and 4 selected in columns 5 to 8.
Nothing will happen if flag 600 is OFF, in which case the program will proceed to the step next to the last EDIF.
If the following data is stored in columns 1 to 8, CP movement to position No. 1 occurs.

| 1 2 | 3 4 | 5 6 | 7 8 | |
|---|---|---|---|---|
| XA | XS | GO | FD | |

## ● ELSE (Declare execution destination when IF command condition is not satisfied)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | ELSE | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] An ELSE command is used arbitrarily in conjunction with an IF☐☐ or IS☐☐ command to declare the command part to be executed when the condition is not satisfied.

[Example 1] Refer to the sections on IF☐☐ and IS☐☐.

## ● EDIF (End IF□□)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | EDIF | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Declare the end of an IF□□ or IS□□ command.

[Example 1] Refer to the sections on IF□□ and IS□□.

## [14] Structural DO

### ● DW☐☐ (DO WHILE)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | DW☐☐ | Variable number | Data | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Compare the content of the variable specified in operand 1 with the value specified in operand 2, and execute the subsequent commands up to EDDO while the condition is satisfied.
The program will proceed to the step next to the corresponding EDDO if the condition is no longer satisfied.
A LEAV command can be used to forcibly end a loop.
If the input condition is not satisfied and the DW☐☐ command is not executed, the program will proceed to the step next to the corresponding EDDO.
A maximum of 15 nests are supported when IF☐☐ and IS☐☐ are combined.

DW☐☐
- EQ ............ Operand 1 = Operand 2
- NE ............ Operand 1 ≠ Operand 2
- GT ............ Operand 1 > Operand 2
- GE ............ Operand 1 ≥ Operand 2
- LT ............ Operand 1 < Operand 2
- LE ............ Operand 1 ≤ Operand 2

[Example 1]

```
008    DWEQ    1    0
         :
         :
       EDDO
```

Repeat the command up to an EDDO command while variable 1 contains “0”.

If DW☐☐ is specified at the start and input 8 is OFF, nothing will occur and the program will proceed to the step next to EDDO.

(Note) Using a GOTO command to branch out of or into a DW☐☐-EDDO syntax is prohibited.

## ● LEAV (Pull out of DO WHILE)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | LEAV | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Pull out of a DO☐☐ loop and proceed to the step next to EDDO.

| [Example 1] | | | | | Repeat the commands up to an EDDO command while variable 1 contains “0”. |
|---|---|---|---|---|---|
| | | DWEQ | 1 | 0 | |
| | | : | | | |
| | 600 | LEAV | | | Forcibly end the loop if flag 600 is ON and proceed to the step next to an EDDO command. |
| | | : | | | |
| | | EDDO | | | |

## ● ITER (Repeat)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ITER | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Forcibly switch the control to EDDO while in a DO☐☐ loop.

[Example 1]

```
        DWEQ   1   0
          :
   600  ITER
          :
        EDDO
```

Repeat the commands up to an EDDO command while variable 1 contains “0”.
Forcibly switch the control to an EDDO command and perform end judgment, if flag 600 is ON.

## ● EDDO (End DO WHILE)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | EDDO | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Declare the end of a loop that began with DW☐☐.
If the DW☐☐ condition is not satisfied, the program will proceed to the step next to this command.

[Example 1] Refer to the section on DW☐☐.

## [15] Multi-Branching

### ● SLCT (Start selected group)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SLCT | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Branch to the step next to any WH□□ or WS□□ command that exists before an EDSL command and whose condition is satisfied, or to the step next to an OTHE command if none of the conditions are satisfied.
A SLCT command must be followed by a WH□□, WS□□ or EDSL command.
A maximum of 15 nests are supported.

(Note) Using a GOTO command to branch out of or into a SLCT-EDSL syntax is prohibited.

[Example 1]

| | | | | |
|---|---|---|---|---|
| | SCPY | 1 | 'Right' | Assign 'right' to columns 1 and 2. |
| | : | | | |
| 600 | SLCT | | | Jump to a W□□□ whose condition is satisfied. |
| | WSEQ | 1 | 'Right' | If 'right' is stored in columns 1 and 2, this command will be executed. |
| | : | | | |
| | WSEQ | 1 | 'Left' | If 'left' is stored, this command will be executed. |
| | : | | | |
| | OTHE | | | If the content of columns 1 and 2 is neither of the above, this command will be executed. |
| | : | | | |
| | EDSL | | | If flag 600 is OFF, the processing will move here upon execution of any of the conditions. |

## ● WH☐☐ (Select if true; variable)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | WH☐☐ | Variable number | Data | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] This command is used between SLCT and EDSL commands to execute the subsequent commands up to the next W☐☐☐ command or an OTHE or EDSL command when the comparison result of the content of the variable specified in operand 1 with the value specified in operand 2 satisfies the condition.

DW☐☐
- EQ ........... Operand 1 = Operand 2
- NE ........... Operand 1 ≠ Operand 2
- GT ........... Operand 1 > Operand 2
- GE ........... Operand 1 ≥ Operand 2
- LT ........... Operand 1 < Operand 2
- LE ........... Operand 1 ≤ Operand 2

[Example 1]

| | | | |
|---|---|---|---|
| LET | 1 | 20 | Assign 20 to variable 1. |
| LET | 2 | 10 | Assign 10 to variable 2. |
| : | | | |
| SLCT | | | Execute multi-branching. |
| WHEQ | 1 | 10 | (1) will be executed if the content of variable 1 is 10. Since variable 1 contains 20, however, the next condition will be referenced. |
| : | | | |
| (1) | | | |
| : | | | |
| WHGT | 1 | *2 | This command will be executed if the content of variable 1 is greater than the content of variable 2. Since variable 1 (= 20) > variable 2 (=10), (2) will be executed. |
| : | | | |
| (2) | | | |
| : | | | |
| OTHE | | | This command will be executed if none of the conditions are satisfied. In this example, since (2) was executed, (3) will not be executed. |
| : | | | |
| (3) | | | |
| : | | | |
| EDSL | | | The processing will move here if any of the conditions were satisfied and the applicable command executed. In this example, (2) and (4) will be executed. |
| : | | | |
| (4) | | | |
| : | | | |

* If multiple conditions are likely to be satisfied, remember that the first W☐☐☐ will become valid and any subsequent commands will not be executed. Therefore, state from the command with the most difficult condition or highest priority.

## ● WS□□ (Select if true; character)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | WS□□ | Column number | Column number, character literal | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] This command is used between SLCT and EDSL commands to execute the subsequent commands up to the next W□□□ command or an OTHE or EDSL command when the comparison result of the character strings in the columns specified in operands 1 and 2 satisfies the condition.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.

```
WS□□
  ├─ EQ  ...........  Operand 1 = Operand 2
  └─ NE  ...........  Operand 1 ≠ Operand 2
```

[Example 1]

| | | | |
|---|---|---|---|
| SLEN | 3 | | Set the number of comparing characters to 3. |
| SCPY | 1 | 'ABC' | Assign 'ABC' to column 1. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| : | | | |
| SLCT | | | Execute multi-branching. |
| WSEQ | 1 | 'XYZ' | (1) will be executed if columns 1 to 3 contain 'XYZ'. Since columns 1 to 3 contain 'ABC', however, this command will not be executed. |
| : | | | |
| (1) | | | |
| : | | | |
| WSEQ | 2 | *1 | (2) will be executed if the content of the number of characters specified by SLEN after column 2 is the same as the content of the column specified in variable 1. |
| : | | | |
| (2) | | | |
| : | | | |
| OTHE | | | This command will be executed if none of the conditions are satisfied. In this example, since (2) was executed, (3) will not be executed. |
| : | | | |
| (3) | | | |
| : | | | |
| EDSL | | | The processing will move here if any of the conditions were satisfied and the applicable command executed. In this example, (2) and (4) will be executed. |
| : | | | |
| (4) | | | |
| : | | | |

* If multiple conditions are likely to be satisfied, remember that the first W□□□ will become valid and any subsequent commands will not be executed. Therefore, state from the command with the most difficult condition or highest priority.

## ● OTHE (Select other)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | OTHE | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] This command is used between SLCT and EDSL commands to declare the command to be executed when none of the conditions are satisfied.

[Example 1] Refer to the sections on SLCT, WH□□ and WS□□.

## ● EDSL (End selected group)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | EDSL | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Declare the end of a SLCT command.

[Example 1] Refer to the sections on SLCT, WH□□ and WS□□.

## [16] System Information Acquisition

### ● AXST (Get axis status)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | AXST | Variable number | Axis number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Store in the variable specified in operand 1 the status (axis error number) of the axis specified in operand 2.

(Note 1) If the obtained result is "0", it means no axis error is present.
(Note 2) Since the error lists are written in hexadecimals, they must be converted to decimals.

[Example] AXST 1 2 Read the error number for axis 2 to variable 1.

If 3188 (decimal) is stored in variable 1 after the execution of this command:

3188 / 16 = 199 ...4
199 / 16 = 12 (= C) ...7

$3188 = 12 (= C) \times 16^2 + 7 \times 16^2 + 4$
= C74 (HEX) (Hexadecimal number)

Therefore, an "Error No. C74, Actual-position soft limit over error" is present.

## ●PGST (Get program status)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PGST | Variable number | Program number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Store in the variable specified in operand 1 the status (program error number) of the program specified in operand 2.

(Note 1) If the obtained result is "0", it means no program error is present.
(Note 2) Although the error lists are written in hexadecimals, the status to be stored (program error number) is a decimal.
Therefore, the decimal program error numbers must be converted to hexadecimals.

[Example] PGST 1 2 Read the error number for program No. 2 to variable 1.

## ● SYST (Get system status)

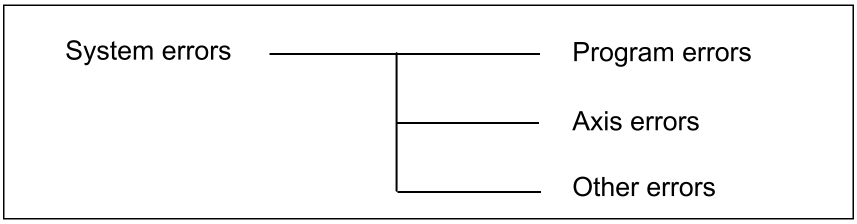
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SYST | Variable number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Store the system status (top-priority system error number) in the variable specified in operand 1.

(Note 1) If the obtained result is “0”, it means no system error is present.
(Note 2) Since the error lists are written in hexadecimals, they must be converted to decimals.
(Note 3) Relationship of error statuses




* An axis error that generates during operation with a program command will be registered both as a program error and an axis error.

[Example] SYST 1 Read the system error number to variable 1.

## ● GARM (Dedicated SCARA command/Get current arm system)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GARM | Variable number | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | ○ | ○ | ○ | ○ | × | × | ○ (PCX/PGX only) |

[Function] Acquire the current arm system and set one of the following values corresponding to this arm system in the variable specified in operand 1:

Arm system indeterminable = 0
Right arm system = 1
Left arm system = -1

(Note 1) The arm system effective immediately after the command execution is set. It is not that the arm system is always monitored.

(Note 2) In XSEL-RX/SX/RXD/SXD, XSEL-RAX/SAX/RAXD/SAXD 8-axes Series, GRP and BASE Command are available also in the actuator control declaration commands SLTL, SLWK, WGHT, PTPR, PTPL PTPE, PTPD, RIGH, LEFT and the system information acquirement command GARM. Establish the setting to have all the SCARA axes valid. Error No. C30 “Axis Pattern Error” will occur if even one axis is set invalid by GRP and BASE Commands.
When GRP and BASE Commands are undeclared, all the axes are effective (equivalent to GRP 11111111).

(Note 3) When GRP Command is undeclared, or GRP 11111111 (1st to 8th axes effective) is declared, the current arm system of the SCARA axes (1st to 4th axes) is set. When an acquirement of the current arm system for the SCARA axes (5th to 8th axes) is required, make only the 5th to 8th axes valid in GRP Command and execute GARM Command.

[Example 1]

| | | |
|---|---|---|
| GRP | 1111 | It makes the 1st to 4th axes effective. |
| GARM | 200 | Acquire the current arm system of the SCARA axes (1st to 4th axes) to Variable No. 200. |

[Example 2]

| | | |
|---|---|---|
| GRP | 11110000 | It makes the 5th to 8th axes effective. |
| GARM | 200 | Acquire the current arm system of the SCARA axes (5th to 8th axes) to Variable No. 200. |

[17] Zone

## ● WZNA (Dedicated linear axis command/Wait for zone ON based on AND gate)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | WZNA | Zone number | Axis pattern | CP |

| Applicable models |
|---|
| XSEL-JX/KX ×<br>Other than XSEL-JX/KX ○ |



[Function] Wait for the zone status of all axes (AND) specified by the axis pattern in operand 2 to become ON (inside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).

(Note 2) A maximum of four areas can be set as zones for each axis (“Axis-specific parameter No. 86 to 97”).

(Note 3) Zone output can be specified using “Axis-specific parameter No. 88, 91, 94 and 97” irrespective of this command.

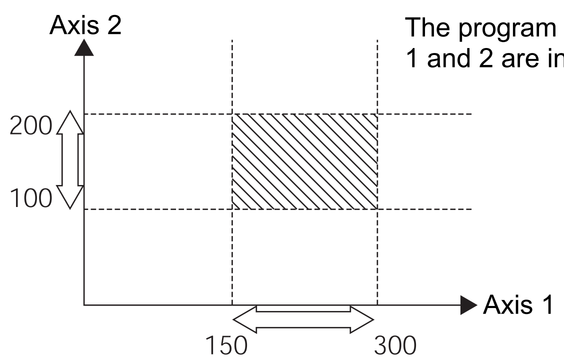
(Note 4) The zone signal is a dedicated command for linear axes. If a SCARA axis is specified for this command, “Error No. B80: Specification-prohibited axis error” occurs.

[Example 1] WZNA 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 and 2 becomes ON (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    5    3      Assign 3 to variable 5.
WZNA   1    *5
```

| | Axis 1 | Axis 2 |
|---|---|---|
| “Axis-specific parameter No. 86, Zone 1 max.” (Value is set in units of 0.001mm) | 300000 | 200000 |
| “Axis-specific parameter No. 87, Zone 1 min.” (Value is set in units of 0.001mm) | 150000 | 100000 |




The program will proceed to the next step if both axes 1 and 2 are inside the shaded area.

## ● WZNO (Dedicated linear axis command/Wait for zone ON based on OR gate)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | WZNO | Zone number | Axis pattern | CP |

| Applicable models |
|---|
| XSEL-JX/KX ×<br>Other than XSEL-JX/KX ○ |

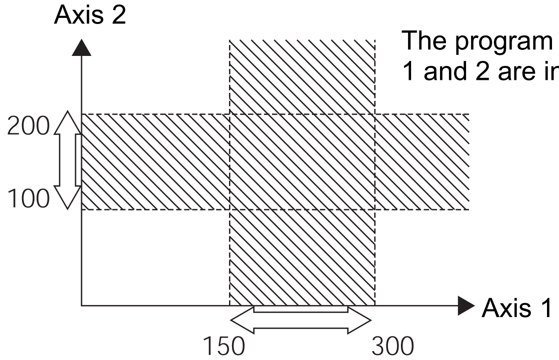
[Function] Wait for the zone status of any of the axes (OR) specified by the axis pattern in operand 2 to become ON (inside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).
(Note 2) A maximum of four areas can be set as zones for each axis ("Axis-specific parameter No. 86 to 97").
(Note 3) Zone output can be specified using "Axis-specific parameter No. 88, 91, 94 and 97" irrespective of this command.
(Note 4) The zone signal is a dedicated command for linear axes. If a SCARA axis is specified for this command, "Error No. B80: Specification-prohibited axis error" occurs.

[Example 1] WZNO 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 or 2 becomes ON (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)
LET 5 3 Assign 3 to variable 5.
WZNO 1 *5

| | Axis 1 | Axis 2 |
|---|---|---|
| "Axis-specific parameter No. 86, Zone 1 max." (Value is set in units of 0.001mm) | 300000 | 200000 |
| "Axis-specific parameter No. 87, Zone 1 min." (Value is set in units of 0.001mm) | 150000 | 100000 |




The program will proceed to the next step if both axes 1 and 2 are inside the shaded area.

## ● WZFA (Dedicated linear axis command/Wait for zone OFF based on AND gate)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | WZFA | Zone number | Axis pattern | CP |

| Applicable models |
|---|
| XSEL-JX/KX ×<br>Other than XSEL-JX/KX ○ |



[Function] Wait for the zone status of all axes (AND) specified by the axis pattern in operand 2 to become OFF (outside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).

(Note 2) A maximum of four areas can be set as zones for each axis (“Axis-specific parameter No. 86 to 97”).

(Note 3) Zone output can be specified using “Axis-specific parameter No. 88, 91, 94 and 97” irrespective of this command.

(Note 4) The zone signal is a dedicated command for linear axes. If a SCARA axis is specified for this command, “Error No. B80: Specification-prohibited axis error” occurs.

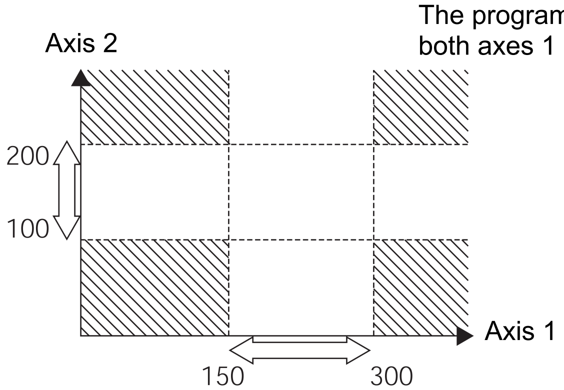
[Example 1] WZFA 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 and 2 becomes OFF (inside the shaded area shown in the diagram below)

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

| | | | |
|---|---|---|---|
| LET | 5 | 3 | Assign 3 to variable 5. |
| WZFA | 1 | *5 | |

| | Axis 1 | Axis 2 |
|---|---|---|
| “Axis-specific parameter No. 86, Zone 1 max.” (Value is set in units of 0.001mm) | 300000 | 200000 |
| “Axis-specific parameter No. 87, Zone 1 min.” (Value is set in units of 0.001mm) | 150000 | 100000 |

The program will proceed to the next step if both axes 1 and 2 are inside the shaded



## ● WZFO (Dedicated linear axis command/Wait for zone OFF based on OR gate)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | WZFO | Zone number | Axis pattern | CP |

| Applicable models |
|---|
| XSEL-JX/KX ×<br>Other than XSEL-JX/KX ○ |



[Function] Wait for the zone status of any of the axes (OR) specified by the axis pattern in operand 2 to become OFF (outside zone) with respect to the zone specified in operand 1.

(Note 1) The zone status of axes not yet completing home return will remain OFF (outside zone).

(Note 2) A maximum of four areas can be set as zones for each axis (“Axis-specific parameter No. 86 to 97”).

(Note 3) Zone output can be specified using “Axis-specific parameter No. 88, 91, 94 and 97” irrespective of this command.

(Note 4) The zone signal is a dedicated command for linear axes. If a SCARA axis is specified for this command, “Error No. B80: Specification-prohibited axis error” occurs.

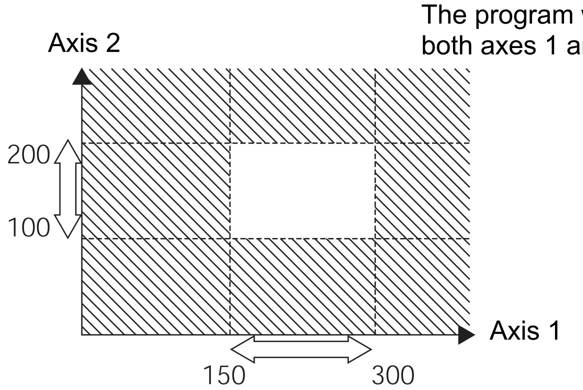
[Example 1] WZFO 1 11 If the parameters are set as follows, the program will wait until the zone status of axes 1 or 2 becomes OFF (inside the shaded area shown in the diagram below).

[Example 2] The axis pattern can be specified indirectly using a variable.
When the command in [Example 1] is rephrased based on indirect specification using a variable:
11 (binary) → 3 (decimal)

```
LET    5    3      Assign 3 to variable 5.
WZFO   1    *5
```

| | Axis 1 | Axis 2 |
|---|---|---|
| “Axis-specific parameter No. 86, Zone 1 max.” (Value is set in units of 0.001mm) | 300000 | 200000 |
| “Axis-specific parameter No. 87, Zone 1 min.” (Value is set in units of 0.001mm) | 150000 | 100000 |

The program will proceed to the next step if both axes 1 and 2 are inside the shaded



## [18] Communication

### ● OPEN (Open channel)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OPEN | Channel number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Open the channel specified in operand 1.
The specified channel will be enabled to send/receive hereafter.
Prior to executing this command, a SCHA command must be used to set an end character.

[Example 1]

```
SCHA   10
OPEN   1
```

Specify 10 (= LF) as the end character.
Open channel 1.

> Note: The following controller if "Open 1" is executed the teaching pendant connector (D-sub25pin) is cut off. (This is because channel 0 is used by both the teaching pendant and PC software.)
> XSEL-P/Q/PCT/QCT/PX/QX/R/S/RX/SX/RXD/SXD/RA/SA/RAX/SAX/RAXD/SAXD, ASEL/PSEL/SSEL
>
> The following controller if "Open 0" is executed the teaching pendant connector (D-sub25pin) is cut off. (This is because channel 1 is used by both the teaching pendant and PC software.)
> XSEL-J/JK, TT, TTA, MSEL,
> XSEL-K/KE/KT/KET/KX/KETX

## ● CLOS (Close channel)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CLOS | Channel number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Close the channel specified in operand 1.
The specified channel will be disabled to send/receive hereafter.

[Example 1]

```
CLOS    1
Close channel 1.

LET     1      2
CLOS    *1
        Assign 2 to variable 1.
        Close the content of variable 1 (channel 2).
```

## ● READ (Read)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | READ | Channel number | Column number | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Read a character string from the channel specified in operand 1 to the column specified in operand 2.
Read will end when the character specified by a SCHA command is received.
Either a local or global column may be specified.
A return code will be stored in a local variable (variable 99 under the factory setting) immediately after this command is executed.
Whether or not the command has been executed successfully can be checked based on this return code. Define appropriate processing to handle situations where the command execution failed due to an error.
Setting "0" in operand 2 will specify a dummy read (receive buffer cleared and receive disabled) (the return code will indicate that the command was successfully executed). The versions of tools in which 0 can be entered in operand 2 are specified below. With tools of these versions, 0 can be specified indirectly if it cannot be entered directly from the tool:

- PC software Ver.1.1.1.0 or later
- Teaching pendant TB-02 (D): First edition or later, TB-01 (D): First edition or later, SEL-T (D): First edition or later, IA-T-X (D): Ver.1.06 or later

[Example 1]

| | | | |
|---|---|---|---|
| SCHA | 10 | | Set LF (= 10) as the end character. |
| OPEN | 1 | | Open channel 1. |
| READ | 1 | 2 | Read a character string from channel 1 to column 2 until LF is received. |
| TRAN | 1 | 99 | Assign the return code (variable 99) to variable 1. |
| CLOS | 1 | | Close the channel 1. |
| SLCT | | | The processing flow branches out in accordance with each return code.<br>(Note) Using a GOTO command to branch out of an SLCT-EDSL syntax or to other branch processing within the syntax is prohibited. |
| WHEQ<br>:<br>(1)<br>: | 1 | 0 | If the content of variable 1 is "0" (Completed successfully), (1) will be executed. In (1), define the processing that should take place upon successful command execution. |
| WHEQ<br>:<br>(2)<br>: | 1 | 1 | If the content of variable 1 is "1" (Timeout), (2) will be executed. In (2), define appropriate processing to handle this situation, if necessary. |
| WHEQ<br>:<br>(3)<br>: | 1 | 2 | If the content of variable 1 is "2" (Timer cancelled), (3) will be executed. In (3), define appropriate processing to handle this situation, if necessary. |
| OTHE<br>:<br>(4)<br>: | | | If the content of variable 1 is not "0", "1" or "2", (4) will be executed. In (4), define appropriate error handling, if necessary. |
| EDSL | | | Once one of the specified conditions was met and the corresponding command has been executed, the processing will move here. |

(Note 1) A READ command must be executed before the other side sends the end character.

(Note 2) Dummy read (operand 2: 0) cannot be specified for channel No. 31 to 34 (Ethernet option).

```
SCHA    10
OPEN    1
READ    1        2
     <------------------ Other side
CLOS    1
```

- Return code of the READ command
  The return code is stored in a local variable. The variable number can be set by “Other parameter No. 24”. The default variable number is 99.

| | |
|---|---|
| 0 | : READ completed successfully (Receive complete) |
| 1 | : READ timeout (the timeout value is set by a TMRD command) (Continue to receive) |
| 2 | : READ timer cancelled (the wait status is cancelled by a TIMC command) (Continue to receive) |
| 3 | : READ SCIF overrun error (Receive disabled) |
| 4 | : READ SCIF receive error (framing error or parity error) (Receive disabled) |
| 5 | : READ factor error (program abort error) (Receive disabled) (Cannot be recognized by SEL commands) |
| 6 | : READ task ended (program end request, etc.) (Receive disabled) (Cannot be recognized by SEL commands) |
| 7 | : READ SCIF receive error due to other factor (Receive disabled) |
| 8 | : READ expansion SIO overrun error (Receive disabled) |
| 9 | : READ expansion SIO parity error (Receive disabled) |
| 10 | : READ expansion SIO framing error (Receive disabled) |
| 11 | : READ expansion SIO buffer overflow error (Receive disabled) |
| 12 | : READ expansion SIO receive error due to other factor (Receive disabled) |
| 13 to 20 | : Used only in Ethernet (optional) |
| 21 | : READ SIO receive temporary queue overflow error (Receive disabled) |
| 22 | : READ SIO slave receive queue overflow error (Receive disabled) |

## ● TMRD (Set READ timeout value)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TMRD | Timer period | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| ○ | × | × | × | ○ | × | × | × | × | TT:○, TTA:× | × |

[Function] Set the timeout to be applied to a READ command.
The timer setting specified in operand 1 will set the maximum time the program will wait for the character string read to end when a READ command is executed. If the end character could not be read before the timer is up during the execution of the READ command, a timeout will occur and the program will move to the next step.
(You can check if a timeout has occurred by checking the return code which is stored in a local variable (factory setting: variable 99) immediately after the READ command has been executed. If necessary, program an appropriate processing to be performed when a timeout occurs.)
Setting the timer to “0” will allow the READ command to wait infinitely, without timeout, until the end character is read.
The timer setting is input in seconds (setting range: 0 to 99.00sec) including up to two decimal places.

(Note) TMRD is set to “0” in the default condition before TMRD setting is performed.

[Example]

| | | | |
|---|---|---|---|
| SCHA | 10 | | Set LF (=10) as the end character. |
| TMRD | 30 | | Set the READ timeout value to 30sec. |
| OPEN | 1 | | Open channel 1. |
| READ | 1 | 2 | Read the character string from channel 1 to column 2 until LF is read. |
| TRAN | 1 | 99 | Assign the return code to variable 1. |
| CLOS | 1 | | Close the channel. |
| SLCT | | | The processing flow branches out in accordance with each return code. (Note) Using a GOTO command to branch out of an SLCT-EDSL syntax or to other branch processing within the syntax is prohibited. |
| WHEQ<br>:<br>(1)<br>: | 1 | 0 | If the content of variable 1 is “0” (Completed successfully), (1) will be executed. In (1), define the processing that should take place upon successful command execution. |
| WHEQ<br>:<br>(2)<br>: | 1 | 1 | If the content of variable 1 is “1” (Timeout), (2) will be executed. In (2), define appropriate processing to handle this situation, if necessary. |
| WHEQ<br>:<br>(3)<br>: | 1 | 2 | If the content of variable 1 is “2” (Timer cancelled), (3) will be executed. In (3), define appropriate processing to handle this situation, if necessary. |
| OTHE<br>:<br>(4)<br>: | | | If the content of variable 1 is not “0”, “1” or “2”, (4) will be executed. In (4), define appropriate error handling, if necessary. |
| EDSL | | | Once one of the specified conditions was met and the corresponding command has been executed, the processing will move here. |

Read completes successfully within 30sec → Variable No. 1 = 0
Timeout occurs → Variable No. 1 = 1

* The return code of READ command may not be limited to 0 or 1. The variable to store the return code can be set in “Other parameter No. 24”. (Main application Ver.0.21 or later) For details, refer to the explanation of the READ command.

## ● TMRW (Set READ/WRIT timeout value)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TMRW | Read timer setting | (Write timer setting) | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | ○ | ○ | ○ | ○ | TT:○, TTA:× | ○ |

[Function] Set the timeout to be applied to a READ/WRIT command.
The timer setting specified in operand 1 will set the maximum time the program will wait for the character string read to end when a READ command is executed.
If the end character could not be read before the timer is up during the execution of the READ command, a timeout will occur and the program will move to the next step. (You can check if a timeout has occurred by checking the return code which is stored in a local variable (factory setting: variable 99) immediately after the READ command has been executed.)
If the timer period is set to 0, the READ command causes the program to wait infinitely until the end characters are read, by assuming that there is no timeout.
The timer setting is input in seconds (setting range: 0 to 99.00sec) including up to two decimal places.
A variable can be specified indirectly in operand 1.

(Note) TMRW is set to “0” in the default condition before TMRW setting is performed.

| [Example] | SCHA | 10 | | Set LF (=10) as the end character. |
|---|---|---|---|---|
| | TMRW | 30 | | Set the READ timeout value to 30sec. |
| | OPEN | 1 | | Open channel 1. |
| | READ | 1 | 2 | Read the character string from channel 1 to column 2 until LF is read. |
| | TRAN | 1 | 99 | Assign the return code to variable 1. |
| | CLOS | 1 | | Close the channel. |

Read completes successfully within 30sec → Variable No. 1 = 0
Timeout occurs → Variable No. 1 = 1

* The return code of READ command may not be limited to 0 or 1. The variable to store the return code can be set in "Other parameter No. 24". Refer to the explanation of READ command for details.

For the time period specified in operand 2, set the timeout value to be applied when a WRIT command is executed (maximum wait time for completion of send). (Maximum wait time for end based on flow control)
The write timer period is valid only for standard SIO (channels 1 and 2 supporting flow control).

For the time period specified in operand 2, set the timeout value to be applied when a WRIT command is executed (maximum wait time for completion of send). (Maximum wait time for end based on flow control) (Arbitrary)
The write timer setting is available only on standard SIO (flow control support channels 1 and 2).
This command is recognized as a TMRD on XSEL-JX/KX controllers, and as TMRW on XSEL-PX/QX controllers. If a program created for an XSEL-JX/KX controller is transferred to an XSEL-PX/QX controller, the PC software automatically converts "TMRD" to "TMRW" before the file is transferred. This command is recognized as a TMRD on XSEL-JX/KX controllers, and as TMRW on XSEL-PX/QX controllers. If a program created for an XSEL-JX/KX controller is transferred to an XSEL-PX/QX controller, the PC software automatically converts "TMRD" to "TMRW" before the file is transferred.

## ●WRIT (Write)

| Extension condition (LD,A,O,AB,OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | WRIT | Channel number | Column number | CC(Note 1) |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Write the character string in the column specified in operand 2 to the channel specified in operand 1.
The operation will end when the character specified by a SCHA command is written.
Either a local or global column can be specified.

[Example]

| | | | |
|---|---|---|---|
| SCHA | 10 | | Set LF (= 10) as the end character. |
| OPEN | 1 | | Open channel 1. |
| WRIT | 1 | 2 | Write the character string in column 2 to channel 1 until LF is written. |
| CLOS | 1 | | Close the channel. |

With a standard SIO (channel 1 or 2), WRIT is supported by (can be sent in) a task other than the one that opened the channel, as long as the channel is currently open. Accordingly, by sending WRIT in other task after executing READ in a task that opened the channel, a response can be received from the other side without delay after sending from XSEL.

(Note 1) CP is performed if the channel is other than 1 and 2.

Return code of WRIT command (channels 1 and 2 only)
The return code is stored in a local variable. The variable number can be set by “Other parameter No. 24”. The default variable number is 99.
0 : WRIT completed successfully
1 : WRIT timeout (the timeout value is set by a TMRW command)
2 : WRIT timer cancelled (the wait status is cancelled by a TIMC command)
3 to 4 : Reserved by the system
5 : WRIT factor error (program abort error) (Cannot be recognized by SEL commands)
6 : WRIT task ended (program end request, etc.) (Cannot be recognized by SEL commands)

## ● SCHA (Set end character)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SCHA | Character code | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the end character to be used by a READ or WRIT command.
Any character from 0 to 255 (character code used in BASIC, etc.) can be specified.

[Example] Refer to the sections on READ and WRIT commands.

## ● IPCN (Connected Destination IP address / Port Number Setting) [Ethrnet]

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | IPCN | Channel number | Integer variable number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] It is a command used in the non-procedure communication in Ethernet.
The storage area for the connected destination IP address and port number in the free-for-user TCP/IP channel should be set established. The connected destination information stored in the five integer variables in a row starting from an integer variable number indicated in Operation 2 should be set as the connected destination of the free-for-user TCP/IP channel number indicated in Operation 1.
Make sure that this command is executed before OPEN Command.

Variable Number Indicated in Operation 2

| variable number | |
|---|---|
| n | ····Connected Destination IP Address (H) Stored Variable Number |
| n+1 | ····Connected Destination IP Address (MH) Stored Variable Number |
| n+2 | ····Connected Destination IP Address (ML) Stored Variable Number |
| n+3 | ····Connected Destination IP Address (L) Stored Variable Number |
| n+4 | ····Connected Destination Port Number Stored Variable Number |

(Note 1) The channel numbers available to indicate in Operation 1 in the Ethernet option should be from 31 to 34.

(Note 2) If this command gets executed during open, it should be set as the command for the next open.

[Example]

| | | | |
|---|---|---|---|
| LET | 90 | 192 | Connected Destination IP Address (H) = 192 |
| LET | 91 | 168 | Connected Destination IP Address (MH) = 168 |
| LET | 92 | 72 | Connected Destination IP Address (ML) = 72 |
| LET | 93 | 101 | Connected Destination IP Address (L) = 101 |
| LET | 94 | 64514 | Connected Destination Port Number = 64514 |
| IPCN | 31 | 90 | Declares Channel 21 connected destination IP address / port number stored area = Local integer variables from 90 to 94. |

In this example, IP address 192. 168. 72. 101 and port number 64514 are set as the connected destination for Free-for-User TCP/IP Channel No. 31.

## [19] String Operation

### ● SCPY (Copy character string)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SCPY | Column number | Column number, character literal | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Copy the character string in the column specified in operand 2 to the column specified in operand 1.
Copy will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, copy will be performed for the entire length of the literal.

[Example 1]

| | | | |
|---|---|---|---|
| SCPY | 1 | ‘ABC’ | Copy ‘ABC’ to column 1. |
| SLEN | 10 | | Set the copying length to 10 bytes. |
| SCPY | 100 | 200 | Copy 10 bytes from column 200 to column 100. |

## ● SCMP (Compare character strings)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SCMP | Column number | Column number, character literal | EQ |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Compare the column specified in operand 1 with the column specified in operand 2.
Comparison will be performed for the length set by a SLEN command.
If a character literal is specified in operand 2, comparison will be performed for the entire length of the literal.

[Example 1]

| | | | | |
|---|---|---|---|---|
| SCMP | 1 | 'ABC' | 600 | Flag 600 will turn ON if columns 1 to 3 contain 'ABC'. |
| SLEN | 5 | | | Set the comparing length to 5 bytes. |
| SCMP | 10 | 30 | 999 | Turn ON flag 999 if five bytes from columns 30 and 10 match. |

## ● SGET (Get character)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SGET | Variable number | Column number, character literal | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Assign one character from the column specified in operand 2 to the variable specified in operand 1.
If a character-string literal is specified in operand 2, the first character will be assigned.

[Example 1]

```
SGET    1     100
Assign one byte from column 100 to variable 1.

LET     1     3       Assign 3 to variable 1.
LET     2     1       Assign 1 to variable 2.
SCPY    1     'A'     Copy 'A' to column 1.
SGET    *1    *2      Assign 'A' from the content of variable 2 (column 1)
                      to the content of variable 1 (variable 3).
```

## ● SPUT (Set character)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SPUT | Column number | Data | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the data specified in operand 2 in the column specified in operand 1.

[Example 1]

| | | | |
|---|---|---|---|
| SPUT | 5 | 10 | Set 10 (LF) in column 5. |
| | | | |
| LET | 1 | 100 | Assign 100 to variable 1. |
| LET | 2 | 50 | Assign 50 to variable 2. |
| SPUT | *1 | *2 | Set the content of variable 2 (50 (‘2’)) in the content of variable 1 (column 100). |

## ● STR (Convert character string; decimal)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | STR | Column number | Data | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Copy to the column specified in operand 1 a decimal character string converted from the data specified in operand 2.
The data will be adjusted to the length set by a SLEN command.
If the data exceeds the specified length, it will be cut off at the length set by a SLEN command.
If the entire data has been converted within the length set by a SLEN command, the output will turn ON.

(Note) If the data specified in operand 2 is a 10 digit integer including eight or more valid digits, conversion of the values in the eighth and subsequent digits will not be guaranteed (the values through the seventh digits will be converted properly.)

[Example]

| | | | |
|---|---|---|---|
| SLEN | 5.3 | | Set a length consisting of five integer digits and three decimal digits. |
| STR | 1 | 123 | The following values will be set in columns 1 to 9: |

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | . | 0 | 0 | 0 |

| | | | |
|---|---|---|---|
| LET | 1 | 10 | Assign 10 to variable 1. |
| LET | 102 | 987.6543 | Assign 987.6543 to variable 102. |
| SLEN | 2.3 | | Set a length consisting of two integer digits and three decimal digits. |
| STR | *1 | *102 | The following values will be set in columns 10 to 15: |

| 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|
| 8 | 7 | . | 6 | 5 | 4 |

Since the data exceeds the specified length, 87 without 9 in the 100s place is set in the integer part, while 654 with 3 in the fourth decimal place rounded is set in the fraction part.

## ● STRH (Convert character string; hexadecimal)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | STRH | Column number | Data | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



[Function] Copy to the column specified in operand 1 a hexadecimal character string converted from the data specified in operand 2.
Only the integer part will be adjusted to the length set by a SLEN command.
If the data exceeds the specified length, it will be cut off at the length set by a SLEN command.
If the entire data has been converted within the length set by a SLEN command, the output will turn ON.

(Note) If the data specified in operand 2 is a negative value, 8 columns will be required to covert the entire data.

[Example]

| | | | |
|---|---|---|---|
| SLEN | 5 | | Set a format consisting of 5 integer digits. |
| STRH | 1 | 255 | The following values will be set in columns 1 to 5: |

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| | | | F | F |

| | | | |
|---|---|---|---|
| LET | 1 | 10 | Assign 10 to variable 1. |
| LET | 102 | 987.6543 | Assign 987.6543 to variable 102. |
| SLEN | 2.3 | | Set a length consisting of 2 integer digits and 3 decimal digits. |
| STRH | *1 | *102 | The following values will be set in columns 10 and 11: |

| 10 | 11 |
|---|---|
| D | B |

“.3” in the SLEN command and “.6543” in variable 102, which are the decimal part, will be ignored.
The integer part is expressed as ‘3DB’ in hexadecimal. Since the length is two digits, however, “3” in the third digit will be cut off.

## ● VAL (Convert character string data; decimal)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | VAL | Variable number | Column number, character literal | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Convert the decimal data in the column specified in operand 2 to a binary and assign the result to the variable specified in operand 1.
Conversion will be performed for the length set by a SLEN command.
If a character-string literal is specified in operand 2, conversion will be performed for the entire length of the literal.

(Note) Keep the converting length to 18 characters or less.

[Example]

| | | | |
|---|---|---|---|
| SCPY | 10 | ‘1234’ | Set ‘1234’ in column 10. |
| SLEN | 4 | | Set the converting length to 4 bytes. |
| VAL | 1 | 10 | Assign 1234, which is a binary converted from ‘1234’ in column 10, to variable 1. |

| | | | |
|---|---|---|---|
| LET | 1 | 100 | Assign 100 to variable 1. |
| LET | 2 | 20 | Assign 20 to variable 2. |
| SCPY | 20 | ‘1234’ | Copy ‘1234’ to column 20. |
| SCPY | 24 | ‘.567’ | Copy ‘.567’ to column 24. |
| SLEN | 8 | | Set the converting length to 8 bytes. |
| VAL | *1 | *2 | Assign 1234.567, which is a binary converted from ‘1234.567’ in the content of variable 2 (column 20) to the content of variable 1 (variable 100). |

## ● VALH (Convert character string data; hexadecimal)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | VALH | Variable number | Column number, character literal | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Convert the hexadecimal data in the column specified in operand 2 to a binary and assign the result to the variable specified in operand 1.
Conversion will be performed for the length set by a SLEN command.
Only the integer part will be converted, with the decimal part being ignored.
If a character-string literal is specified in operand 2, conversion will be performed for the entire length of the literal.

(Note) Keep the converting length to 8 characters or less.

[Example]

| | | | |
|---|---|---|---|
| SCPY | 10 | '1234' | Set '1234' in column 10. |
| SLEN | 4 | | Set the converting length to 4 bytes. |
| VALH | 1 | 10 | Assign 4660, which is a binary converted from hexadecimal '1234' in column 10, to variable 1. |
| | | | |
| LET | 1 | 100 | Assign 100 to variable 1. |
| LET | 2 | 20 | Assign 20 to variable 2. |
| SCPY | 20 | 'ABCD' | Copy 'ABCD' to column 20. |
| SLEN | 4 | | Set the converting length to 4 bytes. |
| VALH | *1 | *2 | Assign 43981, which is a binary converted from hexadecimal 'ABCD' in the content of variable 2 (column 20) to the content of variable 1 (variable 100). |

## ● SLEN (Set length)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SLEN | Character string length | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the length to be processed by a string command.
This must always be set before using the following commands:

SCMP ............... Decimal part is invalid.
SCPY ............... Decimal part is invalid.
IS□□ ............... Decimal part is invalid.
WS□□ ............... Decimal part is invalid.
STRH ............... Decimal part is invalid.
VAL, VALH ............... Decimal part is invalid.
STR ............... Decimal part is valid.

[Example] Refer to the examples of the above commands:

[20] Arch-Motion

## ● ARCH (Arch motion)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ARCH | Position number | Position number | PE |

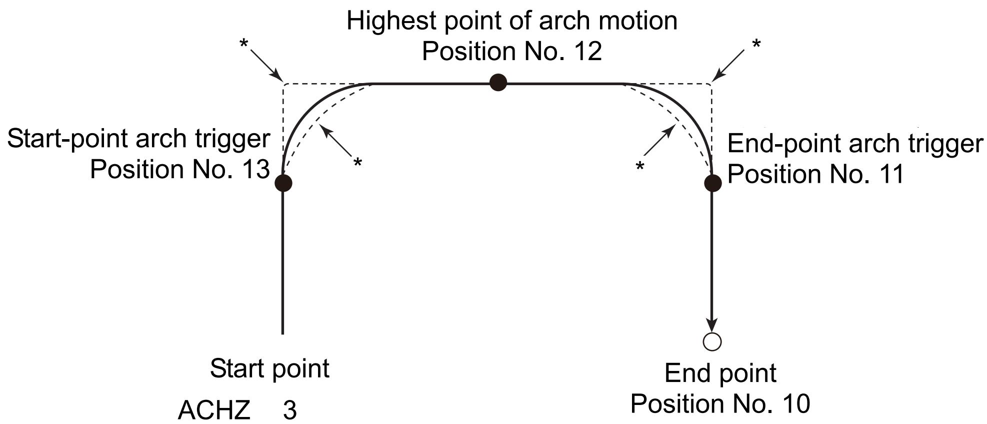
| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Perform arch motion from the current point and move to the specified points.

- Move to the points specified in operand 1, via arch motion.
- Movements in directions other than the arch-motion Z-axis direction will begin after rising from the current point to the start-point arch trigger. After the Z point specified in operand 2 (as the highest point) is passed and movements in directions other than the arch-motion Z-axis direction are complete, the axes will come down to the end-point arch trigger and reach the specified point.
- Palletizing arch triggers must be set using an ATRG command.

(Note 1) If the arch motion setting that SCARA axis and linear drive axis exist together is established, 421 “SCARA/Linear Drive Axes Double Indication Error” will occur. Also, if the arch motion setting that SCARA axes for two units exist together is established, B80 “Indication Prohibited Axis Error” will occur. Establish the arch motion setting with a consideration to have the operation axes all the same SCARA axes or all linear drive axes.

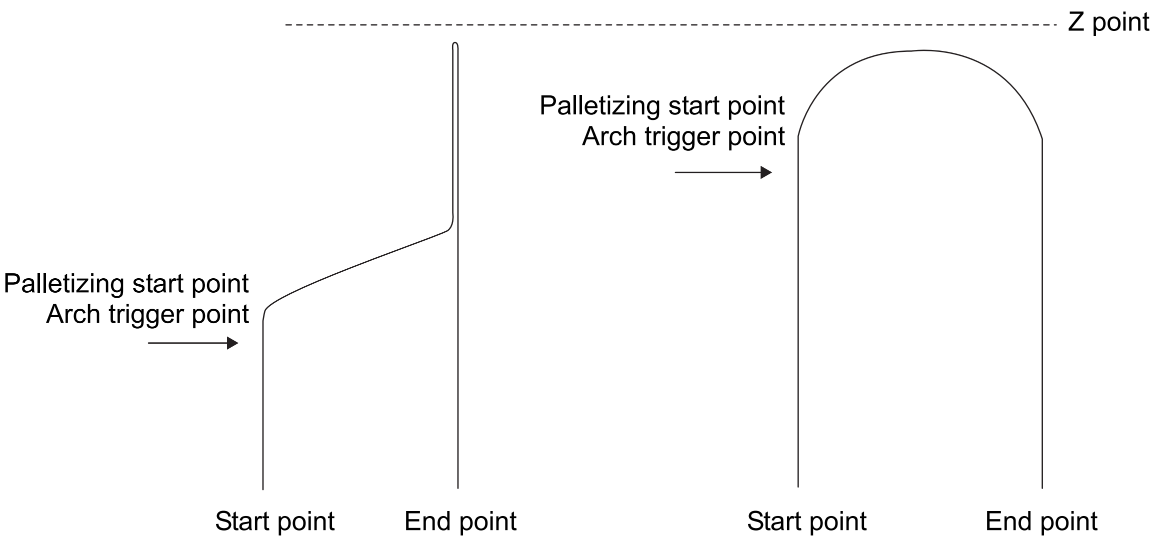
(Note 2) The arch motion operation of SCARA axis is PTP operation and the linear drive axis is CP operation.




```
ACHZ   3
ATRG   13    11
 •
 •
 •
ARCH   10    12
```

* When the operation is resumed after a pause, depending on the position where the operation is resumed the locus may follow the lines (dotted lines) indicated by asterisks in the diagram for the composite section from ascent to horizontal movement or from horizontal movement to descent. Be careful not to cause interference.

- The arch-motion Z-axis coordinate of the end point will become the arch-motion Z-axis component of the point data specified in operand 1, if any, plus the arch-motion Z-axis offset. If there is no arch-motion Z component, the arch-motion Z-axis coordinate of the end point will become the arch-motion Z-axis coordinate of the start point plus the arch-motion Z-axis offset. (Normally the offset is added to all arch-motion positions, such as the arch triggers and Z point.)
- An error will generate if the start-point arch trigger is set below the start point or the end-point arch trigger is set below the end point. (Note: Up/down has nothing to do with +/- on the coordinate system.)
- The arch-motion Z-axis up direction refers to the direction toward the Z point from the start point (the down direction refers to the opposite direction), and has nothing to do with the size of coordinate value. Therefore, be sure to confirm the actual operating direction when using this command.
- The arch-motion Z-axis will come down after a rise-process command value is output. Therefore, one of the following operations will be performed depending on how the arch-trigger point and Z point are set.
  If the resulting operation is undesirable, change the arch trigger and/or Z point to improve the efficiency of movement.




- As for the arch-trigger end position data, movement also starts/ends above the applicable arch trigger for any effective axis, other than the arch motion Z-axis, if data of such axis is included in the position data.
- If the end position data includes R-axis data, movement of the R-axis starts/ends above the applicable arch trigger.
- If a composite arch trigger motion is set, a given effective axis, other than the arch motion Z-axis, also moves if data of such axis is included in the end point data. In this case, movement of the axis also starts/ends above the applicable arch trigger.

## ● ACHZ (Declare arch-motion Z-axis)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ACHZ | Axis number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Specify the axis number representing the arch-motion Z direction.
The axis number specified in operand 1 will be set as the axis number representing the arch-motion Z direction.
If the output field is specified, the output will turn ON after this command is executed.

[Example] ACHZ 3

(Note 1) The arch motion Z-axis is available for indication only on the work coordinate system Z-axis (Axis No. 3 or Axis No. 7).

(Note 2) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, R-axis of the coordinate system definition unit cannot be indicated for the arch motion Z-axis. Also, X and Y-axes of the coordinate system definition unit can be indicated only when the offset of R-axis of the work coordinate system is set to 0.

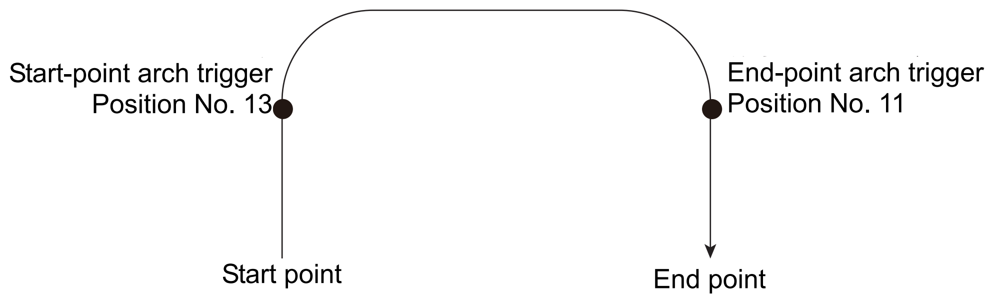
## ● ATRG (Set arch triggers)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ATRG | Position number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Set the arch triggers used for arch motion.

(This setting becomes valid when an ARCH command is executed.)
Set the arch-motion Z-axis position data in the point data specified in operand 1 as the start-point arch trigger, and set the arch-motion Z-axis position data in the point data specified in operand 2 as the end-point arch trigger.




ATRG 13 11

(Refer to “Palletizing Setting” – “Arch triggers” under “How to Use”.)
For an arch-motion operation, set it so that a horizontal movement will begin when the start-point arch trigger is reached during ascent from the start point, and that the end-point arch trigger will be reached after a horizontal movement is completed during descent.
If the output field is specified, the output will turn ON after this command is executed.

## ● AEXT (Set composite arch motion)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | AEXT | (Position number) | Prohibited | CP |

| Applicable models |
|---|
| ASEL/PSEL/SSEL ×<br>Other than ASEL/PSEL/SSEL ○ |

Set a composite arch motion. Set coordinate values other than the arch motion Z-axis at the end position of arch motion.
Use the position number specified in operand 1 for setting composite motion.
With SCARA robots, the R-axis becomes a composite arch motion axis.
When the arch motion is executed, the end coordinate of the composite axis corresponds to effective axis data, other than that of the arch motion Z-axis, included in the arch-motion end point data.
If nothing is specified in operand 1, the position number already declared for setting composite motion becomes invalid. If the output is specified, it turns ON after this command has been executed.

(Note 1) Setting of the arch motion composition axes cannot be established for linear drive axes in PX/QX.

## ● OFAZ (Set arch-motion Z-axis offset)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OFAZ | Offset value | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Set the offset in the arch-motion Z-axis direction.
The value specified in operand 1 will be set as the offset in the arch-motion Z-axis direction.
The offset amount is set in mm and the effective resolution is 0.001mm.
A negative value can also be specified as the offset, as long as the operation range will not be exceeded.
This offset is valid only at the end point of ARCH (arch motion) operation.
If the output field is specified, the output will turn ON after this command is executed.

## [21] Palletizing Definition

### ● BGPA (Declare start of palletizing setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | BGPA | Palletizing number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Declare the start of a palletizing setting.
Once this command is executed, palletizing setting for the palletizing number specified in operand 1 will be enabled.
(In the case of an ACHZ, AEXT, OFAZ or ATRG command, setting is enabled without declaring BGPA.)
The input range of palletizing number is from 1 to 10. XSEL-RA/SA/RAX/SAX/RAXD/SAXD is 1 or more 32 or less.
When the palletizing setting is complete, execute EDPA.
Nested BGPAs are not supported. To declare start of another palletizing setting, execute an EDPA command and then execute a BGPA command again.
If the output field is specified, the output will turn ON after this command is executed.

(Note) Using a GOTO command to branch out of or into a BGPA-EDPA syntax is prohibited.

## ● EDPA (Declare end of palletizing setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | EDPA | Prohibited | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Declare the end of a palletizing setting.
If a palletizing-setting command (excluding BGPA, ACHZ, ATRG, AEXT and OFAZ) is executed before another BGPA is declared following an execution of this command (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

## ● PAPI (Set palletizing counts)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PAPI | Count | Count | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

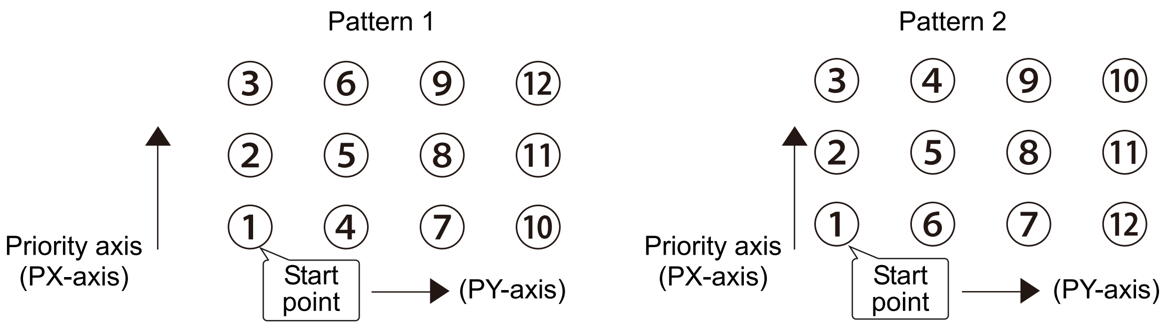
Set counts in the palletizing-axis directions.
The count specified in operand 1 will apply to the preferential-axis (PX-axis) direction, while the count specified in operand 2 will apply to the PY-axis direction.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

## ● PAPN (Set palletizing pattern)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PAPN | Pattern number | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Set a palletizing pattern.
The palletizing pattern specified in operand 1 will be set (1 = Pattern 1, 2 = Pattern 2).
If this command is not declared, pattern 1 will be used.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.



## ● PASE (Declare palletizing axes)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PASE | Axis number | Axis number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



Set the two axes to be used in palletizing (PX and PY-axes).
The axis specified in operand 1 will be set as the preferential axis (PX-axis).
The axis specified in operand 2 will be set as the PY-axis.
This command is used in conjunction with PAPT and PAST.
It cannot be used together with a 3-point teaching (PAPS) command. Whichever is set later will be given priority.

3-point teaching (PAPS) is recommended for palletizing that requires precision.

If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

If the arch motion setting that SCARA axis and linear drive axis exist together is established, 421 “SCARA/Linear Drive Axes Double Indication Error” will occur.
Also, if the arch motion setting that SCARA axes for two units exist together is established, B80 “Indication Prohibited Axis Error” will occur.
Establish the arch motion setting with a consideration to have the operation axes all the same SCARA axes or all linear drive axes.

## ● PAPT (Set palletizing pitches)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PAPT | Pitch | Pitch | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Set palletizing pitches.
The value specified in operand 1 will be set as the pitch for the preferential axis (PX-axis), while the value specified in operand 2 will be set as the pitch for the PY-axis.
This command is used in conjunction with PASE and PAST.
If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error will generate.
If the output field is specified, the output will turn ON after this command is executed.

## ● PAST (Set palletizing reference point)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PAST | (Position number) | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



Set the reference point for PX-axis (priority axis), PY-axis and PZ-axis (when palletizing Z-axis declaration is effective) to be used in palletizing calculation.
If a value is set in operand 1, that position number specified in operand 1 will be used to store the reference point data.
If no value is set in operand 1, the position-number setting for storing reference point data will become invalid.
This command is used in conjunction with PASE and PAPT.

If this command is not set, the reference point is defined as X = 0, Y = 0.
Palletizing positions are calculated as points on the palletizing plane constituted by the reference point, PX-axis and PY-axis.
Accordingly, position data of the reference point must include valid coordinate components for PX-axis, PY-axis and PZ-axis (when palletizing Z-axis declaration is effective). If these coordinate components are invalid, an error occurs during palletizing position coordinate calculation for PAPG (Get palletizing calculation data) or other palletizing movement command. Coordinate components of other axes are ignored during palletizing position coordinate calculation.

An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed.

If the arch motion setting that SCARA axis and linear drive axis exist together is established, 421 “SCARA/Linear Drive Axes Double Indication Error” will occur.
Also, if the arch motion setting that SCARA axes for two units exist together is established, B80 “Indication Prohibited Axis Error” will occur.
Establish the arch motion setting with a consideration to have the operation axes all the same SCARA axes or all linear drive axes.

(Note 1) In the case of SCARA robots, executing a palletizing movement command while the work coordinate system selection number is set to 0 (base coordinate system) and this command is not yet set generates an error because the palletizing start position is (0, 0) and thus movement is disabled.

(Note 2) In the case of SCARA robots, the R-axis should be excluded from the effective axes, if already set in the position data, with a GRP command.
(This is not required if the R-axis field is blank.)
Set the R-axis data at the palletizing position using a PEXT command.

## ● PAPS (Set palletizing points) For 3-point or 4-point teaching

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PAPS | Position number | (Palletizing position setting type) | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Set palletizing positions in 3-point teaching.
It can also be used to set palletizing positions in 4-point teaching, in which case the pallet plane can be set to any quadrilateral other than a square, rectangle or parallelogram.
In operand 1, set the position number of the start point needed to set palletizing positions in 3-point teaching. If "n" is set as the position number for the start point, position data for the end point in the PX-axis direction will be stored in position No. n+1, while position data for the end point in the PY-axis direction will be stored in position No. n+2.
In the case of 4-point teaching, position data for the end point should be stored in position No. n+3.

(Note 1) If the arch motion setting that SCARA axis and linear drive axis exist together is established, 421 "SCARA/Linear Drive Axes Double Indication Error" will occur. Also, if the arch motion setting that SCARA axes for two units exist together is established, B80 "Indication Prohibited Axis Error" will occur. Establish the arch motion setting with a consideration to have the operation axes all the same SCARA axes or all linear drive axes.

In operand 2, specify the applicable palletizing position setting type.

[Palletizing position setting type]
If operand 2 is "0" or blank, 3-point teaching will be specified.
As shown in Fig. 1 (a), palletizing positions will be set on the quadrilateral pallet plane determined by the three points including the start point, end point in the PX-axis direction and end point in the PY-axis direction.

If operand 2 is "2," 4-point teaching will be specified.
As shown in Fig. 1 (b), palletizing positions will be set on the quadrilateral pallet plane determined by the four points including the start point, end point in the PX-axis direction, end point in the PY-axis direction, and end point. Note, however, that whether the shape is planar or not varies depending on the end point data.

Fig. 1 shows two different arrangements of palletizing positions.

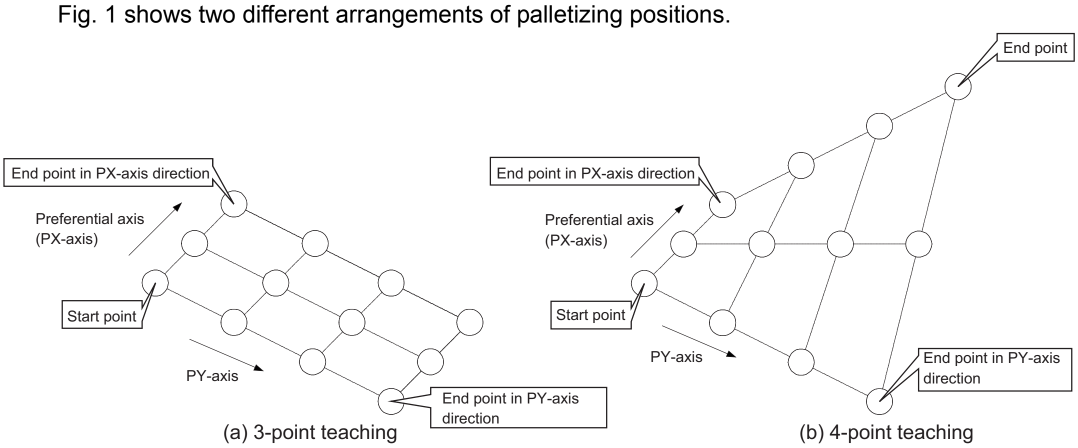



Fig. 1 Layout of Palletizing Positions

(Note) Since ASEL, PSEL and SSEL controllers are 2-axis controllers, setting 2 in operand 2 results in the planar type, just like 1 is set.

If palletizing positions are set by 4-point teaching, it is recommended that the non-planar type be specified as long as all four points are known to be on the plane and the palletizing requires precision.

If operand 2 is set to 1, 4-point teaching (planar type) is set.
Fig. 2-(a)
The plane is determined by three points including the start point, end point in PX-axis direction and end point in PY-axis direction. The end point is moved in parallel in PZ direction (vertical direction) and the point of intersection with the aforementioned plane is defined as the end point for this type of palletizing.
Palletizing positions are placed on the quadrilateral pallet surface determined by these four points.

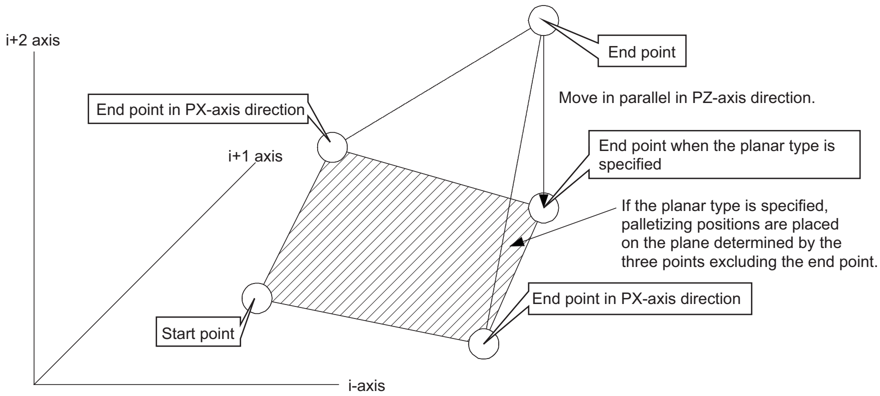



Fig. 2-(a)

Take note, however, that the moving direction of the end point varies if the three points other than the end point meet the conditions specified in Table 1. This is when the plane determined by the three points other than the end point is vertical to the ground. In this case, moving the end points in parallel with PZ direction (vertical direction) does not find a point of intersection with this plane.

Table 1 Moving Direction of End Point Based on Planar Type Specification

| Condition | Moving direction of end point |
|---|---|
| Point data other than i-axis component matches among the three points other than the end point. (Refer to Fig. 2-(b)) | Move in parallel in i-axis direction. |
| Point data other than PZ-axis component matches between the start point and end point in PX-axis direction. (Refer to Fig. 2-(c)) | Move in parallel in the direction of one of the two axes other than the PZ-axis, whichever has the smaller axis number. |
| Point data other than PY-axis component matches between the start point and end point in PX-axis direction. (Refer to Fig. 2-(c)) | |
| Point data other than PZ-axis component matches between the end point in PX-direction and end point in PY-axis direction. (Refer to Fig. 2-(c)) | |

* i indicates the axis number of one of the two axes other than the PZ-axis.

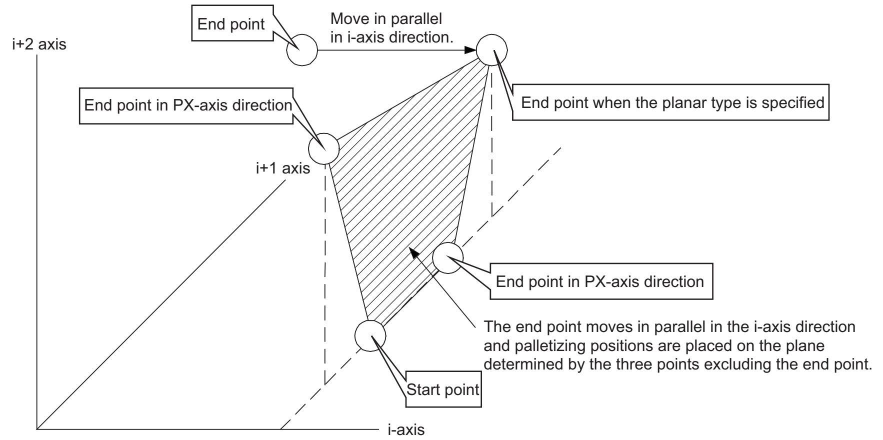



Fig. 2-(b)

The point data for i-axis component matches among the three points other than the end point:

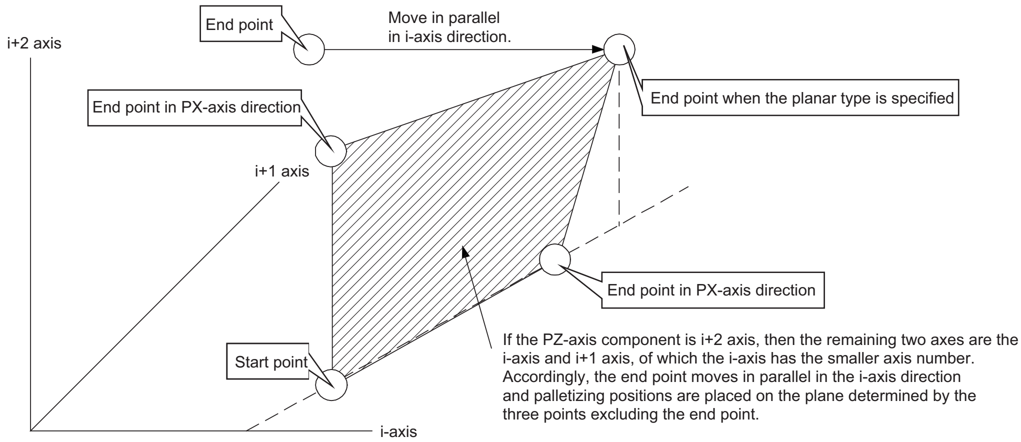



Fig. 2-(c)

The point data other than PZ-axis component matches between two of the three points other than the end point:
(In the figure above, the point data other than PZ-axis component matches between the start point and end point in PY-axis direction.)

- If the valid axis pattern does not match the point data for 3-point teaching or 4-point teaching, an error “CB0, Mismatched valid axes for palletizing 3-point teaching data” will generate. If a PAPS command is executed after specifying the applicable axes using a GRP command, only the point data corresponding to the specified axes, among all axes whose point data is valid, will be used as palletizing point data. Executing a GRP command thereafter with a different setting will have no effect.
- If the PZ-axis has been declared, there must be two effective axes other than the PZ-axis. If the PZ-axis is not yet declared, there must be two or three effective axes. If there are not enough effective axes, a “CAE: Insufficient effective axes for palletizing point data by 3-point teaching” occurs. If there are too many effective axes, on the other hand, a “CAF: Excessive effective axes for palletizing point data by 3-point teaching” occurs. If the planar type is specified and PZ-axis is not yet declared, set two effective axes. If the number of effective axes is other than 2, a “CB4: Arch motion Z-axis non-declaration error” occurs.
- This command cannot be used with a PASE (set palletizing axes) command. Whichever was set later will be given priority. (A single PAPS command can substitute a set of PASE, PAPT and PAST commands.)
- If this command is executed before BGPA is declared (= while palletizing setting is not enabled), an error, “CB5, BGPA not declared at palletizing setting” will generate.
- If the output field is specified, the output will turn ON after this command is executed.

## ● PSLI (Set zigzag)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PSLI | Offset amount | (Count) | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Set a zigzag palletizing.
The value specified in operand 1 will be set as the offset amount for even-numbered rows.
The count specified in operand 2 will be set as the count for even-numbered rows.

[Refer to 3.6.5 Palletizing Function]

If operand 2 is not specified, the count for even-numbered rows will become the same as the count for odd-numbered rows.
If palletizing is set with PAPS (Set palletizing points) based on 3-point teaching, the PX and PY-axes need not be parallel with the corresponding axes on the work coordinate system. In this case, the offset direction is parallel with the PX-axis. If the offset value is positive, the measure in the direction of the PX-axis end point defines the offset. If the offset value is negative, the measure in the direction of the reference point defines the offset.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed

## ● PCHZ (Declare palletizing Z-axis): Only when there are 3 or more axes

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PCHZ | (Axis number) | Prohibited | CP |

| Applicable models |
|---|
| ASEL/PSEL/SSEL ×<br>Other than ASEL/PSEL/SSEL ○ |

Specify the axis number in palletizing Z direction.
Specify the axis number specified in operand 1 as the axis number in palletizing Z direction.
If operand 1 is not specified, the palletizing Z-axis which is specified and already declared becomes invalid.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed

(Note 1) Only Z-axis (either Axis No. 3 or Axis No. 7) in the work coordinate system is available to indicate for the palletizing Z-axis of the SCARA axes. Setting of the palletize Z-axis cannot be established for linear drive axes in PX/QX.

(Note 2) When Coordinate System Definition 1 Control (All Axes Parameter No. 55) is set to 1 valid, and constructing axes (coordinate system definition unit axes) is selected in All Individual Parameter No. 56 (Coordinate System Definition 1 Constructed Axes Setting) in MSEL-PC/PG and TTA, R-axis of the coordinate system definition unit cannot be indicated for the arch motion Z-axis. Also, X and Y-axes of the coordinate system definition unit can be indicated only when the offset of R-axis of the work coordinate system is set to 0.

[Example] PCHZ 3

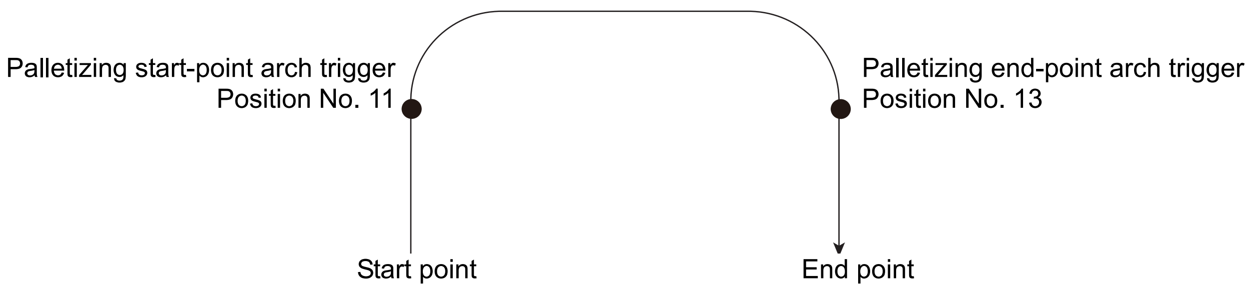
## ● PTRG (Set palletizing arch triggers)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTRG | Position number | Position number | CP |

| Applicable models |
|---|
| ASEL/PSEL/SSEL × <br> Other than ASEL/PSEL/SSEL ○ |



Set arch triggers for arch motion to a palletizing point.
(This command is valid when a PACH command is executed.)
Set as the palletizing start-point arch trigger the palletizing Z-axis (PZ-axis) position data corresponding to the point data specified in operand 1, and set as the palletizing end-point arch trigger the PZ-axis position data corresponding to the point data specified in operand 2.




PTRG 11 13

[Refer to 3.6.5 Palletizing Function]

Among the point data, data of the PZ-axis specified by a PCHZ command must be effective.
Set the arch motion operation through palletizing points in such a way that when the axis rises from the start point, it starts parallel movement after reaching the start-point arch trigger, whereas, when descending, the axis completes parallel movement and then reaches the end-point arch trigger.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed.

(Note 1) Setting of the palletize arch trigger cannot be established for linear drive axes in PX/QX.

## ● PEXT (Set composite palletizing)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PEXT | (Position number) | Prohibited | CP |

| Applicable models |
|---|
| ASEL/PSEL/SSEL ×<br>Other than ASEL/PSEL/SSEL ○ |

Set composite palletizing.
Set the position number specified in operand 1 for setting composite palletizing.
When a palletizing movement command is executed, effective axis data other than data of the PX and PY (and PZ) axes among the specified point data defines the end coordinate of the composite axis.
With SCARA robots, the R-axis becomes a composite palletizing axis.
If nothing is specified in operand 1, the position number already declared for setting composite palletizing becomes invalid.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed.

(Note 1) The palletizing composition axes setting cannot be made to the linear drive axis for PX/QX.

## ● OFPZ (Set palletizing Z-axis offset)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | OFPZ | Offset value | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Set the offset in palletizing Z-axis direction.
Set the value specified in operand 1 as the offset in PZ-axis/palletizing Z-axis direction.
The setting unit of offset is mm. The effective resolution of the set value is 0.001mm.
A negative value can also be set for the offset within the range of operation.
This offset is effective only on the end point of PACH (Arch motion to palletizing point) operation.
An error occurs if this command is executed when BGPA is not yet declared (palletizing setting is not permitted).
If the output is specified, it turns ON after this command has been executed.

(Note) Setting of the palletize Z-axis offset cannot be established for linear drive axes in PX/QX.

## [22] Palletizing Calculation

### ● PTNG (Get palletizing position number)

| Extension condition (LD,A,O,AB,OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PTNG | Palletizing number | Variable number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Assign the palletizing position number for the palletizing number specified in operand 1 to the variable specified in operand 2.
If the output field is specified, the output will turn ON after this command is executed.

## ● PINC (Increment palletizing position number by 1)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PINC | Palletizing number | Prohibited | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Increment by 1 the palletizing position number for the palletizing number specified in operand 1. If the incremented value is considered normal as a palletizing position number calculated under the current palletizing setting, the value will be updated. If not, the value will not be updated.
If the output field is specified, the output will turn ON when the value was successfully incremented, and turn OFF if the increment failed.

## ● PDEC (Decrement palletizing position number by 1)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PDEC | Palletizing number | Prohibited | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Decrement by 1 the palletizing position number for the palletizing number specified in operand 1. If the decremented value is considered normal as a palletizing position calculated under the current palletizing setting, the value will be updated. If not, the value will not be updated.
If the output field is specified, the output will turn ON when the value was successfully decremented, and turn OFF if the decrement failed.

## ● PSET (Set palletizing position number directly)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PSET | Palletizing number | Data | CC |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Set the value specified in operand 2 as the palletizing position number for the palletizing number specified in operand 1.
If the specified value is considered normal as a palletizing position calculated under the current palletizing setting, the value will be set. If not, the value will not be set.
If the output field is specified, the output will turn ON when the palletizing position number was successfully updated, and turn OFF if the update failed.

## ● PARG (Get palletizing angle)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PARG | Palletizing number | Axis number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |



Obtain the palletizing angle.
Calculate the palletizing angle (degrees) from the physical axis specified in operand 2 for the palletizing number specified in operand 1, and store the result in variable 199.

This command need not be executed, if not necessary.
If this command is executed following PAPS (Set palletizing by 3-point teaching), the angle formed by the priority axis and specified axis on the work coordinate system is calculated automatically. An error occurs if this command is executed when PAPS is not yet executed or after PASE has been executed following PAPS.
The axis to be used with a GRP command can be specified before PAPS is executed (refer to the detailed explanation of PAPS). If the effective axis pattern for 3-point teaching data does not match, an “CB0: Mismatched effective axes for palletizing point data by 3-point teaching” error occurs.
If the number of effective point data axes (number of effective axes excluding the PZ-axis (palletizing Z-axis) if the PZ-axis is declared) is less than two, a “CAE: Insufficient effective axes for palletizing point data by 3-point teaching” error occurs. If the number of effective point data axes is greater than two, a “CB9: PX/PY-axis indeterminable error at acquisition of palletizing angle” occurs.
If the axis corresponding to the axis number in operand 2 does not specify one of the two valid axes associated with the point data, an error “CBA, Reference-axis/PX/PY-axis mismatch error at palletizing angle acquisition” will generate.
If the data other than PZ-axis component is identical between the reference point and PX-axis end point in 3-point teaching, a “Reference point/PX-axis end point identical error at acquisition of palletizing angle” occurs and angle calculation is disabled.
If the output field is specified, the output will turn ON after this command is executed.

Definitions related to the angle direction (sign) can be changed in Other Parameter No. 47 “Other Setting Bit Pattern 2” setting for MSEL-PC/PG and TTA.
(MSEL-PC/PG, TTA main application V2.00 or later)
[Refer to the instruction manuals of MSEL-PC/PG and TTA Instruction Manual for details]

## ● PAPG (Get palletizing calculation data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PARG | Palletizing number | Position number | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Store the position coordinate data of the palletizing point corresponding to the palletizing number specified in operand 1, under the position number specified in operand 2.

## [23] Palletizing Movement

### ● PMVP (Move to palletizing points via PTP)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PMVP | Palletizing number | (Position number) | PE |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

Move to the calculated palletizing points via PTP.
The axes will move to the palletizing points specified in operand 1, via PTP.
Executing this command will not increment the palletizing position number by 1.

On controllers other than ASEL, PSEL and SSEL, movement does not occur in directions other than PX/PY-axis directions if the PX/PY-axis coordinates of palletizing points alone are effective (such as when the PZ-axis (palletizing Z axis) is not specified). If the PZ-axis coordinates of palletizing points are also effective, movement occurs in PZ-axis direction.
If a position number is specified in operand 2, however, the palletizing calculation result of Z-direction position is ignored and the axis moves to the height corresponding to the specified position number.
If data of any axis other than the Z-axis specified by palletizing is set under the position number specified in operand 2, such data is ignored. An error handling occurs if no PZ-axis data is available.
If composite palletizing is set, any axis whose data is available, other than the PX-axis and PY-axis (and PZ-axis), also operates.
If operand 2 is specified, the palletizing Z-axis must be declared (PCHZ) in the palletizing setting.
An error occurs if the palletizing Z-axis is not declared.

## ● PMVL (Move to palletizing points via interpolation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PMVL | Palletizing number | (Position number) | PE |

| Applicable models |
|---|
| XSEL-JX/KX and MSEL-PCX/PGX ×<br>Other than XSEL-JX/KX and MSEL-PCX/PGX ○ |

Move to the calculated palletizing points via interpolation.
The axes will move to the palletizing points specified in operand 1, via interpolation.
Executing this command will not increment the palletizing position number by 1.

(Note 1) "Error No. B80 Indication Prohibited Axis Error" will be issued if the palletizing setting to operate the SCARA axes is indicated.
For the palletizing setting at PMVL movement, establish the setting to make the all of the operating axes the liner axes.

If a position number is specified in operand 2, however, the palletizing calculation result of Z-direction position is ignored and the axis moves to the height corresponding to the specified position number.
If data of any axis other than the Z-axis specified by palletizing is set under the position number specified in operand 2, such data is ignored. An error handling occurs if no PZ-axis data is available.
If composite palletizing is set, any axis whose data is available, other than the PX-axis and PY-axis (and PZ-axis), also operates.
Executing this command does not increment the palletizing position by 1.
If operand 2 is specified, the palletizing Z-axis must be declared (PCHZ) in the palletizing setting.
An error occurs if the palletizing Z-axis is not declared.

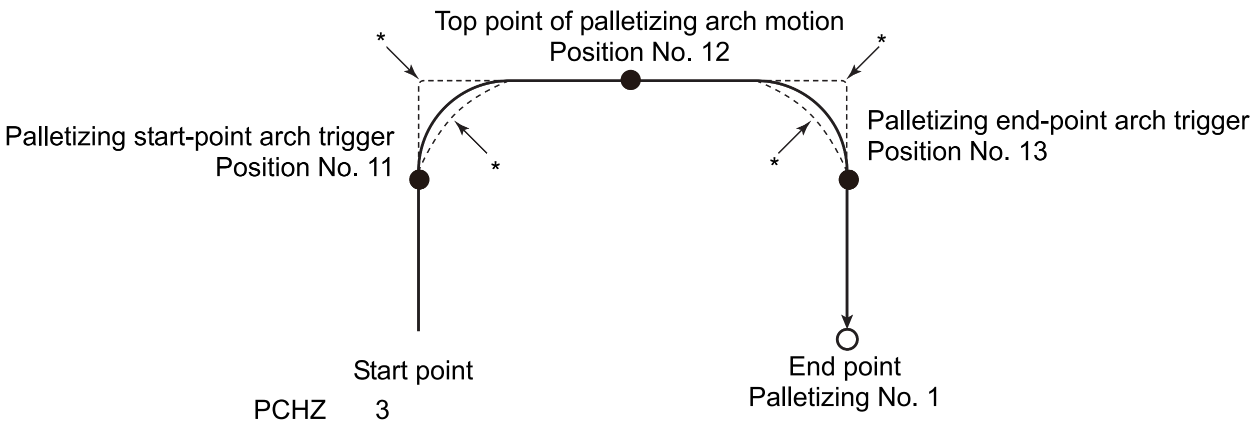
## ● PACH (Arch motion to palletizing point)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | PACH | Palletizing number | Position number | PE |

| Applicable models |
|---|
| ASEL/PSEL/SSEL ×<br>Other than ASEL/PSEL/SSEL ○ |

Perform arch motion from the current point to move to the palletizing points.

- Move via arch motion to the palletizing point specified in operand 1.
- Rise from the current point to palletizing start-point arch trigger and then start moving in PX/PY-axis directions. Pass the top point which is the Z point specified in operand 2, complete the movement in PX/PY-axis directions, and then reach the calculated palletizing point by passing near the palletizing end-point arch trigger.
- Palletizing arch triggers must be set for the PTRG command.




```
PCHZ    3
PTRG    11        13
  -
  -
  -
PACH    1         12
```

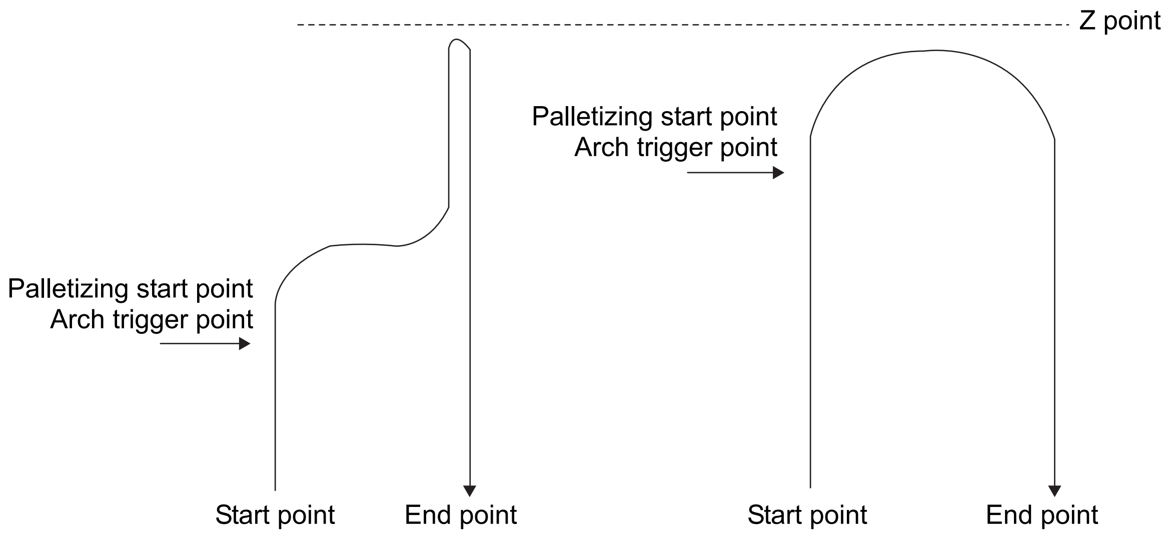
* When the operation is paused and then resumed, the rise operation → horizontal operation composite part and horizontal operation → rise operation composite part follow the paths denoted by * (dotted lines) in the figure depending on the position of resumption. Exercise caution to prevent contact.

(Note 1) When a palletizing setting with the SCARA axes and the liner axes existing together is indicated, "Error No. 421 SCARA and Liner Axes Simultaneous Indication Error" will occur. Also, when a palletizing setting with two units of SCARA axes existing together is indicated, "Error No. B80 Indication Prohibited Axis Error" will occur. Establish the setting to make all the operating axes the same SCARA axes or the linear axes for the palletizing setting at the palletizing point arch motion movement.

(Note 2) The palletize point arch motion operation of SCARA axis is PTP operation and the linear drive axis is CP operation.

(Note 3) The palletize point arch motion operation cannot be performed for linear drive axes in PX/QX.

- The PZ-axis coordinate of the end point corresponds to the PZ-axis component of the position coordinate of the palletizing point, if any, plus the palletizing Z-axis offset. If the PZ component is not available, then the PZ-axis coordinate of the start point, plus the palletizing Z-axis offset, is used. (Normally the offset is added to all applicable positions such as arch trigger and Z points.)
- An error occurs if the palletizing start-point arch trigger is set below the start point, or palletizing end-point arch trigger is set below the end point. (Note: “Above” and “below” have nothing to do with the positive and negative directions of coordinates.)
- The PZ-axis up direction refers to the direction of moving from the start point to Z point (or opposite direction in the case of down direction) and has nothing to do with the magnitude correlation of coordinate values. Accordingly, always check the actual operating directions when this command is used.
- PZ-axis down operation is performed after an up process command value has been output. Accordingly, the following operations may take place depending on how the palletizing arch trigger and Z points are set.




In these cases, change the palletizing arch triggers and PZ point to increase the efficiency of operation.

- If composite palletizing is set (PEXT), any axis whose data is available, other than the PX, PY and PZ-axes, also operates. However, the composite axis starts/ends its operation at a position above the applicable arch trigger. If the R-axis is set with a PEXT command, the R-axis starts/ends its operation above the applicable arch trigger.
- Executing this command does not increment the palletizing position by 1.

## [24] Building of Pseudo-Ladder Task

### ● CHPR (Change task level)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | CHPR | 0 or 1 | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Specify "1" (User HIGH) if you wish the target task to be processed before other tasks.
This command can also be used with non-ladder tasks.
Task level change (0: User NORMAL, 1: User HIGH) is not a required component, but specifying User HIGH will require a TSLP command explained below.
(Without TSLP, tasks of the User NORMAL level will not be processed.)

## ● TPCD (Specify processing to be performed when input condition is not specified)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | TPCD | 0 or 1 | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Specify the processing to be performed when input condition is not specified. (0: Execute, 1: Follow the input condition in the last executed step)
In a ladder task, always input “1” (Follow the input condition in the last executed step) in operand 1.
In a non-ladder task, always input “0” (Execute). (The default value is “0”.)

## ● TSLP (Task sleep)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Prohibited | Prohibited | TCLP | Time | Prohibited | CP |

| Applicable models |
|---|
| All models [Refer to Section 5.1 for details of models] |
| ○ |

[Function] Set the time during which the applicable task will sleep, in order to distribute the processing time to other tasks.
If the task level is set to User HIGH, this command must always be specified.
The applicable task will sleep during the set time.
The time in operand 1 is set in msec.
An appropriate time setting must be examined on the actual system. (Normally approx. 1 to 3 is set.)
(If the ladder statement becomes long, state this command multiple times between steps, as necessary.)
This command can also be used with non-ladder tasks.

[25] Extended Command

## ● ECMD1 (Get motor current value (as percentage of rated current))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 1 | Axis number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | ○ | ○ | ○ | ○ | ○ (TTA only V1.22 or later) | ○ (V1.18 or later) |

[Function] Store the motor current value (percentage of the rated current) corresponding to the “axis number” specified in operand 2, in variable 99.

Note:
- When comparing with “Constant (Non-Pressing) Torque Limit (Upper)” set in Extension Command Code 250, have 5% or more of a margin.

[Example] ECMD 1 2 Extended command 1
Store the motor current value (percentage of the rated current) of axis 2, in variable 99.

## ● ECMD2 (Get home sensor status)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 2 | Axis number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | ○ | ○ | × | × | × |

[Function] Reflect in the output the status of the home sensor corresponding to the “axis number” specified in operand 2.

Note:
- The acquired home sensor status is not the electrical level of H/L, but the operating/non-operating status determined by taking into consideration the setting of axis-specific parameter No. 14, “Home sensor input polarity”.
  If 0 (Not used) is set in axis-specific parameter No. 14, “Home sensor input polarity”, the sensor status (output) is deemed indeterminable and use of the sensor is prohibited.
  The specified output port/flag is operated only when this command has been executed. Accordingly, this command must be executed repeatedly if you want to constantly reflect the sensor status in the output port/flag.

[Example] ECMD 2 3 315 Output the home sensor status of axis 1 in output port No. 315.

## ● ECMD3 (Get overrun sensor status)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 3 | Axis number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | ○ | ○ | × | × | × |



[Function] Reflect in the output the status of the overrun sensor corresponding to the "axis number" specified in operand 2.

Note:
- The acquired overrun sensor status is not the electrical level of H/L, but the operating/non-operating status determined by taking into consideration the setting of axis-specific parameter No. 15, "Overrun sensor input polarity".
  If 0 (Not used) is set in axis-specific parameter No. 15, "Overrun sensor input polarity", the sensor status (output) is deemed indeterminable and use of the sensor is prohibited.
  The specified output port/flag is operated only when this command has been executed. Accordingly, this command must be executed repeatedly if you want to constantly reflect the sensor status in the output port/flag.

[Example] ECMD 3 1 890 Output the overrun sensor status of axis 1 in global No. 890.

## ● ECMD4 (Get creep sensor status)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 4 | Axis number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | ○ | ○ | × | × | × |

[Function] Reflect in the output the status of the creep sensor corresponding to the "axis number" specified in operand 2.

Note:
- The acquired creep sensor status is not the electrical level of H/L, but the operating/non-operating status determined by taking into consideration the setting of axis-specific parameter No. 16, "Creep sensor input polarity".
  If 0 (Not used) is set in axis-specific parameter No. 16, "Creep sensor input polarity", the sensor status (output) is deemed indeterminable and use of the sensor is prohibited.
  The specified output port/flag is operated only when this command has been executed. Accordingly, this command must be executed repeatedly if you want to constantly reflect the sensor status in the output port/flag.

[Example] ECMD 4 2 315 Output the creep sensor status of axis 2 in output port No. 315.

## ● ECMD5 (Get axis operation status)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 5 | Axis number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | ○ | ○ | × | × | ○ |

[Function] Store the status of the axis specified in operand 2, in variable 99.
The axis status is indicated by the ON/OFF level of each bit, as shown below.
Accordingly, the obtained value must be converted to a binary value for interpretation.

| Variable 99 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | | | | | |

- Servo axis in use (0 = Not in use, 1 = In use)
- Home return (00 = Not yet complete, 01 = In progress, 10 = Complete)
- Servo ON/OFF (0 = OFF, 1 = ON)
- Successful execution of movement command (0 = Not yet complete, 1 = Successful)
- Detection of missed work during push-motion operation (0 = Not detected, 1 = Detected)
- Collision detection feature (0 = Not in use, 1 = In use)

* The status of the collision detection feature should be applicable only to XSEL-RAX/SAX/RAXD/SAXD Main Application V1.10 and later.

(Note) If an invalid axis number is specified in Operand 2, CC4 "SEL Data Error" will generate.

[Example] ECMD 5 2 Store the status of axis 2 in variable 99. If 28 (decimal value) was stored in variable 99 after the command was executed, the status of axis 2 is interpreted as follows.

| Variable 99 |
|---|
| 28 |

Binary notation

| Variable 99 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |

Status of axis 2:

- Servo axis not in use
- Home return complete
- Servo ON
- Movement command successful
- Missed work not detected during push-motion operation
- Collision detection feature not used

## ● ECMD6 (Current position acquirement on each axis system (1 axis direct)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 6 | Integer Variable number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | × | × | ○ | ○ (V1.20 or later) | ○ | × | ○ (TTA only) | ○ |

[Function] By using data stored in the four integer variables in a row from the integer variable number indicated in Operation 2, the current position expressed in each axis coordinate system of the indicated axis numbers gets read out to the variable indicated in the current position storage variable number.

• When Operand 2 = Variable number

| Variable No. | Description of setting | I/O |
|---|---|---|
| n | Axis Number | |
| n+1 | Current Position Storage Variable Number | |
| n+2 | 0 | Reserved (to be fixed to 0) |
| n+3 | 0 | Reserved (to be fixed to 0) |

(Note 1) Input an integer variable number in Operation 2.
Local area : 1 to 96, 1001 to 1096
Global area : 200 to 296, 1200 to 1296
(Note) For XSEL-P/Q and some others, there are global domains 20000 to 2796.
[Refer to 4.1 Each Type of Data Available to Handle on the Program and its Range]

(Note 2) The units in the result of the readout of the current position for each axis system are as shown below.
SCARA 1st, 2nd and 4th Axes (5th, 6th, 8th axis) : deg. (degrees)
SCARA 3rd Axis (7th axis) : mm

[Example]

| | | | |
|---|---|---|---|
| LET | 200 | 4 | Set the 4th axis (R-axis) to Variable No. 200 |
| LET | 201 | 300 | Set Current Position Storage Variable No. (300) to Variable No. 201 |
| LET | 202 | 0 | Set 0 to Variable No. 202 |
| LET | 203 | 0 | Set 0 to Variable No. 203 |
| ECMD | 6 | 200 | The current position of each coordinate system on R-axis is read out to Variable No. 300. |

## ● ECMD7 (Get total movement count)

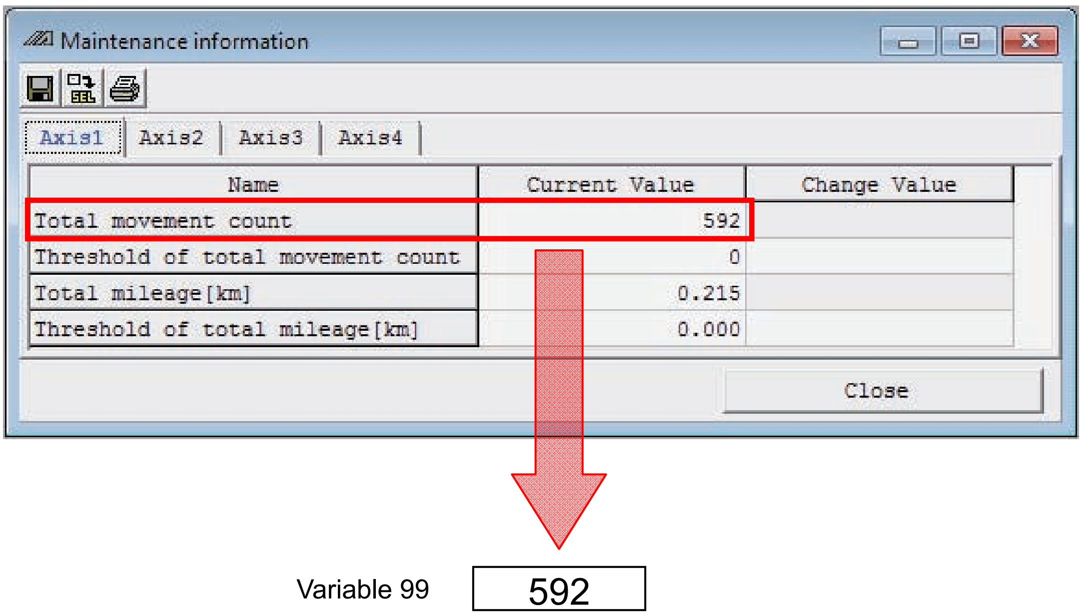
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 7 | Axis number | CC |

| Applicable models |
|---|
| TTA (Main application V2.08 or later)<br>MSEL (Main application V2.08 or later) |

[Function] The total movement count (times) of "Axis number" indicated in operand 2 gets stored in the variable 99.
The total movement count available to obtain by this command is from 0 to 2147483647 times.
What can be obtained by this command is the total movement count in the maintenance information.

[Example] ECMD 7 1 The total movement count of the first axis gets stored in the variable 99.

In the case shown in the figure below, 592 will be stored in the variable 99.



## ● ECMD8 (Get total mileage)

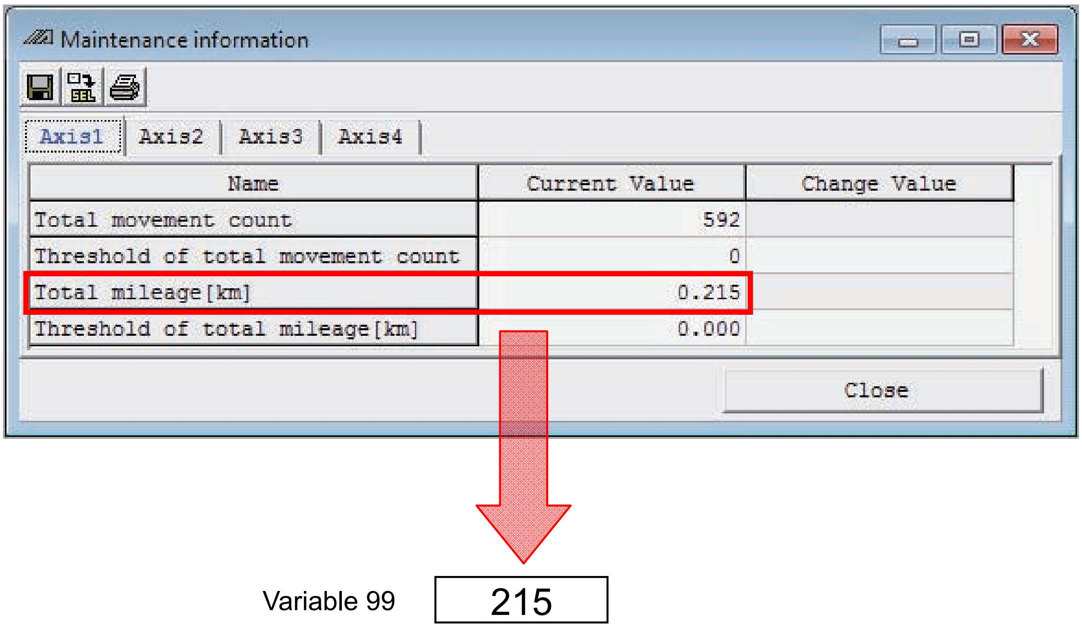
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 8 | Axis number | CC |

| Applicable models |
|---|
| TTA (Main application V2.08 or later)<br>MSEL (Main application V2.08 or later) |

[Function] The total mileage (unit in m if linear drive and in 1000deg if rotary drive) of “Axis number” indicated in operand 2 gets stored in the variable 99.
What can be obtained by this command is the total mileage in the maintenance information.
The total mileage available to obtain by this command is from 0 to 2147483647m.

[Example] ECMD 8 1 The total mileage of the first axis gets stored in the variable 99.

In the case shown in the figure below, 215 will be stored in the variable 99.



## ● ECMD9 (Get position deviation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 9 | Axis number | CC |

| Applicable models |
|---|
| TTA (Main application V2.10 or later)<br>MSEL (Main application V2.10 or later) |

[Function] The position deviation (encoder pulse unit) of "Axis number" indicated in operand 2 gets stored in the variable 99.

[Example] ECMD 9 1 The position deviation of the first axis gets stored in the variable 99.

## ● ECMD10 (Acquirement of Overload Level)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | ECMD | 10 | Axis number | CC |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later.) |



[Function] The overload level (unit in %) of "Axis Number" indicated in Operation 2 should be stored in Variable 99.
"D0A: Driver Overload Error" will be generated if the overload level exceeds 100%. In order to reduce the overload level, it is effective to decrease the acceleration setting or increase the rate of pause.

[Example] ECMD 10 1 Overload level of 1st axis should be stored in Variable 99

## ● ECMD11 (Acquirement of Encoder Overheated Level)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | ECMD | 11 | Axis number | CC |

| Applicable models |
|---|
| XSEL-RA/SA/RAX/SAX/RAXD/SAXD (Main application V1.10 or later.) |

[Function] The encoder overheated level (unit in %) of “Axis Number” indicated in Operation 2 should be stored in Variable 99.
“5C7: Encoder Overheated Error” will be generated if the encoder overheated level exceeds 100%. In order to reduce the encoder overheated level, it is effective to decrease the acceleration setting or increase the rate of pause.

(Note) If this command gets executed to an axis with an encoder not equipped with a thermo sensor, 0 should be stored to Variable 99.

[Example] ECMD 11 1 Encoder overheated level of 1st axis should be stored in Variable 99

## ● ECMD20 (Get parameter value)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 20 | Variable number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | ○ | ○ | ○ | × | ○ |

[Function] Store the value of the specified parameter in variable 99, using the data stored in the three consecutive variables starting from the one corresponding to the variable number specified in operand 2.
The contents and ranges for the variable data settings are as shown below. Setting outside the specified range will generate “CC4 SEL data error”.

• When Operand 2 = n

| Variable No. | Description of setting | Setting value and range for each variable | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | I/O | Common to all axes | Axis-specific | Driver | Encoder | I/O device | Other |
| n | Parameter type | 0 | 1 | 2 | 3 | 4 | 5 | 7 |
| n+1 | Device number/axis number | 0 | 0 | 1 to 8* (up to number of connected axes) | 1 to 8* (up to number of connected axes) | 1 to 8* (up to number of connected axes) | 0 to 9 | 0 |
| n+2 | Parameter number | 1 to 999 | 1 to 400 | 1 to 250 | 1 to 112 | 1 to 30 | 1 to 112 | 1 to 200 |

Specify an integer variable in operand 2 (At this time, make sure that three variables can be ensured.). If a variable of non-integer type is specified, “C3C, Variable number error” will generate.

(Note) Setting of Parameter Type = 10 enables to acquire parameters for the pulse I/O board. [See the next page.]

[Example]

| | | | |
|---|---|---|---|
| LET | 1250 | 0 | Variable No. 1250 = Parameter type (I/O) |
| LET | 1251 | 0 | Variable No. 1251 = Device number (0, in the case of I/O parameter) |
| LET | 1252 | 30 | Variable No. 1252 = Parameter number (No. 30) |
| ECMD | 20 | 1250 | Extended command 20 (Use variable No. 1250 through 1252) Store the value of I/O parameter No. 30, “Input function selection 000”, in variable 99. |

## ● ECMD20 (Get parameter value (Extension Motion Control Board Parameters)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 20 | Variable number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | × | × | × | × | × | × |



[Function] Store the value of the specified parameter in variable 99, using the data stored in the five consecutive variables starting from the one corresponding to the variable number specified in Operand 2.
The contents and ranges for the variable data settings are as shown below. Setting outside the specified range will generate “C44 SEL data error”.
When Parameter Type = 10 (Pulse I/O board: Feature dedicated for XSEL , (XSEL-RA/SA excluded) is selected, Parameter Detail Type (n + 3) and Channel Number (n + 4) are required of following.

• When Operand 2 =n

| Variable No. | Description of setting | Setting value and range for each variable |
|---|---|---|
| n | Parameter type | 10: Pulse I/O board |
| n+1 | Device number | 0 to 1 |
| n+2 | Parameter number | 1 to 100 |
| n+3 | Parameter detail type | 0 : Common<br>1 : Input Channel<br>2 : Output Channel |
| n+4 | Channel number | The range may differ depending on parameter detail type (n+3).<br>Parameter detail type (n+3) / Range for channel number (n+4)<br>0 (Common) / Reserved (to be fixed to 0)<br>1 (Input channel) / 0 to 1<br>2 (Output channel) / 0 to 7 |

Specify an integer variable in operand 2. At this time, make sure to secure five (for XSEL) consecutive variables. C3C will be generated when a variable other than integer variables is indicated.

[Example]

| | | | |
|---|---|---|---|
| LET | 1250 | 10 | Variable No. 1250 = Parameter type (Pulse I/O board) |
| LET | 1251 | 1 | Variable No. 1251 = Device number |
| LET | 1252 | 2 | Variable No. 1252 = Parameter number (No. 2) |
| LET | 1253 | 2 | Variable No. 1252 = Parameter Detail Type (Output Channel) |
| LET | 1253 | 5 | Variable No. 1252 = Channel number (Output channel 5) |
| ECMD | 20 | 1250 | Extended command 20 (Use variable No. 1250 through 1252)<br>Data for the value set in No. 2 (Pulse output mode) in the pulse I/O board parameter output channel 5 is stored to Variable 99. |

## ● ECMD250 (Set torque limit/torque limit over detection time)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | ECMD | 250 | Integer variable number | CC |

| Applicable models | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/QCT | XSEL -R/S | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/SXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | ○ | ○ | × SSEL ○ (V0.52 or later) | × | Δ(Note1) |

Note 1 Appliance to High-Resolution
Applicable if main application V2.08 or later and driver V1.0 or later

[Function] Set the steady-state (non-push) torque limit (upper limit)/steady-state (non-push) torque limit over detection time. Use the data stored in three successive integer variables, starting from the integer variable number specified in operand 2, to temporarily change the applicable parameters (including internal parameters).
Operand 2 = n

Variable No. n ----- Target axis pattern (decimal entry)
* Example of decimal entry: 1 = Axis 1 only
2 = Axis 2 only
3 = Axes 1 and 2
7 = Axes 1, 2 and 3
15 = Axes 1, 2, 3 and 4

Variable No. n+1 = Set value of steady-state (non-push) torque limit (upper limit)
(1% or more of the rating to the value set in driver card parameter No. 40, "Maximum torque limit (%) Except, individual upper limit for each axis for MSEL")
* If the set value is greater than the upper limit specific to each axis, the upper limit specific to the axis is set.

Variable No. n+2 = Set value of steady-state (non-push) torque limit over detection time
(0 to 20000msec)
* Set 1 or greater if you want to use this command to "detect a contact/heavy load" or move an axis.
* If 0 is set, the detection time becomes invalid (infinite). This setting is used mainly to "limit the torque of the supporting axis (horizontal only) in fitting application". If 0 (infinite) is set, the "steady-state (non-push) torque limit (upper limit)" is limited to a maximum of 70% to prevent overheating.

Variable No. n+3 = 0 is set. (Reserved. * May be made accessible in the future.)

Variable No. n+4 = 0 is set. (Reserved. * May be made accessible in the future.)
If a command specifying the "steady-state (non-push) torque limit (upper limit)" has remained effective for the "steady-state (non-push) torque limit over detection time" or longer in steady state (not pushing), appropriate processing is performed based on the parameter below. Note that processing based on the following parameter is not performed if the "steady-state (non-push) torque limit over detection time" is set to 0 (infinite):
All-axis parameter No. 19, "Type of processing upon steady-state (non-push) torque limit over (priority on overload and other driver errors)"
0: Operation-cancellation level error (Recommended)
(Error No. 420: Steady-state (non-push) torque limit over error)
1: Operation cancellation (SEL command output = OFF)

| | | | | |
|---|---|---|---|---|
| [Example 1] | LET | 290 | 3 | Set the target axis pattern (axes 1 and 2) in integer variable 290. |
| | LET | 291 | 80 | Set the steady-state torque limit in integer variable 291. |
| | LET | 292 | 1000 | Set the steady-state torque limit over detection time in integer variable 292. |
| | ECMD | 250 | 290 | Read the values of three successive variables, starting from variable 290.<br>Set axes 1 and 2.<br>Steady-state torque limit = 80%, steady-state torque limit over detection time = 10000msec |
| | MOVP | 2 | | Move to position No. 2 under the condition set by ECMD250. |

* To return to a normal state:

| | | | | |
|---|---|---|---|---|
| [Example 2] | LET | 290 | 3 | Set the target axis pattern (axes 1 and 2) in integer variable 290. |
| | LET | 291 | 1000 | Set the steady-state torque limit in integer variable 291 (specification of the upper limit specific to each axis). |
| | LET | 292 | 20000 | Clear the steady-state torque limit over detection time in integer variable 292. (Clear 20000.) |
| | STOP | *290 | | Clear the low-torque axis deviation counter. |
| | ECMD | 250 | 290 | Read the values of three successive variables, starting from variable 290.<br>Steady-state torque limit = Upper limit specific to each axis (maximum torque return)<br>Steady-state torque limit over detection time (20000msec) |
| | MOVP | 2 | | Move to position 2 at the steady-state torque. |

(Note 1) If the torque is set low, dropping (vertical axis, etc.) and overshooting occurs. If the torque is lowered during high-speed operation, overshooting occurs due to insufficient torque.

(Note 2) If the torque is lowered during high-speed operation, normal deceleration cannot be performed due to insufficient torque and overshooting occurs as a result, creating a dangerous situation.

(Note 3) If positioning operation is performed at low torque, the axis may remain stopped near the positioning target due to insufficient torque.
When moving an axis, be sure to set the “steady-state (non-push) torque limit over detection time” to 1msec or longer to detect a steady-state (non-push) torque limit over event (timeout).

* If the “steady-state (non-push) torque limit over detection time” is set to 0 to “limit the torque of the supporting axis (horizontal only) in fitting application”, positioning operation to the coordinate of the torque-limited axis is performed when returning after the fitting operation, if the position data for return operation after the fitting operation (via a PUSH command, etc.) includes the coordinate of the supporting axis (torque-limited axis) in fitting application.
  As a result, the axis may remain stopped near the target position due to insufficient torque. For the position data used in the return operation after the fitting operation, set only the coordinate of the fitting operation axis (axis used by a PUSH command, etc.).

(Note 4) If the torque is set extremely low, servo ON axes may move at very slow speed due to an analog offset error, etc.

(Note 5) Even when the load is normal, the torque becomes slightly higher during acceleration/deceleration. Determine appropriate settings (steady-state torque limit and steady-state torque limit over detection time) to prevent false detection of steady-state torque limit over events.

(Note 6) "Error No. C6B: Deviation overflow error" or "Error No. CA5: Stop deviation overflow error" may be detected before "Error No. 420: Steady-state (non-push) torque limit over error". This is normal.

(Note 7) If the torque is changed to a high level from a low level at which axis movement can no longer be guaranteed, be sure to issue a STOP command to low-torque axes and clear the deviation counter before increasing the torque (from a low level). If the torque setting is changed from low to high when deviations are still accumulated, the axes may move without their speed being limited and thus a dangerous situation may occur.

(Note 8) To return to the normal condition (maximum torque), expressly specify 1000% for the "steady-state (non-push) torque limit (upper limit)" and 20000msec for the "steady-state (non-push) torque limit over detection time".

* If a value greater than the upper limit specific to each axis is set for the "steady-state (non-push) torque limit (upper limit)" of that axis, the upper limit specific to the axis (approx. 100 to 400%) is set.

(Note 9) The following values are used upon power ON reset, software reset and start of home return:
Steady-state (non-push) torque limit (upper limit) = Driver card parameter No. 40, "Maximum toque limit (%) Except, individual upper limit for each axis for MSEL"
Steady-state (non-push) torque limit over detection time = 20000msec

(Note 10) If the "steady-state (non-push) torque limit (upper limit)" and "steady-state (non-push) torque limit over detection time" are changed, the new settings will remain effective even after the SEL program ends.
When building a system using this extended command, therefore, expressly set the "steady-state (non-push) torque limit (upper limit)" and "steady-state (non-push) torque limit over detection time" in all SEL programs, before any operation is started in each program, using this extended command.
If you assume that the "steady-state (non-push) torque limit (upper limit)" and "steady-state (non-push) torque limit over detection time" will be reset after the end of operation in other programs, unexpected settings of "steady-state (non-push) torque limit (upper limit)" and "steady-state (non-push) torque limit over detection time" may be applied should the program abort due to an error, etc., in which case unforeseen problems may result.

(Note 11) This extended command does not rewrite the value of driver card parameter No. 40, "maximum torque limit" (except, individual upper limit for each axis for MSEL) itself (main CPU flash memory) (in non-volatile memory).

(Note 12) The motion taken when the controller is not applicable is as stated below.

| Conditon | Action |
|---|---|
| When main application part is in a version not applicable | "B16: SEL Operand Indication Error" should occur |
| When built-in PC board is not applicable | "C7A: Servo Unsupported Feature Error" should occur |
| When driver is in a version not applicable | "B25: Driver Unsupported Feature Error" should occur |

## ● ECMD280/290 (Conversion from each axis coordinates to work coordinates in wrist unit equipped robot)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | ECMD | 280/290 | Integer variable number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/ TTA | MSEL |
| × | × | × | ○ | × | × | × | × | × | × | × |

[Function] The position in each axis coordinate system in a robot equipped with a wrist unit should be converted into the position in the work coordinate system.

■ Operand 2: Contents in Integer Variable No. n

| Variable No. | Data | Contents |
|---|---|---|
| n | Top position data number for storage of data before conversion | Indicate the position number that the each axis coordinate system position data to be converted was stored. (*1) |
| n+1 | Top position data number for storage of data after conversion | Indicate the position number that the converted work coordinate system position data is to be stored. (*1) |
| n+2 | Top position data number for storage of work coordinate offset | Indicate the position number that the work coordinate offset was stored. (*1) |
| n+3 | Top position data number for storage of tool coordinate offset | Indicate the position number that the tool coordinate offset was stored. (*1) |
| n+4 | Robot combination type | Refer to "5.3.6. Settings in Common for Coordinate Conversion Commands". |
| n+5 | Wrist unit type | 0: S type 1: M type |
| n+6 | Top axis number | Indicate the top axis number of the position that the data was stored.<br>* Only ECMD 290 is effective. -1 should always be indicated for ECMD 280. |

* 1 The top position number should be indicated for the standard motion control position.

* Work Coordinate Offset
  For TTA and MSEL, set the work coordinate offset amount in the work coordinate system number and execute SLWK Command, and the position will be at the point that is offset by the amount set in the work coordinate system number.
  In the same manner, if the offset amount is set in the position of the work coordinate offset storage top position data number and the work coordinate offset storage top position data number is indicated, the position after offset will be figured out.
  [Refer to 1.4.5 (2) Positioning on Work Coordinate System]

* Tool Coordinate Offset
  For TTA and MSEL, set the tool coordinate offset amount in the tool coordinate system number and execute SLTL Command, and the position will be at the point that is offset by the amount set in the tool coordinate system number.
  In the same manner, if the offset amount is set in the position of the tool coordinate offset storage top position data number and the tool coordinate offset storage top position data number is indicated, the position after offset will be figured out.
  [Refer to 1.4.5 (2) Positioning using Tool Coordinate System Offset]

## ● ECMD281/291 (Conversion from work coordinates to each axis coordinates in wrist unit equipped robot)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | ECMD | 281/291 | Integer variable number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/ TTA | MSEL |
| × | × | × | ○ | × | × | × | × | × | × | × |

[Function] The position in work coordinate system in a robot equipped with a wrist unit should be converted into the position in the each axis coordinate system.

■ Operand 2: Contents in Integer Variable No. n

| Variable No. | Data | Contents |
|---|---|---|
| n | Top position data number for storage of data before conversion | Indicate the position number that the position data to be converted was stored. (* 1) The work coordinate system position data to be converted and the each axis coordinate system current position data will be stored in the positions in a row. |
| n+1 | Form to be converted | Indicate the robot form to be converted.<br>0: Flip 4: Non Flip 8: Form Automatic Select *<br>* Form to have R-axis angle (absolute value) minimum selected. |
| n+2 | Top position data number for storage of data after conversion | Indicate the position number that the converted each axis coordinate system position data is to be stored. (* 1) |
| n+3 | Top position data number for storage of work coordinate offset | Indicate the position number that the work coordinate offset was stored. (* 1) |
| n+4 | Top position data number for storage of tool coordinate offset | Indicate the position number that the tool coordinate offset was stored. (* 1) |
| n+5 | Robot combination type | Refer to "5.3.6. Settings in Common for Coordinate Conversion Commands". |
| n+6 | Wrist unit type | 0: S type 1: M type |
| n+7 | Top axis number | Indicate the top axis number of the position that the data was stored.<br>* Only ECMD 291 is effective. -1 should always be indicated for ECMD 281. |

* 1 The top position number should be indicated for the standard motion control position.

(Note 1) Indicate the form automatic select when it is not necessary to specify the form for the form to be converted. In this, the form to have the R-axis angle (absolute value) minimum will be selected, which enables to have the movement time short.

(Note 2) When the form automatic select is indicated, and if the R-axis is an absolute value at the same angle, NON FLIP will be selected.

* Work Coordinate Offset
For TTA and MSEL, set the work coordinate offset amount in the work coordinate system number and execute SLWK Command, and the position will be at the point that is offset by the amount set in the work coordinate system number.
In the same manner, if the offset amount is set in the position of the work coordinate offset storage top position data number and the work coordinate offset storage top position data number is indicated, the position after offset will be figured out.
[Refer to 1.4.5 (2) Positioning on Work Coordinate System]

* Tool Coordinate Offset
For TTA and MSEL, set the tool coordinate offset amount in the tool coordinate system number and execute SLTL Command, and the position will be at the point that is offset by the amount set in the tool coordinate system number.
In the same manner, if the offset amount is set in the position of the tool coordinate offset storage top position data number and the tool coordinate offset storage top position data number is indicated, the position after offset will be figured out.
[Refer to 1.4.5 (2) Positioning using Tool Coordinate System Offset]

## ● ECMD282/292 (Conversion from tool coordinates to work coordinates in wrist unit equipped robot)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | ECMD | 282/292 | Integer variable number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/ TTA | MSEL |
| × | × | × | ○ | × | × | × | × | × | × | × |

[Function] The position in the tool coordinate system on a robot equipped with a wrist unit (relative position from the tool tip) should be converted into the position in the work coordinate system.

■ Operand 2: Contents in Integer Variable No. n

| Variable No. | Data | Contents |
|---|---|---|
| n | Top position data number for storage of data before conversion | Indicate the position number that the position data to be converted was stored. (* 1) The tool coordinate system position data to be converted and the work coordinate system current position data will be stored in the positions in a row. |
| n+1 | Top position data number for storage of data after conversion | Indicate the position number that the converted work coordinate system position data is to be stored. (* 1) |
| n+2 | Top position data number for storage of work coordinate offset | Indicate the position number that the work coordinate offset was stored. (* 1) |
| n+3 | Top position data number for storage of tool coordinate ffset | Indicate the position number that the tool coordinate offset was stored. (* 1) |
| n+4 | Robot combination type | Refer to “5.3.6. Settings in Common for Coordinate Conversion Commands”. |
| n+5 | Wrist unit type | 0: S type 1: M type |
| n+6 | Top axis number. | Indicate the top axis number of the position that the data was stored.<br>* Only ECMD 292 is effective. -1 should always be indicated for ECMD 282. |

* 1 The top position number should be indicated for the standard motion control position.

* Work Coordinate Offset

For TTA and MSEL, set the work coordinate offset amount in the work coordinate system number and execute SLWK Command, and the position will be at the point that is offset by the amount set in the work coordinate system number.
In the same manner, if the offset amount is set in the position of the work coordinate offset storage top position data number and the work coordinate offset storage top position data number is indicated, the position after offset will be figured out.
[Refer to 1.4.5 (2) Positioning on Work Coordinate System]

* Tool Coordinate Offset
  For TTA and MSEL, set the tool coordinate offset amount in the tool coordinate system number and execute SLTL Command, and the position will be at the point that is offset by the amount set in the tool coordinate system number.
  In the same manner, if the offset amount is set in the position of the tool coordinate offset storage top position data number and the tool coordinate offset storage top position data number is indicated, the position after offset will be figured out.
  [Refer to 1.4.5 (2) Positioning using Tool Coordinate System Offset]

## ● ECMD300 (User system error output)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | ECMD | 300 | Integer variable number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/ TTA | MSEL |
| × | × | × | ○ | × | × | × | ○ | × | × | × |

[Function] If a system error (such as communication error to another controller and timeout) is detected, a user system error (Error No. A00 (message level) or No. C00 (operation cancel level)) can be caused by executing this command. Also, in case that program stop is required by detecting a system error, the program can be stopped by executing this command.
Set the error information to the integral numbers from No. n+1 to No. n+4 set in operand 2 and execute this command, and the information can be displayed in Info. 1 to Info. 4 in the error list. The output part should turn ON if executed in normal condition when Output Error Level = 0 (message level) and be alway OFF when Output Error Level = 1 (operation cancel level).

■ Operand 2: Contents in Integer Variable No. n

| Variable No. | Data | Contents |
|---|---|---|
| n | Output error level | Indicate the error level to output.<br>0: message level (* 1)<br>(Error No. A00 “User System Error” occurred.)<br>1: operation cancel level<br>(Error No. C00 “User System Error” occurred.) |
| n+1 | Error detail information 1 | Set value should be displayed in Info. 1 in the error list. |
| n+2 | Error detail information 2 | Set value should be displayed in Info. 2 in the error list. |
| n+3 | Error detail information 3 | Set value should be displayed in Info. 3 in the error list. |
| n+4 | Error detail information 4 | Set value should be displayed in Info. 4 in the error list. |

* 1 Even if an occurred error is message level, it will be registered in the error list.

## [26] RC gateway function commands

### ● RPGT (Read RC-axis position data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPGT | RC-axis number | Position number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Read the RC-axis position into variable 199.

[Example 1] RPGT 1 2 Read the position corresponding to RC position No. 2 of axis 1 into variable 199.

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.3 | 0 | 0.10 |
| 2 | 200.00 | 300 | 0.3 | 0 | 0.10 |

→ 200.00 is stored in variable 199.

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| RPGT | *1 | *2 | Read into variable 199 the RC position corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2. |

## ● RPPT (Write RC-axis position data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPPT | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Write the value of variable 199 to the position corresponding to the specified position data [mm].

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 150 | Assign 150 to variable 199. |
| RPPT | 1 | 2 | Write the content of variable 199, or 150, to RC position No. 2 of axis 1. |

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.3 | 0 | 0.10 |
| 2 | 150.00 | 300 | 0.3 | 0 | 0.10 |

| Variable 199 | 150 |
|---|---|

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 15 | Assign 150 to variable 199. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| RPPT | *1 | *2 | Write the content of variable 199, or 150, to the RC position corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2. |

## ● RPCR (Clear RC-axis position data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPCR | RC-axis number | Variable number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Clear position data in the range specified by variable No. n and variable No. n+1. After the data is cleared, the fields become blank.

| Variable | Description of setting |
|---|---|
| n | Clear start position number |
| n+1 | Clear end position number |

[Example 1]

| | | | |
|---|---|---|---|
| LET | 200 | 0 | Assign 0 to variable 200. |
| LET | 201 | 1 | Assign 1 to variable 201. |
| RPCR | 1 | 200 | Clear 1 from position No. 0 of axis 1. |

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | | | | | |
| 1 | | | | | |
| 2 | 200.00 | 300 | 0.3 | 0 | 0.10 |

Cleared.

5. SEL Commands

## ● RPCP (Copy RC-axis position data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPCP | RC-axis number | Variable number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Copy the position data specified by variable No. n and variable No. n+1.

| Variable | Description of setting |
|---|---|
| n | Position number to copy data to |
| n+1 | Position number to copy data from |

| [Example 1] | LET | 200 | 2 | Assign 2 to variable 200. |
|---|---|---|---|---|
| | LET | 201 | 0 | Assign 0 to variable 201. |
| | RPCP | 1 | 200 | Copy the data of position No. 0 of axis 1 specified by the variable, to position No. 2. |

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 100 | 0.2 | 0 | 0.20 |
| 1 | 380.00 | 300 | 0.3 | 0 | 0.10 |
| 2 | 5.00 | 100 | 0.2 | 0 | 0.20 |

Copy

## ● RPRD (Read current RC-axis position)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPRD | Position number | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |



[Function] Read into a position number the current position of each axis specified by an RAXS command.

Important note: Before executing this command, set an axis pattern using an RAXS command. If not, a “(43B) RC-axis pattern not-set error” occurs.

[Example 1]

| | | | |
|---|---|---|---|
| RAXS | 0 | 11 | Set an axis pattern consisting of axes 0, 1 and 2. |
| RPRD | 100 | | Read the current positions of axes 0 to 2 into RC position No. 100. |

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 100 | 5.00 | 300 | 0.3 | 0 | 0.10 |

The current position of axis 1

Position data of axis 2

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 100 | 500.00 | 200 | 0.3 | 0 | 0.10 |

The current position of axis 2

Position data of axis 3

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 100 | 100.00 | 300 | 0.3 | 0 | 0.10 |

The current position of axis 3

[Example 2]

| | | | |
|---|---|---|---|
| RAXS | 0 | 111 | Set an axis pattern consisting of axes 0, 1 and 2. |
| LET | 1 | 100 | Set 100 in variable 1. |
| RPRD | *1 | | Read the current positions of axes 0 to 2 into the RC position corresponding to the content of variable 1, or 100. |

## ● RPRQ (Read current RC-axis position (single-axis direct))

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | ○ Can be used |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPDQ | RC-axis number | Variable number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Read the current position of the RC-axis into the variable specified in operand 2. The current position can be acquired faster than when a RPRD command is used.

[Example] RPRQ 2 100 Read the current position of axis 2 into variable No. 100.

## ● RPVL (Write RC-axis speed data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPVL | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Write the value of variable 199 to the speed [mm/s] corresponding to the position data specified in operand 2.

[Example 1]

```
LET    199  100   Assign 100 to variable 199.
RPVL   1    2     Write the speed in variable 199, or 100mm/s, to RC
                  position No. 2 of axis 1.
```

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.3 | 0 | 0.10 |
| 2 | 200.00 | 100 | 0.3 | 0 | 0.10 |

| Variable 199 | 100 |
|---|---|

[Example 2]

```
LET    199  100   Assign 100 to variable 199.
LET    1    2     Assign 2 to variable 1.
LET    2    3     Assign 3 to variable 2.
RPVL   *1   *2    Write the speed in variable 199, or 100mm/s, to the
                  RC position number corresponding to the content of
                  variable 2, or 3, of the axis corresponding to the
                  content of variable 1, or 2.
```

5. SEL Commands

## ● RPAD (Write RC-axis acceleration/deceleration data)

*1 439 RC Position Data Use Method Error when a command was executed

| | | |
|---|---|---|
| RC position-data use mode | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPAD | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Write the value of variable 199 to the acceleration/specification [G] corresponding to the position data specified in operand 2.

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 0.1 | Assign 0.1 to variable 199. |
| RPAD | 1 | 2 | Write the acceleration/specification in variable 199, or 0.1G, to RC position No. 2 of axis 1. |

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.3 | 0 | 0.10 |
| 2 | 200.00 | 300 | 0.1 | 0 | 0.10 |

| Variable 199 | 0.1 |
|---|---|

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 0.3 | Assign 0.3 to variable 199. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| RPAD | *1 | *2 | Write the speed in variable 199, or 0.3G, to the RC position number corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2. |

## ● RPIP (Write RC-axis in-position width data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPIP | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Write the value of variable 199 to the in-position width [mm] corresponding to the position data specified in operand 2.

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 0.2 | Assign 0.2 to variable 199. |
| RPIP | 1 | 2 | Write the in-position band in variable 199, or 0.2mm, to RC position No. 2 of axis 1. |

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.3 | 0 | 0.10 |
| 2 | 200.00 | 300 | 0.3 | 0 | 0.20 |

| Variable 199 | 0.2 |
|---|---|

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 0.2 | Assign 0.2 to variable 199. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| RPIP | *1 | *2 | Write the in-position width in variable 199, or 0.2mm, to the RC position number corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2. |

## ● RPTQ (Write RC-axis current-limiting value data for push-motion operation)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPTQ | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Write the value of variable 199 to the current-limiting value for push-motion operation [%] corresponding to the position data specified in operand 2.

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 50 | Assign 50 to variable 199. |
| RPTQ | 1 | 2 | Write the current-limiting value in variable 199, or 50%, to RC position No. 2 of axis 1. |

Position data of axis 1

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.3 | 0 | 0.10 |
| 2 | 200.00 | 300 | 0.3 | 50 | 0.10 |

| Variable 199 | 50 |
|---|---|

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 50 | Assign 50 to variable 199. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| RPTQ | *1 | *2 | Write the current-limiting value in variable 199, or 50%, to the RC position number corresponding to the content of variable 2, or 3, of the axis corresponding to the content of variable 1, or 2. |

## ● RGVL (Read RC-axis speed data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RGVL | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Read into variable 199 the speed [mm/s] corresponding to the position data specified in operand 2.

[Example] RGVL 2 1 Read into variable 199 the speed specified under RC position No. 1 of axis 2.

Position data of axis 2

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 200 | 0.3 | 0 | 0.10 |

| Variable 199 | 200 |
|---|---|

## ● RGAD (Read RC-axis acceleration/deceleration data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RGAD | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Read into variable 199 the acceleration/deceleration [G] corresponding to the position data specified in operand 2.

[Example 1] RGAD 2 1 Read into variable 199 the acceleration/deceleration specified under RC position No. 1 of axis 2.

Position data of axis 2

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.2 | 0 | 0.10 |

→

| Variable 199 | 0.2 |
|---|---|

## ● RGIP (Read RC-axis in-position width data)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RGIP | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Read into variable 199 the in-position width [mm] corresponding to the position data specified in operand 2.

[Example] RGIP 2 1 Read into variable 199 the in-position width specified under RC position No. 1 of axis 2.

Position data of axis 2

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.2 | 0 | 0.10 |

| Variable 199 | 0.10 |
|---|---|

5. SEL Commands

## ● RGTQ (Read RC-axis current-limiting value data for push-motion operation)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used[*1] |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RGTQ | RC-axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Read into variable 199 the current-limiting value for push-motion operation [%] corresponding to the position data specified in operand 2.

[Example] RGTQ 2 1 Read into variable 199 the current-limiting value specified under RC position No. 1 of axis 2.

Position data of axis 2

| No. | Pos | Vel | Acc | Push | Inp |
|---|---|---|---|---|---|
| 0 | 5.00 | 300 | 0.3 | 0 | 0.10 |
| 1 | 380.00 | 300 | 0.2 | 30 | 0.10 |

| Variable 199 | 30 |
|---|---|

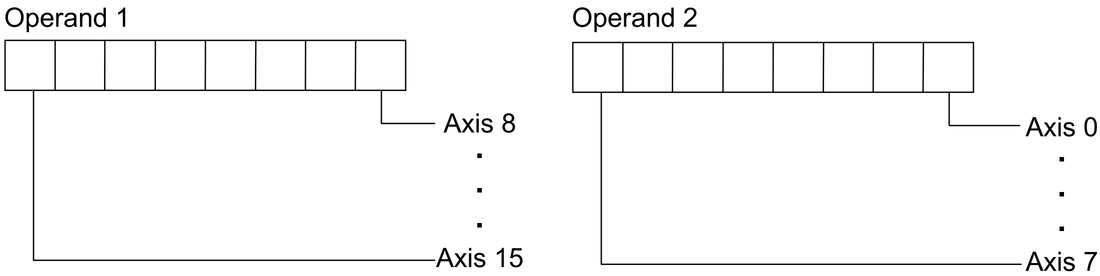
## ● RAXS (Set RC-axis pattern)

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | ○ Can be used |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RAXS | Axis pattern, upper | Axis pattern, lower | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Set an axis pattern covering axes 8 to 15 in operand 1, and axis pattern covering axes 0 to 7 in operand 2.
The axes set by the axis pattern are operated simultaneously.




Always set an axis pattern if the commands listed below are used.
(Set 1 for the axis numbers used, and 0 for the axis number not used.)
If an axis pattern is not set, a “(43B) RC-axis pattern not-set error” occurs:

- RPRD : Read current RC-axis position
- RSON : Turn ON RC-axis servo
- RSOF : Turn OFF RC-axis servo
- RHOM : Return RC-axis to home
- RMVP : Move RC-axis by position specification
- RMPI : Move RC-axis incrementally by position specification
- RSTP : Decelerate RC-axis to stop

| | | | | |
|---|---|---|---|---|
| [Example] | RAXS | 1010101 | 10101010 | Set an axis pattern consisting of axes 1, 3, 5, 7, 8, 10, 12 and 14. |
| | RSON | | | Turn ON the servos of the specified axes. |
| | RMVP | 20 | | Move the specified axes to the positions corresponding to position No. 20. |

## ● RSON (Turn ON RC-axis servo)

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | ○ Can be used |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RSON | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Turn ON the servo of each RC-axis specified by an RAXS command.

> Important note: Before executing this command, set an axis pattern using an RAXS command.
> If not, a "(43B) RC-axis pattern not-set error" occurs.

[Example] RAXS 0 1100 Set an axis pattern that uses axes 2 and 3.

RSON Turn ON the servos of the specified axes.

## ● RSOF (Turn OFF RC-axis servo)

| | | |
|---|---|---|
| RC position-data use mode | XSEL | ○ Can be used |
| | RC | ○ Can be used |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RSOF | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Turn OFF the servo of each RC-axis specified by an RAXS command.

> Important note: Before executing this command, set an axis pattern using an RAXS command. If not, a "(43B) RC-axis pattern not-set error" occurs.

[Example]

| | | | |
|---|---|---|---|
| RAXS | 0 | 1100 | Set an axis pattern consisting of axes 2 and 3. |
| RSOF | | | Turn OFF the servos of the specified axes. |

## ● RHOM (Return RC-axis to home)

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | ○ Can be used |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RHOM | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Return each RC-axis specified by an RAXS command to its home.
The servo of the axis to be returned home turns ON automatically.

[Example]

| | | | |
|---|---|---|---|
| RAXS | 0 | 1100 | Set an axis pattern consisting of axes 2 and 3. |
| RHOM | | | Return the specified axes to their home. |

## ● RMVP (Move RC-axis by position specification)

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | ○ Can be used |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RMVP | Position number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Move each RC-axis specified by an RAXS command to the position number in operand 1.
The output turns OFF when the axis movement is started, and turns ON when completed.

⚠ Caution: The specific operation varies between the XSEL position-data use mode and RC position-data use mode.

1) XSEL position-data use mode
   → Move via PTP to the position corresponding to the position number in operand 1.
2) RC position-data use mode
   → The specific operation varies depending on the position data in the RC controller.

| No. | Position data item in RC | | |
|---|---|---|---|
| | Push-motion | Incremental | Description of operation |
| 1 | 0 | 0 | Move via PTP to the position corresponding to the position number in operand 1. |
| 2 | 0 | 1 | Move incrementally (via PTP) by the travel corresponding to the position number in operand 1. |
| 3 | Other than 0 | 0 | Move to the position corresponding to the position number in operand 1 and then perform push-motion operation.<br>The output turns OFF if any one axis has been pushed missed the load. |
| 4 | Other than 0 | 1 | Move to the position corresponding to the position number in operand 1 and then perform push-motion operation.<br>The output turns OFF if any one axis has been pushed and missed the load. |

Important note: Before executing this command, set an axis pattern using an RAXS command. If not, a “(43B) RC-axis pattern not-set error” occurs.

| | | | | |
|---|---|---|---|---|
| [Example 1] | RAXS | 0 | 11 | Set an axis pattern consisting of axes 0 and 1. |
| | RMVP | 10 | | Move the specified axes to the positions corresponding to position No. 10. |
| [Example 2] | RAXS | 0 | 11 | Set an axis pattern consisting of axes 0 and 1. |
| | LET | 1 | 10 | Assign 10 to variable 1. |
| | RMVP | *1 | | Move the specified axes to the positions corresponding to position No. 10 according to the content of variable 1, or 10. |

## ● RMPI (Move RC-axis incrementally by position specification)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RMPI | Position number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Move each RC-axis specified by an RAXS command by the travel corresponding to the position data number in operand 1.
The output turns OFF when the axis movement is started, and turns ON when completed.

> Important note: Before executing this command, set an axis pattern using an RAXS command. If not, a "(43B) RC-axis pattern not-set error" occurs.

[Example 1]

| | | | |
|---|---|---|---|
| RAXS | 0 | 11 | Set an axis pattern consisting of axes 0 and 1. |
| RHOM | 10 | | Move by the travel corresponding to position No. 10. |

[Example 2]

| | | | |
|---|---|---|---|
| RAXS | 0 | 11 | Set an axis pattern consisting of axes 0 and 1. |
| LET | 1 | 10 | Assign 10 to variable 1. |
| RMPI | *1 | | Move the specified axes by the travels corresponding to position No. 10 according to the content of variable 1, or 10. |

(Note 1) 405 RC Gateway Communication Type Error will occur in Fieldbus Type.

## ● RMVD (Move RC-axis absolutely by direct numerical specification of position)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RMVD | RC-axis number | Variable number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Perform absolute position movement using the values in variable No. n to variable No. n+3.
The output turns OFF when the axis movement is started, and turns ON when completed.

| Variable number | Description of setting |
|---|---|
| n | Target position |
| n+1 | Speed [mm/s] |
| n+2 | Acceleration/deceleration [G] |
| n+3 | In-position width [mm] |

[Operand 1 setting type]

| Operand 1 | Specification of execution axis |
|---|---|
| 0 to 15 | The axis corresponding to the specified RC-axis number performs absolute position movement. |
| -1 | Each RC-axis specified by an RAXS command performs absolute position movement. |

* Specifying -1 is valid with XSEL_P/Q/PCT/QCT Ver.0.87 or later and XSEL_PX/QX Ver.0.42 or later.

[Example]

| | | | |
|---|---|---|---|
| LET | 300 | 100 | Set the target position to 100mm. |
| LET | 301 | 200 | Set the speed to 200mm/s. |
| LET | 302 | 0.3 | Set the acceleration/deceleration to 0.3G. |
| LET | 303 | 0.1 | Set the in-position width to 0.1mm. |
| RMVD | 1 | 300 | Move RC-axis 1 absolutely to the specified position. |

## ● RMDI (Move RC-axis incrementally by direct numerical specification of position)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RMDI | RC-axis number | Variable number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Perform incremental position movement using the values in variable No. n to variable No. n+3.
The output turns OFF when the axis movement is started, and turns ON when completed.

| Variable number | Description of setting |
|---|---|
| n | Travel |
| n+1 | Speed [mm/s] |
| n+2 | Acceleration/deceleration [G] |
| n+3 | In-position width [mm] |

[Operand 1 setting type]

| Operand 1 | Specification of execution axis |
|---|---|
| 0 to 15 | The axis corresponding to the specified RC-axis number performs incremental position movement. |
| -1 | Each RC-axis specified by an RAXS command performs incremental position movement. |

* Specifying -1 is valid with XSEL_P/Q/PCT/QCT Ver.0.87 or later and XSEL_PX/QX Ver.0.42 or later.

[Example]

| | | | |
|---|---|---|---|
| LET | 300 | 50 | Set the travel to 50mm. |
| LET | 301 | 200 | Set the speed to 200mm/s. |
| LET | 302 | 0.3 | Set the acceleration/deceleration to 0.3G. |
| LET | 303 | 0.1 | Set the in-position band to 0.1mm. |
| RMDI | 1 | 300 | Move RC-axis 1 incrementally to the specified position. |

(Note 1) 405 RC Gateway Communication Type Error will occur in Fieldbus Type.

## ● RPUS (Move RC-axis via push motion)

*1 439 RC Position Data Use Method Error when a command was executed

| RC position-data use mode | | |
|---|---|---|
| | XSEL | ○ Can be used |
| | RC | × Cannot be used*1 |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RPUS | RC-axis number | Position number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] The axis moves to the target position corresponding to the position number in operand 2, and then push the load over the in-position width specified by the position data.
The push force is set by the current-limiting value for push-motion operation among the position data.
The output turns ON when a push action is confirmed, and turns OFF if a miss is detected.

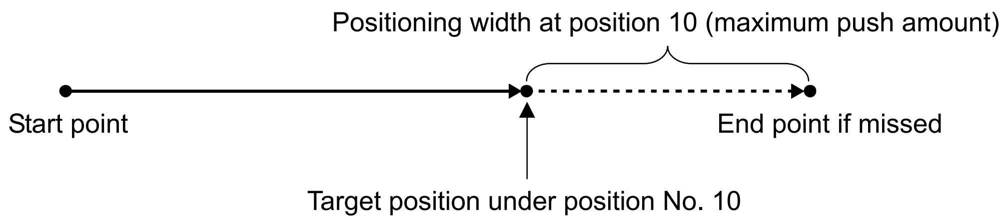
[Operand 1 setting type]

| Operand 1 | Specification of execution axis | Output specification |
|---|---|---|
| 0 to 15 | The axis corresponding to the specified RC-axis number moves via push motion. | The output turns ON when pushing of the command axis is confirmed. |
| -1 | Each RC-axis specified by an RAXS command moves via push motion. | The output turns ON when pushing of all command axes is confirmed. |
| -2 | Each RC-axis specified by an RAXS command moves via push motion. | The output turns ON when pushing of any one of all command axes is confirmed. |

* Specifying -1 is valid with XSEL_P/Q/PCT/QCT Ver.0.87 or later and XSEL_PX/QX Ver.0.42 or later.

⚠ Caution:
- If a positive sign is appended to positioning width data, the load is pushed in the direction of increasing coordinates from the start point of the RPUS command toward the target position.
- If a negative sign is appended, the load is pushed in the direction of decreasing coordinates. (The operation is different from when a PUSH command is used.)

[Example] PRUS 3 10 Move RC-axis 3 to the position corresponding to position No. 10 and cause it to push the load.



## ● RSTP (Cancel RC-axis movement)

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | ○ Can be used |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RSTP | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Decelerate each RC-axis specified by an RAXS command to a stop.
This command is valid with respect to all RC-axis control commands other than RSOF.

> Notice : Before executing this command, set an axis pattern using an RAXS command.
> If not, a “(43B) RC-axis pattern not-set error” occurs.

[Example]

| | | | |
|---|---|---|---|
| RAXS | 0 | 11 | Set an axis pattern consisting of axes 0 and 1. |
| RSTP | | | Decelerate the specified axes to a stop. |

## ● RCST (Read RC-axis status)

| RC position-data use mode | XSEL | ○ Can be used |
|---|---|---|
| | RC | ○ Can be used |

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | RCST | Variable number | RC-axis number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | ○ | ○ | × | × | × | × |

[Function] Read the RC-axis status into the variable number in operand 1.
Read the completed position number into variable n+1. (Refer to "Note 2".)

Notice 1: The specific status varies between the XSEL position-data use mode and RC position-data use mode.

| Variable number | Acquired data |
|---|---|
| n | RC-axis status |
| n+1 | Completed position number |

RC-axis status bit structure

| Bit | XSEL position-data use mode | | RC position-data use mode | |
|---|---|---|---|---|
| | Name | Explanation | Name | Explanation |
| 27-31 | – | Reserved | – | Reserved |
| 26 | ALMX | RC-axis alarm<br>(Error detected by the XSEL)<br>* When ALM turns ON, ALMX also turns ON. However, ALM may not turn ON even if ALMS turns ON depending on the error. | ALMX | RC-axis alarm<br>(Error detected by the XSEL)<br>* When ALM turns ON, ALMX also turns ON. However, ALM may not turn ON even if ALMS turns ON depending on the error. |
| 25 | USE | RC-axis in use | USE | RC-axis in use |
| 24 | LNK | RC-axis linked | LNK | RC-axis linked |
| 16-23 | – | Reserved | – | Reserved |
| 15 | RMDS | Operation Mode | RMDS | Operation Mode |
| 14 | ALML | Light Error Alarm | ALML | Light Error Alarm |
| 13 | ZON2 | Zone 2 | ZON2 | Zone 2 |
| 12 | ZON1 | Zone 1 | ZON1 | Zone 1 |
| 11 | – | Reserved | PZON | Position zone |
| 10 | – | Reserved | MODS | Teaching mode status |
| 9 | SFTY | Safety speed enabled | SFTY | Safety speed enabled |
| 8 | BALM | Battery voltage low | BALM | Battery voltage low |
| 7 | EMG | Emergency stop | EMG | Emergency stop |
| 6 | PSFL | Load not pushed | PSFL | Load not pushed |
| 5 | CRDY | Controller ready | CRDY | Controller ready |
| 4 | SON | Servo ON | SON | Servo ON |
| 3 | MOVE | Moving | MOVE | Moving |
| 2 | HEND | Home return complete | HEND | Home return complete |
| 1 | PEND | Positioning complete | PEND | Positioning complete |
| 0 | ALM | Operation-disabling alarm<br>(Error detected by the RC-axis) | ALM | Operation-disabling alarm<br>(Error detected by the RC-axis) |

Notice 2: Completed position numbers are set only in the RC position-data use mode.
In the XSEL position-data use mode, this bit is always set to 0.

[Example] RCST 200 10 Acquire the status of RC-axis 10 into variable 200.

## [27] Extension Motion Control Function

● XCRP (Clear input counter record for extension motion control)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XCRP | Pulse input channel number | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] This clears the specified pulse input channel counter to 0.

⚠ Caution
The counter clear cannot be performed when the pulse I/O board axis is in synchronizing operation with the specified channel used as the master axis.

[Example 1] XCRP 0 It clears the counter for the pulse input channel No. 0.

● XGTP (Acquire the current value for extension motion control pulse input counter)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XGTP | Pulse input channel number | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the current value for the pulse input channel counter specified in Operand 1 into Variable 99.

[Example 1] XGTP 0 It acquires the pulse input channel No. 0 counter in Variable 99.

> ⚠ Caution
> The pulse I/O board input channel is a signed 32-bit counter.

● XPGT (Read extension motion control axis position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPGT | Axis number | Position number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the position data location [mm] specified in Operand 2 on the Extension motion control axis specified in Operand 1 into Variable 199 (minimum effective digit number = 3).

[Example 1]

| | | | |
|---|---|---|---|
| XPGT | 2 | 3 | It reads the position set in Position No. 3 of the 2nd axis into Variable 199. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| XPGT | *1 | *2 | It reads the position set in Position No. 3 (content of Variable 2) of the 2nd axis (content of Variable 1) into Variable 199. |

⚠ Caution
If ineffective position data is specified in Operand 2, Variable 199 becomes non-operated and the output section is turned OFF.

● XPPT (Write extension motion control axis position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPPT | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the position data location [mm] specified in Operand 2 on the Extension motion control axis specified in Operand 1 into Variable 199 (minimum effective digit number = 3).

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 150 | Assign 150 to variable 199. |
| XPPT | 2 | 3 | It writes Content 150 in Variable 199 in Position No. 3 of the 2nd axis. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 150 | Assign 150 to variable 199. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| XPPT | *1 | *2 | It writes Content 150 in Variable 199 in the position set in Position No. 3 (content of Variable 2) of the 2nd axis (content of Variable 1). |

● XPCR (Erase extension motion control axis position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPCR | Axis number | Variable number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It erases the Extension motion control axis position data specified in Operand 1 by using the two variables in a row from Variable No. n specified in Operand 2. The erased data becomes a blank.

| Variable No. | Description of setting |
|---|---|
| n | Start position number |
| n+1 | End position number |

[Example 1]

| | | | |
|---|---|---|---|
| LET | 200 | 10 | Assign 10 to variable 200. |
| LET | 201 | 20 | Assign 20 to variable 201. |
| XPCR | 1 | 200 | It erases Positions No. 10 to 20 in the 1st axis. |

● XPCP (Copy extension motion control axis position data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPCP | Axis number | Variable number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It copies the Extension motion control axis position data specified in Operand 1 by using the two variables in a row from Variable No. n specified in Operand 2.

| Variable No. | Description of setting |
|---|---|
| n | Position number to copy data to |
| n+1 | Position number to copy data from |

[Example 1]

| | | | |
|---|---|---|---|
| LET | 200 | 20 | Assign 20 to variable 200. |
| LET | 201 | 10 | Assign 10 to variable 201. |
| XPCP | 1 | 200 | It copies Position No. 10 data in the 1st axis to Position No. 20. |

● XPRD (Read extension motion control axis current command position)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPRD | Positio number | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the current order position of the Extension motion control axis specified by XAXS Command into the position number specified in Operand 1.

⚠ Caution
- Make sure to set the axis pattern by XAXS or XA16 Command before executing this command. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD).
- he position acquired in this command is the current order position from the pulse I/O board. Make sure to perform a home-return operation before executing this command. (XSEL-P/Q/PCT/QCT/R/S)

[Example 1]

| | | | |
|---|---|---|---|
| XAXS | 0 | 111 | Set an pattern that uses axes 0,1 and 2. |
| XPRD | 100 | | It reads the current order position of 0 to 2 axes into Position No. 100. |

[Example 2]

| | | | |
|---|---|---|---|
| XAXS | 0 | 111 | Set an pattern that uses axes 0,1 and 2. |
| LET | 1 | 100 | Assign 100 to variable 1. |
| XPRD | *1 | | It reads the current order position of 0 to 2 axes into the position in Content No. 100 of Variable 1. |

● XPRQ (Read extension motion control axis current command position (single-axis direct))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPRQ | Axis number | Variable number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the current order position of the Extension motion control axis specified in Operand 1 into variable specified in Operand 2 (minimum effective digit number = 3). It enables a faster acquirement of the current order position than using XPRD Command.

> ⚠ Caution
> The position acquired in this command is the current order position from the pulse I/O board. Make sure to perform a home-return operation before executing this command.
> (XSEL-P/Q/PCT/QCT/R/S)

[Example] XPRQ 2 100 It reads the current order position of the 2nd axis into Variable No. 100.

● XPVL (Write extension motion control axis speed data)

| Extension condition (LD,A,O,AB,OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPVL | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It writes the value in Variable 199 to the position data speed [mm/s] specified in Operand 2 on the Extension motion control axis specified in Operand 1 (minimum effective digit number = 2).

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 100 | Assign 100 to variable 199. |
| XPVL | 2 | 3 | It writes the speed 100mm/s in Variable 199 to Position No. 3 on the 2nd axis. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 100 | Assign 199 to variable 100. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| XPVL | *1 | *2 | It writes the speed 100mm/s in Variable 199 to Position No. 3 (content in Variable 2) on the 2nd axis (content in Variable 1). |

● XPAC (Write extension motion control axis acceleration data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPAC | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It writes the value in Variable 199 to the position data acceleration [G] specified in Operand 2 on the Extension motion control axis specified in Operand 1 (minimum effective digit number = 2).

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 0.3 | Assign 0.3 to variable 199. |
| XPAC | 2 | 3 | It writes the acceleration 0.3G in Variable 199 to Position No. 3 on the 2nd axis. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 0.3 | Assign 0.3 to variable 199. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| XPAC | *1 | *2 | It writes the acceleration 0.3G in Variable 199 to Position No. 3 (content in Variable 2) on the 2nd axis (content in Variable 1). |

● XPDC (Write extension motion control axis deceleration data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPDC | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |



[Function] It writes the value in Variable 199 to the position data deceleration [G] specified in Operand 2 on the Extension motion control axis specified in Operand 1 (minimum effective digit number = 2).

[Example 1]

| | | | |
|---|---|---|---|
| LET | 199 | 0.3 | Assign 0.3 to variable 199. |
| XPDC | 2 | 3 | It writes the deceleration 0.3G in Variable 199 to Position No. 3 on the 2nd axis. |

[Example 2]

| | | | |
|---|---|---|---|
| LET | 199 | 0.3 | Assign 0.3 to variable 199. |
| LET | 1 | 2 | Assign 2 to variable 1. |
| LET | 2 | 3 | Assign 3 to variable 2. |
| XPDC | *1 | *2 | It writes the deceleration 0.3G in Variable 199 to Position No. 3 (content in Variable 2) on the 2nd axis (content in Variable 1). |

● XPIP (Write extension motion control axis positioning complete width data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPIP | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It writes the value in Variable 199 to the position data positioning complete width [mm] specified in Operand 2 on the Extension motion control axis specified in Operand 1 (minimum effective digit number = 3).

| | | | | |
|---|---|---|---|---|
| [Example 1] | LET | 199 | 0.2 | Assign 0.2 to variable 199. |
| | XPIP | 2 | 3 | It writes the positioning complete width 0.2mm in Variable 199 to Position No. 3 on the 2nd axis. |

| | | | | |
|---|---|---|---|---|
| [Example 2] | LET | 199 | 0.2 | Assign 0.2 to variable 199. |
| | LET | 1 | 2 | Assign 2 to variable 1. |
| | LET | 2 | 3 | Assign 3 to variable 2. |
| | XPIP | *1 | *2 | It writes the positioning complete width 0.2mm in Variable 199 to Position No. 3 (content in Variable 2) on the 2nd axis (content in Variable 1). |

● XGVL (Read extension motion control axis speed data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XGVL | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the position data speed [mm/s] specified in Operand 2 on the Extension motion control axis specified in Operand 1 into Variable 199 (minimum effective digit number = 2).

[Example] XGVL 2 3 It reads the speed in Position No. 3 on the 2nd axis into Variable 199.

● XGAC (Read extension motion control axis acceleration data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XGAC | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the position data acceleration [G] specified in Operand 2 on the Extension motion control axis specified in Operand 1 into Variable 199 (minimum effective digit number = 2).

[Example] XGAC 2 3 It reads the acceleration in Position No. 3 on the 2nd axis into Variable 199.

● XGDC (Read extension motion control axis deceleration data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XGDC | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the position data deceleration [G] specified in Operand 2 on the Extension motion control axis specified in Operand 1 into Variable 199 (minimum effective digit number = 2).

[Example] XGDC 2 3 It reads the deceleration in Position No. 3 on the 2nd axis into Variable 199.

● XGIP (Read extension motion control axis positioning complete width data)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XGIP | Axis number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the position data positioning complete width [mm] specified in Operand 2 on the Extension motion control axis specified in Operand 1 into Variable 199 (minimum effective digit number = 3).

[Example] XGIP 2 3 It reads the positioning complete width in Position No. 3 on the 2nd axis into Variable 199.

● XAXS (Extension motion control axis pattern setting (0 to 15 axis))

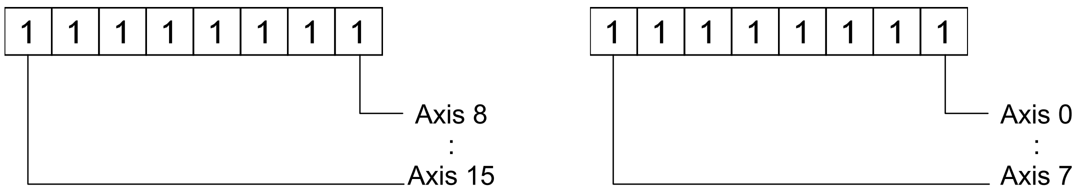
| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XAXS | Axis pattern, upper | Axis pattern, lower | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |



[Function] It performs a setting for the axis patterns of axes 8 to 15 on the Extension motion control axis in Operand 1 and the axis patterns of axes 0 to 7 in Operand 2.

Operand 1 Operand 2




After the program execution is started, make sure to set the axis patterns using this command before the following commands are executed.
If the axis pattern setting command XAXS is not conducted, No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), Error No. 4B3 “Extension motion control board axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.

XPRD : Read extension motion control board axis current command position
XSON : Servo ON of extension motion control board axis
XSOF : Servo OFF of extension motion control board axis
XHOM : Home return of extension motion control board axis
XMVP : Move extension motion control board axis to indicated position
XMPI : Perform extension motion control board axis position relative movement
XMVL : Move extension motion control board axis for position indicated interpolation
XMLI : Move extension motion control board axis for position relative interpolation
XSTP : Cancel operation of extension motion control board axis

| | | | | |
|---|---|---|---|---|
| [Example] | XAXS | 1010101 | 10101010 | Set an axis pattern consisting of axes 1, 3, 5, 7, 8, 10, 12 and 14. |
| | XSON | | | It turns the servo ON for axes 1, 3, 5, 7, 8, 10, 12 and 14. |
| | XMVP | 20 | | It moves the axes 1, 3, 5, 7, 8, 10, 12 and 14 to Position 20. |

● XA16 (Extension motion control axis pattern setting (16 to 31 axis))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XA16 | Axis pattern, upper | Axis pattern, lower | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | ○ | × | × | × | ○ | × | × | × |

[Function] It performs a setting for the axis patterns of axes 24 to 31 on the Extension motion control axis in Operand 1 and the axis patterns of axes 16 to 23 in Operand 2.

Operand 1

1 1 1 1 1 1 1 1

Axis 24
:
Axis 31

Operand 2

1 1 1 1 1 1 1 1

Axis 16
:
Axis 23

After the program execution is started, make sure to set the axis patterns using this command before the following commands are executed.
If the axis pattern setting command XAXS, XA16 is not conducted, Error No. 445 “Extension motion control axis pattern not set error” would occur.

XPRD : Read extension motion control axis current command position
XSON : Servo ON of extension motion control axis
XSOF : Servo OFF of extension motion control axis
XHOM : Home return of extension motion control axis
XMVP : Move extension motion control axis to indicated position
XMPI : Perform extension motion control axis position relative movement
XMVL : Move extension motion control axis for position indicated interpolation
XMLI : Move extension motion control axis for position relative interpolation
XSTP : Cancel operation of extension motion control axis

[Example]

| | | | |
|---|---|---|---|
| XA16 | 1010101 | 10101010 | Set an axis pattern consisting of axes 17, 19, 21, 23, 24, 26, 28 and 30. |
| XSON | | | It turns the servo ON for axes 17, 19, 21, 23, 24, 26, 28 and 30. |
| XMVP | 20 | | It moves the axes 17, 19, 21, 23, 24, 26, 28 and 30. |

● XSON (Extension motion control axis servo ON)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XSON | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |



[Function] It turns the servo ON for the Extension motion control axis specified by XAXS Command.

⚠ Caution
Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 "Extension motion control board axis pattern not set error" (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 "Extension motion control axis pattern not set error" (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.

| [Example] | | | | |
|---|---|---|---|---|
| | XAXS | 01 | 100 | Set an axis pattern that uses axes 2 and 3. |
| | XSON | | | Turn ON the servos of the specified axes. |

● XSOF (Extension motion control axis servo OFF)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XSOF | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It turns the servo OFF for the Extension motion control axis specified by XAXS Command.

> ⚠ Caution
> Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.

[Example]

| | | | |
|---|---|---|---|
| XAXS | 0 | 1100 | Set an axis pattern that uses axes 2 and 3. |
| XSOF | | | Turn OFF the servos of the specified axes. |

● XHOM (Extension motion control axis home return)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XHOM | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |



[Function] It turns the home return for the Extension motion control axis specified by XAXS Command.
The servo of the axis to be returned home turns ON automatically.

⚠ Caution

- Make sure to set the axis pattern by XAXS or XA16 Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.
- The servo would turn OFF if the operation is either paused or stopped to cancel during the home-return operation of the pulse I/O axis. When resuming the operation after a pause, confirm the servo is ON and then make sure to start with a home-return operation.
- When the axis to be object is a slave axis of the absolute encoder, execute this command and some controllers will conduct the absolute reset.
  When it is required to have a move to the home position (when home position confirmation is not necessary), make a move to the home position using XMVP Command instead of XHOM Command.

[Example]

| | | | |
|---|---|---|---|
| XAXS | 0 | 1100 | Set an axis pattern that uses axes 2 and 3. |
| XHOM | | | Home Return the servos of the specified axes. |

● XMVP (Move extension motion control axis to indicated position)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XMVP | Position number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It moves the Extension motion control axis specified by XAXS or XA16 Command by PTP operation to the position number specified in Operand 1.

⚠ Caution
- Make sure to set the axis pattern by XAXS or XA16 Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.
- In the case of XSEL-P/Q/PCT/QCT/R/S, is able to specify another axis that is connected to a different extension motion control board at the same time. However, since the system is controlled by each board, the operation cannot be synchronized.

[Example 1]

```
XAXS   0    11    Set an axis pattern that uses axes 0 and 1.
XMVP   10         Move the specified axes to the positions
                  corresponding to position No. 10.
```

[Example 2]

```
XAXS   0    11    Set an axis pattern that uses axes 0 and 1.
LET    1    10    Assign 10 to variable 1.
XMVP   *1         Move the specified axes to the positions
                  corresponding to position No. 10 according to the
                  content of variable 1, or 10.
```

● XMPI (Perform extension motion control axis position relative movement)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XMPI | Position number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It moves the Extension motion control axis specified by XAXS Command by PTP operation with the position number in Operand 1 taken as the amount of movement.

> ⚠ Caution
> - Make sure to set the axis pattern by XAXS or XA16 Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.
> - In the case of XSEL-P/Q/PCT/QCT/R/S, is able to specify another axis that is connected to a different extension motion control board at the same time. However, since the system is controlled by each board, the operation cannot be synchronized.

[Example 1]

| | | | |
|---|---|---|---|
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| XMPI | 10 | | Move by the travel corresponding to position No. 10. |

[Example 2]

| | | | |
|---|---|---|---|
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| LET | 1 | 10 | Assign 10 to variable 1. |
| XMPI | *1 | | Move the specified axes by the travels corresponding to position No. 10 according to the content of variable 1, or 10. |

● XMVL (Move extension motion control axis for position indicated interpolation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XMVL | Position number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It moves the Extension motion control axis specified by XAXS Command by direct interpolation movement to the position number specified in Operand 1.

(Note 1) Make sure to set the axis pattern by XAXS or XA16 Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control board axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.

(Note 2) If another axis connected to a different pulse I/O board is specified, Error No. C30 “Axis pattern error” would occur. (XSEL-P/Q/PCT/QCT/R/S)

(Note 3) It is necessary to specify the speed, acceleration and deceleration values by VEL, VLMX, ACC, and DCL Commands before executing this command. If not specified, an error would occur.

[Example 1]

| | | | |
|---|---|---|---|
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| XMVL | 10 | | Move the specified axes to the positions corresponding to position No. 10. |

[Example 2]

| | | | |
|---|---|---|---|
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| LET | 1 | 10 | Assign 10 to variable 1. |
| XMVL | *1 | | Move the specified axes to the positions corresponding to position No. 10 according to the content of variable 1, or 10. |

● XMLI (Move extension motion control axis for position relative interpolation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XMLI | Position number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It moves the Extension motion control axis specified by XAXS Command by direct interpolation movement with the position number in Operand 1 taken as the amount of movement.

(Note 1) Make sure to set the axis pattern by XAXS or XA16 Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.

(Note 2) If another axis connected to a different pulse I/O board is specified, Error No. C30 “Axis pattern error” would occur. (XSEL-P/Q/PCT/QCT/R/S)

(Note 3) It is necessary to specify the speed, acceleration and deceleration values by VEL, VLMX, ACC, and DCL Commands before executing this command. If not specified, an error would occur.

[Example 1]

| | | | |
|---|---|---|---|
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| XMLI | 10 | | Move by the travel corresponding to position No. 10. |

[Example 2]

| | | | |
|---|---|---|---|
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| LET | 1 | 10 | Assign 10 to variable 1. |
| XMLI | *1 | | Move the specified axes by the travels corresponding to position No. 10 according to the content of variable 1, or 10. |

● XMVD (Move extension motion control axis to directly indicated absolute position)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XMVD | Axis number | Variable number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It moves the Extension motion control axis specified in Operand 1 by absolute position movement to the values specified in the five variables in a row from Variable No. n in Operand 2.

| Variable No. | Description of setting | Effective Digits |
|---|---|---|
| n | Target position [mm] | 3 digits minimum |
| n+1 | Speed [mm/s] | 2 digits minimum |
| n+2 | Acceleration [G] | 2 digits minimum |
| n+3 | Deceleration [G] | 2 digits minimum |
| n+4 | Positioning complete width [mm] | 3 digits minimum |

(Note 1) VLMX Command is invalid to this command.

[Example]

| | | | |
|---|---|---|---|
| LET | 300 | 100 | Set the target position to 100mm. |
| LET | 301 | 200 | Set the speed to 200mm/s. |
| LET | 302 | 0.3 | Set the acceleration/deceleration to 0.3G. |
| LET | 303 | 0.3 | Set the deceleration/deceleration to 0.3G. |
| LET | 304 | 0.1 | Set the in-position width to 0.1mm. |
| XMVD | 1 | 300 | Move RC-axis 1 absolutely to the specified position. |

● XMDI (Move extension motion control axis to directly indicated relative position)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XMDI | Axis number | Variable number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |



[Function] It moves the Extension motion control axis specified in Operand 1 by relative position movement to the values specified in the five variables in a row from Variable No. n in Operand 2.

| Variable No. | Description of setting | Effective Digits |
|---|---|---|
| n | Travel [mm] | 3 digits minimum |
| n+1 | Speed [mm/s] | 2 digits minimum |
| n+2 | Acceleration [G] | 2 digits minimum |
| n+3 | Deceleration [G] | 2 digits minimum |
| n+4 | Positioning complete width [mm] | 3 digits minimum |

(Note 1) VLMX Command is invalid to this command.

[Example]

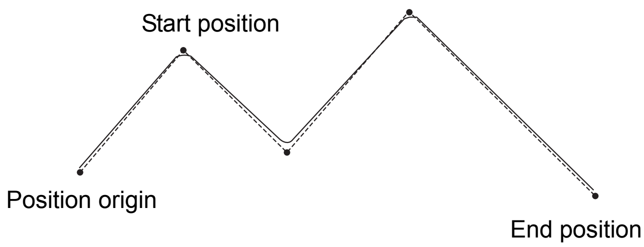
```
LET    300   50     Set the travel to 50mm.
LET    301   200    Set the speed to 200mm/s.
LET    302   0.3    Set the acceleration/deceleration to 0.3G.
LET    303   0.3    Set the deceleration/deceleration to 0.3G.
LET    304   0.1    Set the in-position width to 0.1mm.
XMDI   1     300    Move RC-axis 1 absolutely to the specified position.
```

●XPTH (Extension motion control axis path operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | XPTH | Start position number | End position number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | ○ | × | × | × | ○ | × | × | × |

[Function] Move continuously from the position specified in operand 1 to the position specified in operand 2 (CP Movement). The output type in the output field can be set using an actuator-declaration command POTP. If invalid data is set for any position number between the start and end position numbers, that position number will be skipped during continuous movement.




(Note 1) Set axis patterns using the XAXS or XA16 command before executing this command. If the command has not been executed, error No. 4B3 “No extension motion control board axe pattern setting error” occurs.

(Note 2) This command requires indication of the velocity, acceleration and deceleration in VAL, VLMX, ACC, DCL commands before execution. If those commands have not been executed, an error occurs.

(Note 3) Multi-dimensional movement can be performed using this command. In this case, input in operand 1 the point number of the next target, instead of the predicted current position upon execution of the applicable command.
(Inputting a position number corresponding to the predicted current position will trigger movement to the same position during continuous movement, thereby causing the speed to drop.)

(Note 4) Input this command straight after program steps in a row, and the actuator is able to make a smooth operation without any stop between steps when a program is executed. However, in case there is any step with an input condition being indicated, the actuator will pause at the step before that.

(Note 5) When POTP = 0, the operation timing of the output part during the continuous operation should be when getting close to the final movement position of each command, and when POTP = 1, it should be when getting close to each position. As for the final movement position of the continuous movement final command, it turns ON when the operation is completed. However, in case there is the positioning complete band established at the final movement position of the continuous movement final command, it will not turn ON.

(Note 6) Positioning complete band is valid only for the final movement position. Also, it is valid only for the final movement position of the continuous movement final command while in continuous operation.

(Note 7) It is possible to move through discontinuous positions or move continuously by passing the same position. As shown in the example, specify the number corresponding to the discontinuous position for both the start position number and end position number. In the example, this position is No. 6.

[Example] The actuator moves continuously in the sequence of position No. 1 → 2 → 3 → 4 → 6 → 9 → 10.

| | | | |
|---|---|---|---|
| XPTH | 1 | 4 | |
| XPTH | 6 | 6 | discontinuous position |
| XPTH | 9 | 10 | |

[Example 1] XPTH 100 120 Move continuously from position No. 100 to 120.

● XJ□□ (Perform extension motion control axis jog operation)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XJ□□ | Input, output, flag number | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |



[Function] It moves the Extension motion control shaft specified by XAXS Command in back and forth while the flag on the input port or output port specified in Operand 1 is turning ON and OFF.

XJFN ·········· It moves forward when the specified port is ON.
XJFF ·········· It moves forward when the specified port is OFF.
XJBN ·········· It moves backward when the specified port is ON.
XJBF ·········· It moves backward when the specified port is ON.

(Note 1) Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.

(Note 2) It is effective also to the axis that the home-return operation is incomplete. However, the upper limit for the speed is that set in pulse I/O board command parameter No. 4 “Maximum JOG speed at home return incomplete”. In such a condition, exercise precaution not to crash into the work or stroke end since the coordinate values become meaningless.

(Note 3) In the case of XSEL-P/Q, this command is valid on Main CPU Application Section Ver.1.02 or later. (It is available from the first for XSEL-R/S, XSEL-RA/SA/RAX/SAX/RAXD/SAXD.) And a PC software Ver.7.6.5.0 or later which is applicable for this command is also required.

[Example 1]

| | | | |
|---|---|---|---|
| VEL | 100 | | Set the speed to 100mm/s. |
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| XJBF | 10 | | Move axis 5 backward while input 10 is OFF. |

[Example 2]

| | | | |
|---|---|---|---|
| VEL | 100 | | Set the speed to 100mm/s. |
| LET | 5 | 20 | Assign 20 to variable 5. |
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| XJFN | *5 | | Move axis 0 and 1 forward while the content of variable 5 (input 20), is ON. |

● XPED (Waiting for extension motion control axis to finish positioning operation of axis used by self-program)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XPED | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It waits for the positioning operation of the Extension motion control axis used in the program its own. By this command, it is possible to wait for the completion of the positioning operation (XMVP, XMPI, XMVL, XMLI, XMVD and XMDI) when the positioning complete width is valid. The output becomes ON when the operation is completed in normal condition. The command would not react after an execution of any operation command other than positioning operation. (Output section is OFF.)
For an operation that the positioning complete band is valid, recovery from the operation command is performed once the actuator reaches in front of the positioning complete band at the current position (or current position command when the pulse input and output boards are mounted (XSEL-P/Q/PCT/QCT/R/S)). (Output section is OFF.) It is possible to confirm the positioning is complete by executing this command after the command recovery. Also, the driven axis is usually occupied by the executed program after the operation command recovery. By executing this command, the axis gets released, thus the axis becomes available for a use by other programs.

[Example]

| | | | |
|---|---|---|---|
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| XMVP | 10 | | Move the specified axes to the positions corresponding to position No. 10. For an operation that the positioning complete band is valid, recovery from the operation command is performed once the actuator reaches in front of the positioning complete band at the current position (or current position command when the pulse input and output boards are mounted). |
| BTON | 308 | | It turns ON Output Port No. 308. |
| XPED | | | It waits for the positioning operation axis No. 0 and 1 of the program itself to complete. |

● XSTP (Cancel operation of extension motion control axis)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XSTP | Prohibited | Prohibited | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It decelerates and stops the Extension motion control axis specified by XAXS Command. It is valid for the Extension motion control axis command other than XSOF command.

(Note 1) Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control board axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.

[Example]

| | | | |
|---|---|---|---|
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| XSTP | | | Decelerate the specified axes to a stop |

● XWIP (Waiting for extension motion control axis positioning complete signal to be turned ON)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XWIP | Prohibited | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It waits till the positioning complete signal of the Extension motion control shaft specified by XAXS Command turns ON. This enables to wait for the completion of the positioning operation of the slave shaft (= slave shaft positioning complete signal ON) while in synchronizing process by executing this command to the synchronizing slave shaft after the synchronizing master shaft operation command is complete (*). The positioning complete signal for the slave shaft turns ON when position deviation ≤ positioning complete width. The status would not become waiting unless the pulse order is output from the master shaft side to the slave side.

* The pulse order of the slave shaft is completed by the completion of the master shaft operation command.

(Note 1) Make sure to set the axis pattern by XAXS Command before this command is executed. If the setting is not established, Error No. 445 “Extension motion control board axis pattern not set error” (XSEL-P/Q/PCT/QCT/R/S), No. 4B3 “Extension motion control axis pattern not set error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.

(Note 2) If the positioning complete signal does not turn ON even after the time set in the Extension motion control output channel parameter No. 33 “Positioning complete confirmation time” of the specified axis is passed, Error No. 454 “Extension motion control board axis positioning complete timeout error” (XSEL-P/Q/PCT/QCT/R/S), No. 4C3 “Extension motion control axis positioning complete timeout error” (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) would occur.
In the case of pulse input board (XSEL-P/Q/R/S), please check the positioning complete signal cable is broken.

(Note 3) In the case of XSEL-P/Q, this command is valid on Main CPU Application Section Ver.1.02 or later. (It is available from the first for XSEL-R/S, XSEL-RA/SA/RAX/SAX/RAXD/SAXD.) And a PC software Ver.7.6.5.0 or later which is applicable for this command is also required.

[Example]

| | | | |
|---|---|---|---|
| XCAS | 0 | 10 | It starts to synchronize the electronic cam on axis 0. (*1) |
| XCAS | 1 | 20 | It starts to synchronize the electronic cam on axis 1. (*2) |
| MOVP | 5 | | It moves the 1st axis of the main CPU control axes to Position No. 5. |
| MOVP | 6 | | It moves the 1st axis of the main CPU control axes to Position No. 6. |
| XAXS | 0 | 11 | Set an axis pattern that uses axes 0 and 1. |
| XWIP | | | It waits till the positioning complete signal of axes 0 and 1 to turn ON. (Axes 0 and 1 continue the synchronizing process.) |
| MOVP | 7 | | It moves the 1st axis of the main CPU control axe to Position No. 7. |

* It is assumed the setting that the 1st axis of the main CPU control axes is set to Variables No. 10 to 19 as the electronic cam synchronizing process is established.

* It is assumed the setting that the 1st axis of the main CPU control axes is set to Variables No. 20 to 29 as the electronic cam synchronizing process is established.

● XCAS ((Start synchronizing extension motion control axis electronic cam (indicating main axis)) extension motion control) (1/2)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XCAS | Slave shaft number | Variable number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | ○ | × | × | × | × | × | × | × |



[Function] It starts the synchronizing process with the axis specified in Operand 1 as the slave shaft extension motion control axis the electronic cam table. The synchronizing electronic cam operation settings such as the master shaft on the electronic cam table are to be specified in ten variables in a row from Variable No. n in Operand 2. The output section turns ON when the synchronizing process is started.

■Operand 2 : Synchronizing Electronic Cam Operation Settings

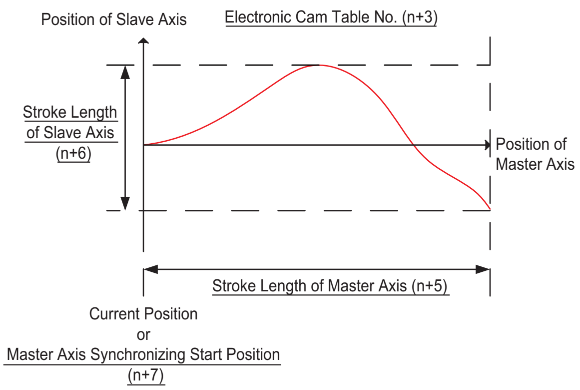
| Variable No. | Data name | Description | |
|---|---|---|---|
| n | Synchronizing type | See below | |
| n+1 | Master shaft type | 0 : Standard motion control axis<br>1 : Extension motion control axis | |
| n+2 | Master shaft number | Master shaft type (n+1) | Setting Content |
| | | 0 | 1 to 8 axis (Standard motion control axis) |
| | | 1 | 0 to 31 axis (Extension motion control axis) |
| n+3 | Electronic cam table number | 0 to maximum cam table number | |
| n+4 | Stroke type | 0 : Master shaft stroke length indication<br>1 : Master shaft stroke end position indication | |

<table>
<tr><th>Variable No.</th><th>Data name</th><th colspan="3">Description</th></tr>
<tr><td rowspan="5">n+5</td><td rowspan="5">Master shaft stroke length / stroke end position (Storage position number)</td><td>Stroke type (n+4)</td><td>Master shaft type (n+1)</td><td>Setting Content</td></tr>
<tr><td rowspan="2">0</td><td>0</td><td>Master shaft stroke length storage position number<br>* Indicate the standard motion control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td>1</td><td>Master shaft stroke length storage position number<br>* Indicate the extension motion control axis master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>Master shaft stroke end position storage position number<br>* Indicate the standard motion control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td>1</td><td>Master shaft stroke end position storage position number<br>* Indicate the extension motion control master shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td>n+6</td><td>Slave stroke length storage position number</td><td colspan="3">Indicate the pulse I/O board control slave shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td rowspan="4">n+7</td><td rowspan="4">Master shaft synchronizing start position (Storage position number)<br>* Effective only when “Master shaft reaches specified synchronizing start position” is selected for synchronizing type</td><td colspan="2">Master shaft type (n+1)</td><td>Setting Content</td></tr>
<tr><td colspan="2">0</td><td>Master shaft synchronizing start position storage position number<br>* Indicate the standard motion control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td colspan="2">1</td><td>Master shaft synchronizing start position storage position number<br>* Indicate the extension motion control master shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td colspan="2">2</td><td>Master shaft synchronizing start position (pulse unit)</td></tr>
<tr><td>n+8</td><td>Reserved</td><td colspan="3">Make sure to set 0</td></tr>
<tr><td>n+9</td><td>Reserved</td><td colspan="3">Make sure to set 0</td></tr>
</table>

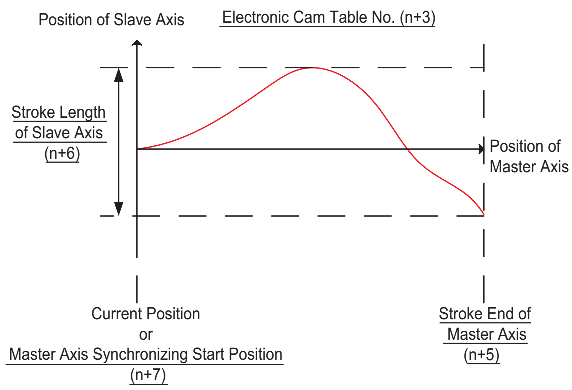
■Synchronizing Type (Variable No.n)

| Set value | Description | |
|---|---|---|
| | Synchronizing start type | Synchronizing process repeat type |
| 0 | Immediately | Operate for 1 cycle |
| 1 | Immediately | Repeated operation |
| 2 | Master shaft reaches specified synchronizing start position | Operate for 1 cycle |
| 3 | Master shaft reaches specified synchronizing start position | Repeated operation |

When Stroke Type (n+4): 0




When Stroke Type (n+4): 1




Synchronous movement continues until:

1) The XSYE command is executed (synchronous movement is stopped).
2) The operation of slave axes is stopped (XSTP/CANC commands).
3) The synchronous movement repetition type is single cycle only and the master axis reached the stroke end.
4) The slave axis movement program that executed the XCAS command is terminated.

- The master axis stroke length/stroke end position and master axis synchronization start position shall be set to position data of the master axis if the master axis is a position data. The slave axis stroke length is set in slave axis position data.
- If master axis stroke end position is specified for the stroke type, the master axis stroke length (single cycle) of electronic CAM tables becomes [master axis stroke end position – synchronization start master axis position]. The relationship of the master axis position with electronic CAM table phases is as follows: phase 0 corresponds to synchronization start master axis position, and the positive phase direction is the direction to move from synchronization start master axis position toward master axis stroke end position.
- If master axis stroke length is specified for the stroke type, the relationship of the master axis position with electronic CAM table phases is as follows: phase 0 corresponds to synchronization start master axis position, and the positive phase direction corresponds to master axis coordinate positive move direction when the stroke length has a positive coordinate value and the minus phase direction corresponds to master axis coordinate positive move direction when the stroke length has a negative coordinate value.

⚠ Caution

(1) If there are any errors in synchronous electronic CAM movement settings, which are specified in variables of operand 2, error No. 4B4 “Extension motion control board synchronous electronic CAM movement setting error” occurs. Info.2 of the error list indicates the variable number of the invalid setting (hexadecimal display).

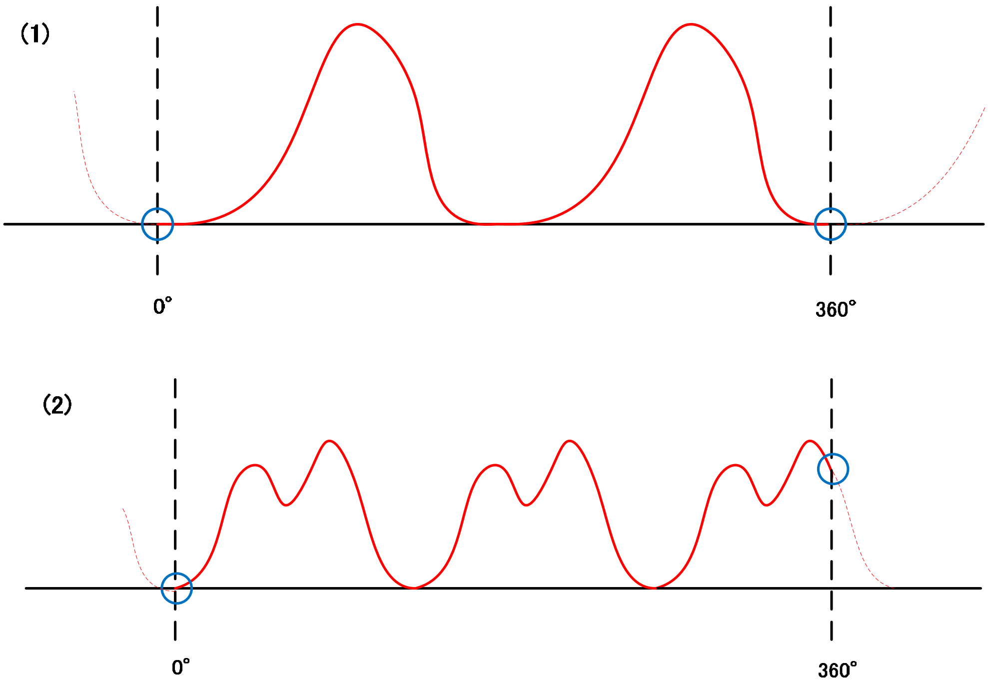
When Error Occurred Variable = “Master Axis No.” Stored Variable (n+2);
- Specified master axis number is invalid or invalid.
- Specified master axis is a synchronized-controlled slave axis or ZR unit axis.
- The specified master axis and slave axis are the same axis.

When Error Occurred Variable = “Stroke Length / Stroke End Position Stored Position No.” Stored Variable (n+5);
- The specified position No. is invalid or position data is invalid.

(2) If the master axis type of synchronous electronic CAM movement settings is specified to be main CPU control axis, the BASE command settings become valid for the master axis number. Moreover, the GRP command settings are invalid for position data storing stroke length and stroke end position.
(3) If slave axes started synchronization while the master axis is moving, excessive speed and/or acceleration and deceleration may occur, leading to an error. Lower the speed and/or acceleration of the master axis when the slave axis synchronization movement is started.
(4) Excessive speed and/or acceleration/deceleration may occur during movement according to the electric CAM table, leading to an error. In this case, change the speed, acceleration/deceleration, and electronic CAM tables of the master axis to set speed and acceleration/deceleration permitted to the axis.
(5) The right to use slave axes that started synchronization is owned exclusively by the slave axis movement program that executed the XCAS command until the XSYE command (stopping synchronous movement) is executed or the program is finished. For this reason, if the axes are used by other programs, error No. C66 “Duplication error” occurs. Moreover, error No. C66 occurs even if a movement command is executed on axes that finished synchronous movement, even in the same program. Execute the XSYE command in order to perform the next movement after synchronous movement.

(6) If the master axis is a controlled to short cut to rotation move axis, the shortcut turn control valid axis passes the position of 0deg (360deg), the command position (and current position) will switch between 0deg and 360deg. In case the master axis of the synchronizing operation makes such operation, it is necessary to set master axis stroke and the profile of the cam table so the displacement matches at the position of 0deg and 360deg.

Explaining in the example of the figure below, the slave axes will operate with no problem even when the master axis passes 0deg (360deg) as the displacement matches at 0deg and 360deg in the case of (1). In the case of (2), the displacement fluctuates suddenly when the master axis passes the position of 0deg (360deg) as the displacement is not matched, and Error No. 4B9 “Synchronizing Command Acceleration/Deceleration Excess Error”, Error No. 4B8 “Synchronizing Command Velocity Excess Error” or sudden move could be caused.



Program Example

[Example] This program example immediately starts synchronization to the main axis of the standard motion control axis. It is executed repeatedly until the main axis reaches the stroke end. A program is required for each slave axis.

| | | | |
|---|---|---|---|
| LET | 200 | 1 | It sets the synchronizing type = 1 (Immediate start, Repeat operation) to Variable No. 200. |
| LET | 201 | 0 | It sets the master shaft type = 0 (Main CPU control shaft) to Variable No. 201. |
| LET | 202 | 1 | It sets the master shaft No. = 1 to Variable No. 202. |
| LET | 203 | 0 | It sets the electronic cam table No. = 1 to Variable No. 203. |
| LET | 204 | 1 | It sets the stroke type = 1 (Master stroke end position specification) to Variable No. 204. |
| LET | 205 | 2 | It sets the master shaft stroke end position storage position No. = 2 to Variable No. 205. |
| LET | 206 | 0 | It sets the slave stroke length storage position No. = 0 to Variable No. 206. |
| LET | 207 | 0 | It sets 0 to Variable No. 207 (Unused data) |
| LET | 208 | 0 | It sets 0 to Variable No. 208 (Reserved area) |
| LET | 209 | 0 | It sets 0 to Variable No. 209 (Reserved area) |
| XAXS | 0 | 1 | It specifies the pulse I/O board 0th axis. |
| XSON | | | It turns the pulse I/O board 0th axis servo ON. |
| XHOM | | | It returns the pulse I/O board 0th axis to home return. |
| XCAS | 0 | 200 | It starts the synchronizing electronic cam operation for the pulse I/O board 0th axis with the synchronizing electronic cam operation settings specified in Variables No. 200 to 209. |
| TAG | 1 | | |
| MOVP | 2 | | It moves the XSEL control master axis to Position No. 2. |
| MOVP | 1 | | It moves the XSEL control master axis to Position No. 1 |
| GOTO | 1 | | |

● XCAS (Start synchronizing extension motion control board axis electronic cam (indicating main axis) electronic CAM control) (2/2)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XCAS | Slave shaft number | Variable number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | × | × | × | × | × | × | × | × |



[Function] It starts the synchronizing process with the axis specified in Operand 1 as the slave shaft following the electronic cam table. The synchronizing electronic cam operation settings such as the master shaft on the electronic cam table are to be specified in ten variables in a row from Variable No. n in Operand 2. The output section turns ON when the synchronizing process is started.

■Operand 2 : Synchronizing Electronic Cam Operation Settings

<table>
<tr><th>Variable No.</th><th>Data name</th><th colspan="2">Description</th></tr>
<tr><td>n</td><td>Synchronizing type</td><td colspan="2">See below</td></tr>
<tr><td>n+1</td><td>Master shaft type</td><td colspan="2">0 : Main CPU control shaft<br>1 : Pulse I/O board control shaft<br>2 : Pulse input channel</td></tr>
<tr><td rowspan="4">n+2</td><td rowspan="4">Master shaft number</td><td>Master shaft type</td><td></td></tr>
<tr><td>0</td><td>1 to 6 axis (XSEL-P/Q), 1to 8 axis (XSEL-R/S)</td></tr>
<tr><td>1</td><td>0 to 15 axis</td></tr>
<tr><td>2</td><td>0 to 3 channel</td></tr>
<tr><td>n+3</td><td>Electronic cam table number</td><td colspan="2">*0 to</td></tr>
<tr><td>n+4</td><td>Stroke type</td><td colspan="2">0 : Master shaft stroke length indication<br>1 : Master shaft stroke end position indication</td></tr>
</table>

<table>
<tr><th>Variable No.</th><th>Data name</th><th colspan="3">Description</th></tr>
<tr><td rowspan="7">n+5</td><td rowspan="7">Master shaft stroke length / stroke end position (Storage position number)</td><td>Stroke type</td><td>Master shaft type</td><td></td></tr>
<tr><td rowspan="3">0</td><td>0</td><td>Master shaft stroke length storage position number<br>* Indicate the main CPU control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td>1</td><td>Master shaft stroke length storage position number<br>* Indicate the pulse I/O board control master shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td>2</td><td>Master shaft stroke length (pulse unit)</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>Master shaft stroke end position storage position number<br>* Indicate the main CPU control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td>1</td><td>Master shaft stroke end position storage position number<br>* Indicate the pulse I/O board control master shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td>2</td><td>Master shaft stroke end position indication (pulse unit)</td></tr>
<tr><td>n+6</td><td>Slave stroke length storage position number</td><td colspan="3">Indicate the pulse I/O board control slave shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td rowspan="4">n+7</td><td rowspan="4">Master shaft synchronizing start position (Storage position number)<br>* Effective only when "Master shaft reaches specified synchronizing start position" is selected for synchronizing type</td><td colspan="2">Master shaft type</td><td></td></tr>
<tr><td colspan="2">0</td><td>Master shaft synchronizing start position storage position number<br>* Indicate the main CPU control master shaft number (from 0 to Max. position No.)</td></tr>
<tr><td colspan="2">1</td><td>Master shaft synchronizing start position storage position number<br>* Indicate the pulse I/O board control master shaft position number (from 0 to Max. position No.)</td></tr>
<tr><td colspan="2">2</td><td>Master shaft synchronizing start position (pulse unit)</td></tr>
<tr><td>n+8</td><td>Reserved</td><td colspan="3">Make sure to set 0</td></tr>
<tr><td>n+9</td><td>Reserved</td><td colspan="3">Make sure to set 0</td></tr>
</table>

■Synchronizing Type (Variable No.n)

| Set value | Description | |
|---|---|---|
| | Synchronizing start type | Synchronizing process repeat type |
| 0 | Immediately | Operate for 1 cycle |
| 1 | Immediately | Repeated operation |
| 2 | Master shaft reaches specified synchronizing start position | Operate for 1 cycle |
| 3 | Master shaft reaches specified synchronizing start position | Repeated operation |

The synchronizing process continues until:
- XSYE Command (to cancel synchronizing process) is executed,
- an operation cancel is executed to the slave shaft (XSTP Command, CANC Command),
- Synchronizing Process Repeat Type is set to 1 cycle and the master shaft reaches to the stroke end, or the slave shaft operation program that XCAS Command is executed is over.

- If the master axis is a main CPU control axis or pulse I/O board control axis, set the master stroke length/stroke end position and master shaft synchronizing start position to the master shaft position data. If the master shaft is the pulse input channel, set it directly to the variable for operation settings. Set the slave shaft stroke length to the slave shaft position data.
- If Stroke Type = Indicate master stroke end position, the master stroke length (1 cycle) on the electronic cam table is [Master shaft stroke end position-Synchronizing start master shaft position]. The relation between the master shaft position and the electronic cam table phase is that the synchronizing start master shaft position is the phase 0, and the direction from the synchronizing start master shaft position to the master shaft stroke end position is the phase positive direction.
- If Stroke Type = Indicate master shaft stroke length, the relation between the master shaft position and the electronic cam table is that the synchronizing start master shaft position is the phase 0, and if the stroke length is a positive value, the positive direction on the master axis coordinate is the phase positive direction and, if the stroke length is a negative value, the positive direction on the master shaft coordinate is the phase negative direction.



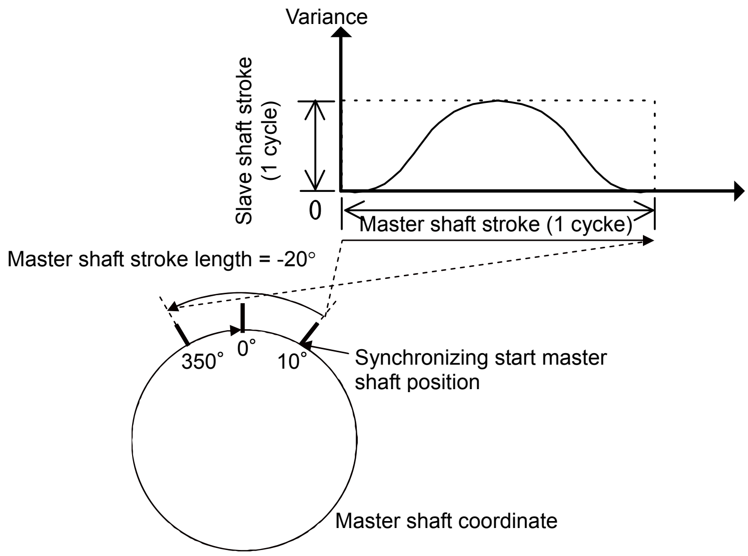
⚠ Caution

- If there is a mistake in the synchronizing electronic cam operation settings specified in the variables in Operand 2, Error No. 455 “Extension motion control board synchronizing electronic cam operation setting error” would occur. The variable numbers with an error setting will be shown on the error list in Info. 2 (in hexadecimal numbers).
  - Variable with an error occurred: Master axis number (Stored variable)
    - The specified master shaft number is inappropriate or invalid
    - The specified master shaft is a synchronized slave shaft or ZR unit shaft (if the master shaft is a main CPU control shaft)
    - The specified master shaft is the shaft specified as the slave (if the master shaft is a pulse I/O board control shaft)
  - The specified master shaft is on a different pulse I/O board or channel from that the slave shaft is on (if the master shaft is a pulse I/O board control shaft or a pulse input channel)
  - Variable with an error occurred: stroke length/stroke end position storage position number
    - The specified position number is inappropriate or position data is invalid
- If the master shaft type for the synchronizing electronic cam operation settings is the main CPU control shaft, BASE Command settings would be effective to the master shaft number. Also, GRP Command settings are invalid to the position data to store the stroke length and stroke end position.
- If the robot is equipped with multiple pulse I/O board, the electronic cam table which is stored to the board that the slave shaft is connected to would be used.
- If the slave shaft starts to move for a synchronizing process during the master shaft is in move, the speed and acceleration/deceleration may get too high and may cause an error. Lower the settings for the speed and acceleration of the master shaft during the slave synchronizing movement starts.
- During a movement following the electronic cam table, the speed and acceleration/deceleration may get too high and may cause an error. Change the settings for the speed, acceleration/deceleration and electronic cam table so they are set to the allowable speed and acceleration/deceleration for the shaft.

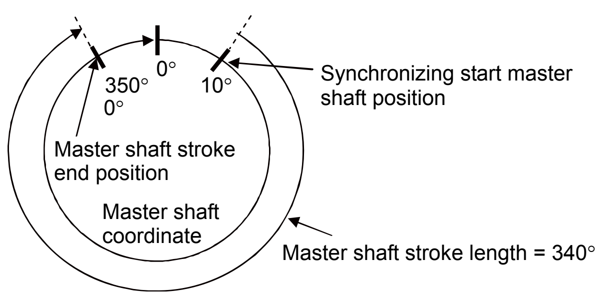
⚠ Caution

- Once the slave shaft starts synchronizing, it will be occupied by the program until XSYE Command (to cancel synchronizing) is executed or the slave shaft operation program that XCAS Command is executed is finished.
  Therefore Error No. 449 “Extension motion control board shaft duplication use error” would occur if the shaft is used by another program. Also, even in the same program, Error No. 449 would occur if an operation command is executed to the shaft that the synchronizing process is already completed. Execute XSYE Command if a next operation is required after the synchronizing process is finished.
  For XSEL-R/S, the number of the main CPU control axes that can be the master axis for the synchronizing operation of such as the electronic cam is six axes at the maximum.
  The axes that can be the master axis can be selected in I/O Parameter No. 529 “Extension Motion Control Board Synchronizing Main CPU Control Master Select Axis Pattern” (dynamic change not available). Please refer to the parameter list in XSEL-R/S Instruction Manual for the details of the parameter.
  When indicating an axis that is not selected as the main axis select in I/O Parameter No. 529 as the main axis in XCAS Command, Error No. 455 “Extension Motion Control Board Synchronizing Electronic Cam Operation Setting Error” will occur.
- If the master shaft is the main CPU control axis with the rotation axis close control, set the master shaft stroke type in the synchronizing electronic cam operation settings to the stroke length setting. If the setting is specified to the stroke end position, it may not perform a synchronizing process that is expected.

  [Example] For the synchronizing process in range of master shaft position = 10° to 350°




  If the stroke end position = 350° assuming the master shaft stroke type = stroke end position, the figure will be as shown below:



Program Example

| | | | |
|---|---|---|---|
| LET | 200 | 1 | It sets the synchronizing type = 1 (Immediate start, Repeat operation) to Variable No. 200. |
| LET | 201 | 0 | It sets the master shaft type = 0 (Main CPU control shaft) to Variable No. 201. |
| LET | 202 | 1 | It sets the master shaft No. = 1 to Variable No. 202. |
| LET | 203 | 0 | It sets the electronic cam table No. = 1 to Variable No. 203. |
| LET | 204 | 1 | It sets the stroke type = 1 (Master stroke end position specification) to Variable No. 204. |
| LET | 205 | 2 | It sets the master shaft stroke end position storage position No. = 2 to Variable No. 205. |
| LET | 206 | 0 | It sets the slave stroke length storage position No. = 0 to Variable No. 206. |
| LET | 207 | 0 | It sets 0 to Variable No. 207 (Unused data) |
| LET | 208 | 0 | It sets 0 to Variable No. 208 (Reserved area) |
| LET | 209 | 0 | It sets 0 to Variable No. 209 (Reserved area) |
| XAXS | 0 | 1 | It specifies the pulse I/O board 0th axis. |
| XSON | | | It turns the pulse I/O board 0th axis servo ON. |
| XHOM | | | It returns the pulse I/O board 0th axis to home return. |
| XCAS | 0 | 200 | It starts the synchronizing electronic cam operation for the pulse I/O board 0th axis with the synchronizing electronic cam operation settings specified in Variables No. 200 to 209. |
| TAG | 1 | | |
| MOVP | 2 | | It moves the XSEL control master axis to Position No. 2. |
| MOVP | 1 | | It moves the XSEL control master axis to Position No. 1 |
| GOTO | 1 | | |

● XCTM (Extension motion control Single Electronic Cam (Time Specification) Movement)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XCTM | Slave shaft number | Variable number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | × | × | × | × |

[Function] It performs a movement following the electronic cam table having the time axis of the shaft specified in Operand 1 as the master shaft. Single electronic cam operation settings such as the movement time and the electronic cam table number, etc., are to be specified in five variables in a row from Variable No. n in Operand 2. The output turns OFF at the same time the command is started and turns ON when the movement is complete.

■Operand 2 : Single Electronic Cam Operation Settings

| Variable No. | Data | Description |
|---|---|---|
| n | Electronic cam table number | |
| n+1 | Moving time | Unit in 0.001sec |
| n+2 | Slave stroke length storage position number | * Indicate the pulse I/O board control slave shaft position number (from 0 to Max. position No.) |
| n+3 | Reserved | Set 0 |
| n+4 | Reserved | Set 0 |

Set the slave shaft stroke length to the slave shaft position data.
When the slave shaft stroke length is a positive value, the movement is made to the coordinate positive direction. When the slave shaft stroke length is a negative value, the movement is made to the coordinate positive direction.

⚠ Caution

- If there is a mistake in the single electronic cam operation settings specified in the variables in Operand 2, Error No. 456 “Extension motion control board single electronic cam operation settings error” (XSEL-P/Q/PCT/QCT/R/S), Error No. 4B6 “Extension shaft operation settings error”(XSEL-RA/SA) would occur. The variable numbers with an error setting will be shown on the error list in Info. 2 (XSEL-P/Q/PCT/QCT/R/S) or Info. 1 (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) (in hexadecimal numbers).

  - Variable with an error occurred: Slave stroke length storage position number (Stored variable)
  - Specified position number is inappropriate or the position data is inefficient.

- If the robot is equipped with multiple pulse I/O board, the electronic cam table which is stored to the board that the slave shaft is connected to would be used.

- During a movement following the electronic cam table, the speed and acceleration/deceleration may get too high and may cause an error. Change the settings for the moving time and electronic cam table so they are set to the allowable speed and acceleration/deceleration for the shaft.

Program Example

| | | | |
|---|---|---|---|
| LET | 200 | 0 | It sets the electronic cam table No. = 0 to Variable No. 200. |
| LET | 201 | 1000 | It sets the movement time 1000ms to Variable 201. |
| LET | 202 | 0 | It sets the slave stroke length storage position No. = 0 to Variable No. 202. |
| LET | 203 | 0 | It sets 0 to Variable No. 203 (Reserved area) |
| LET | 204 | 0 | It sets 0 to Variable No. 204 (Reserved area) |
| XAXS | 0 | 1 | It specifies the pulse I/O board $0^{th}$ axis. |
| XSON | | | It turns the pulse I/O board $0^{th}$ axis servo ON. |
| XHOM | | | It returns the pulse I/O board $0^{th}$ axis to home position. |
| XCTM | 0 | 200 | It starts the single electronic cam operation for the pulse I/O board $0^{th}$ axis with the single electronic cam operation settings specified in Variables No. 200 to 204. |

● XSFS (Extension Motion Montrol control Electronic Shaft Synchronizing Start)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XSFS | Slave shaft number | Variable number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | × | × | × | × |



[Function] It starts the synchronizing operation of the shaft specified in Operand 1 as the slave shaft following the master shaft. The electronic shaft operation settings such as the master shaft are to be specified in the four variables in a row from Variable No. n in Operand 2. The output section turns ON with the synchronizing start.

■Operand 2 : Single Electronic Shaft Operation Settings

| Variable number | Data | Description |
|---|---|---|
| n | Master shaft type | 0 : Standard motion control axis (Main CPU control shaft)<br>1 : Extension motion control axis<br>2 : Pulse input channel (Applicable of pulse input channels is, XSEL-P/Q/PCT/QCT/R/S) |
| n+1 | Master shaft number | * Axes 1 to 6 (XSEL-P/Q), 1 to 8 axis (XSEL-R/S, XSEL-RA/SA) when standard motion control axis (Main CPU control shaft), axes 0 to 15 when extension motion control axis, channels 0 to 3 when pulse input channel (Applicable of pulse input channels is, (XSEL-P/Q/PCT/QCT/R/S) |
| n+2 | Gear ratio numerator | -10000 to -1, 1 to 10000 (XSEL-P/Q/PCT/QCT/R/S)<br>-999999 to -1, 1 to 999999 (XSEL-RA/SA),<br>-99999999 to -1,1 to 99999999 |
| n+3 | Gear ratio denominator | 1 to 10000 (XSEL-P/Q/PCT/QCT/R/S)<br>1 to 99999999 (XSEL-RA/SA) |

The synchronizing process continues until:
• XSYE Command (to cancel synchronizing process) is executed,
• an operation cancel is executed to the slave shaft (XSTP Command, CANC Command),
• Synchronizing Process Repeat Type is set to 1 cycle and the master shaft reaches to the stroke end, or the slave shaft operation program that XSFS Command is executed is over.

If the gear ratio is set to a negative value, the slave shaft operates in a reverse direction against the master shaft.

⚠ Caution

(1) If there is a mistake in the single electronic cam operation settings specified in the variables in Operand 2, Error No. 456 “Extension motion control board single electronic cam operation settings error” (XSEL-P/Q/PCT/QCT/R/S), Error No. 4B6 “Extension shaft operation settings error”(XSEL-RA/SA) would occur. The variable numbers with an error setting will be shown on the error list in Info. 2 (XSEL-P/Q/PCT/QCT/R/S) or Info. 1 (XSEL-RA/SA/RAX/SAX/RAXD/SAXD) (in hexadecimal numbers).

- Variable with an error occurred : Master shaft number
  - The specified master shaft number is inappropriate or invalid
  - The specified master shaft is a synchronized slave shaft or ZR unit shaft (if the master shaft is a XSEL control shaft)
  - The specified master shaft is the shaft specified as the slave (if the master shaft is a pulse I/O control shaft)
  - The specified master shaft is on a different pulse I/O or channel from that the slave shaft is on (if the master shaft is a pulse I/O control shaft or a pulse input channel)

(2) If the master shaft type in the electronic shaft operation settings is the standard motion control axis (main CPU control shaft), BASE Command would be effective to the master shaft number. (XSEL-P/Q/PCT/QCT/R/S)

(3) If the master shaft is the pulse input channel, the slave shaft operates with a condition taking 1 input pulse from the pulse input channel as 0.001mm. (XSEL-P/Q/PCT/QCT/R/S)

(4) If the slave shaft starts to move for a synchronizing process during the master shaft is in move, the speed and acceleration/deceleration may get too high and may cause an error. Lower the settings for the speed and acceleration of the master shaft during the slave synchronizing movement starts.

(5) During a master shaft, the speed and acceleration/deceleration may get too high and may cause an error. Change the settings for the speed, acceleration/deceleration and gear ratio so they are set to the allowable speed and acceleration/deceleration for the shaft.

(6) Once the slave shaft starts synchronizing, it will be occupied by the program until XSYE Command (to cancel synchronizing) is executed or the slave shaft operation program that XSFS Command is executed is finished.
Therefore Error No. 449 “Extension motion control board shaft duplication use error” (XSEL-P/Q-PCT/QCT/R/S), Error No. C66 “Shaft duplication use error” (XSEL-RA/SA) would occur if the shaft is used by another program. Also, even in the same program, Error No. 449 would occur if an operation command is executed to the shaft that the synchronizing process is already completed. Execute XSYE Command if a next operation is required after the synchronizing process is finished.

(7) For XSEL-R/S, the number of the main CPU control axes that can be the master axis for the synchronizing operation of such as the electronic cam is six axes at the maximum.
The axes that can be the master axis can be selected in I/O Parameter No. 529 “Extension Motion Control Board Synchronizing Main CPU Control Master Select Axis Pattern” (dynamic change not available). Please refer to the parameter list in XSEL-R/S Instruction Manual for the details of the parameter.
When indicating an axis that is not selected as the main axis select in I/O Parameter No. 529 as the main axis in XCAS Command, Error No. 457 “Extension Motion Control Board Electronic Cam Operation Setting Error” will occur.

| | | | | |
|---|---|---|---|---|
| [Example] | LET | 200 | 0 | It sets the master shaft type = 0 (Main CPU control shaft) to Variable No. 200. |
| | LET | 201 | 1 | It sets the master shaft No. = 1 to Variable No. 201. |
| | LET | 202 | 1 | It sets the gear ratio numerator = 1 to Variable No. 202. |
| | LET | 203 | 50 | It sets the gear ratio denominator = 1 to Variable No. 203. |
| | XSFS | 0 | 200 | It starts electronic shaft synchronizing process of the pulse I/O board $0^{th}$ shaft with the electronic shaft operation settings specified in Variables No. 200 to 203. |
| | TAG | 1 | | |
| | MOVP | 2 | | It moves the main CPU control master axis to Position No. 2. |
| | MOVP | 1 | | It moves the main CPU control master axis to Position No. 1. |
| | GOTO | 1 | | |

● XSYE (Extension motion control synchronizing process complete)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XSYE | Slave shaft number | (Complete type) | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | × | × | × | × |

[Function] It finishes the synchronizing process of the slave shaft specified in Operand 1. This command is effective to the slave shaft that is in the synchronizing process with the synchronizing electronic cam (master shaft specified) started by the same program or that in electronic shaft synchronizing process (XSFS Command). If another shaft is specified, Error No. 444 “Extension motion control board axis number error” (XSEL-P/Q-PCT/QCT/R/S), No. C2F “Axis number error” (XSEL-RA/SA) would occur.
It is able to specify the complete type in Operand 2. If 0 is select or no selection is done, the synchronizing process would be cancelled. If 1 is selected, it waits for the synchronizing process to finish. It is applicable in a case to wait for the slave shaft to finish its synchronizing process with “Synchronizing type = operate for 1 cycle” in XCAS Command or to wait till the synchronizing process to be cancelled by XSTP Command from another program.
The output section turns ON when 1 is selected for the complete type and the slave shaft finishes 1 cycle of operation with the synchronizing type = “Operate for 1 cycle”.
■Operand 2 : Complete Type
= 0 or no selection: Synchronizing process to be cancelled
= 1: Wait for synchronizing process to finish

⚠ Caution
- In the case “Synchronizing process to be cancelled” is chosen for the complete type, the command would only finishes the synchronizing process and would not confirm the slave shaft positioning completion (Slave Driver positioning complete signal). If waiting for the positioning to complete is desired, wait till the positioning complete signal input port turns ON directly by WTON Command after XSYE Command execution.
- By executing this command the occupation of the shaft is released. Thus, the slave shaft being in the synchronizing process can be used for other programs.

Program Example

| | | | |
|---|---|---|---|
| LET | 200 | 1 | It sets the synchronizing type = 1 (Immediate start, operate for 1 cycle) to Variable No. 200. |
| LET | 201 | 0 | It sets the master shaft type = 0 (Main CPU control shaft) to Variable No. 201. |
| LET | 202 | 1 | It sets the master shaft No. = 1 to Variable No. 202. |
| LET | 203 | 0 | It sets the electronic cam table No. = 1 to Variable No. 203. |
| LET | 204 | 1 | It sets the stroke type = 1 (Master stroke end position specification) to Variable No. 24. |
| LET | 205 | 2 | It sets the master shaft stroke end position storage position No. = 2 to Variable No. 205. |
| LET | 206 | 0 | It sets the slave stroke length storage position No. = 0 to Variable No. 206. |
| LET | 207 | 0 | It sets 0 to Variable No. 207 (Unused data) |
| LET | 208 | 0 | It sets 0 to Variable No. 208 (Reserved area) |
| LET | 209 | 0 | It sets 0 to Variable No. 209 (Reserved area) |
| XCAS | 0 | 200 | It starts the synchronizing electronic cam operation for the pulse I/O board $0^{th}$ axis with the synchronizing electronic cam operation settings specified in Variables No. 200 to 209. |
| XSYE | 0 | 1 | It waits till the pulse I/O board $0^{th}$ shaft to operate 1 cycle for the synchronizing electronic cam operation. (Assuming the master shaft is operated by another program) |
| XAXS | 0 | 1 | It set the pulse I/O board 0th shaft to the axis pattern. |
| XMVP | 10 | | It moves the specified axis to Position No. 10. |

● XAST (Acquire extension motion control axis status)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | XAST | Variable number | Axis number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | ○ | × | × | × |

[Function] It reads the axis status specified in Operand 2 into the variable in Operand 1.

Bit Construction of Axis Status (XSEL-P/Q/PCT/QCT/R/S)

| Bit | Information |
|---|---|
| 27-31 | Reserved |
| 26 | Axis Alarm (error detected by XSEL)<br>* “Axis Alarm” includes “Continuous Impossible Alarm Occurrence (error detected in slave controller)” as well as the errors related to the extension motion control board of XSEL. |
| 25 | Axis in use |
| 24 | Reserved |
| 14-23 | Reserved |
| 13 | Reserved |
| 12 | Reserved |
| 11 | Reserved |
| 10 | Reserved |
| 9 | Safety speed valid status (safety speed is valid for XSEL) |
| 8 | Reserved |
| 7 | Reserved |
| 6 | Reserved |
| 5 | Reserved |
| 4 | Servo ON status. |
| 3 | Reserved |
| 2 | Home return completion<br>* The bit rises when the home-return operation of XSEL controller is finished and the home-return complete status on the slave driver turns on. |
| 1 | Point position completion<br>* The bit rises when an operation command of XSEL controller is finished and the positioning complete status on the slave driver turns on. |
| 0 | Continuity disabled alarm is generated (an error the slave driver generated) |

Bit structure of axis status (XSEL-RA/SA/RAX/SAX/RAXD/SAXD)

| Bit | Description | Valid (o), Invalid (x) | |
|---|---|---|---|
| | | Slave driver | Standard driver |
| 27-31 | Reserved | - | - |
| 26 | Axis alarm (error detected by X-SEL) (* 1) | ○ | ○ |
| 25 | Axes being used | ○ | ○ |
| 24 | Reserved | - | - |
| 16-23 | Reserved | - | - |
| 15 | Operation Mode Status (0=AUTO/1=MANU) | ○ | × |
| 14 | Overload Warning | ○ | × |
| 13 | Zone 2 | ○ | × |
| 12 | Zone 1 | ○ | × |
| 11 | Reserved | - | - |
| 10 | Reserved | - | - |
| 9 | Safety speed valid status (X-SEL safety status valid) | ○ | ○ |
| 8 | Battery Voltage Drop (* 2) | ○ | × |
| 7 | Reserved | - | - |
| 6 | Reserved | - | - |
| 5 | Reserved | - | - |
| 4 | Servo On status | ○ | ○ |
| 3 | Reserved | - | - |
| 2 | Home return completion | ○ | ○ |
| 1 | Position complete (* 3) | ○ | ○ |
| 0 | Operation Unable Alarm Occurred<br>(At slave driver control: error on slave controller) | ○ | ○ |

*1 : "Axis Alarms" includes the state of "Operation Unable Alarm Occurred" (error on slave controller) as well as those errors detected in the XSEL.
*2 : It depends on the connected slave if this status becomes valid or not. If it is necessary to use, refer to the MECHATROLINK-III instruction manual of the RC controller to be used, and check if it is supported.
*3 : The bit turns on when the operation command to the XSEL controller is finished and the positioning complete status on the slave controller has turned on.

[Example] XAST 200 10 Acquire the status of Axis No. 10 to Variable 200.

Assuming 8212 (decimal number) was in Variable 200 after this command was executed;

8212 (decimal number) → 10 0000 0001 0100 (binary number)

| (Bits) | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

and the status of home-return complete (bit 2), servo-on (bit 4) and zone 2 (bit 13) should be on.

[Reference] It can be confirmed that the indicated bit is on by using AND Command.

AND 200 16 900 It should be the logical conjunction of Variable 200 and 16 (*decimal number).
Flag 900 will turn ON when the result of logical conjunction is 0.

16 (decimal number) = 10000 (binary number) By executing the step described above,
- When Bit 4 is ON : Flag 900 turns OFF
- When Bit 4 is OFF : Flag 900 turns ON

against the value of Variable 200.

● XACH (Extension motion control axis arch motion)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | XACH | Position number | Position number | PE |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | ○ | × | × | × | ○ | × | × | × |

[Function] Arch Motion gets performed from the current position to the target position.

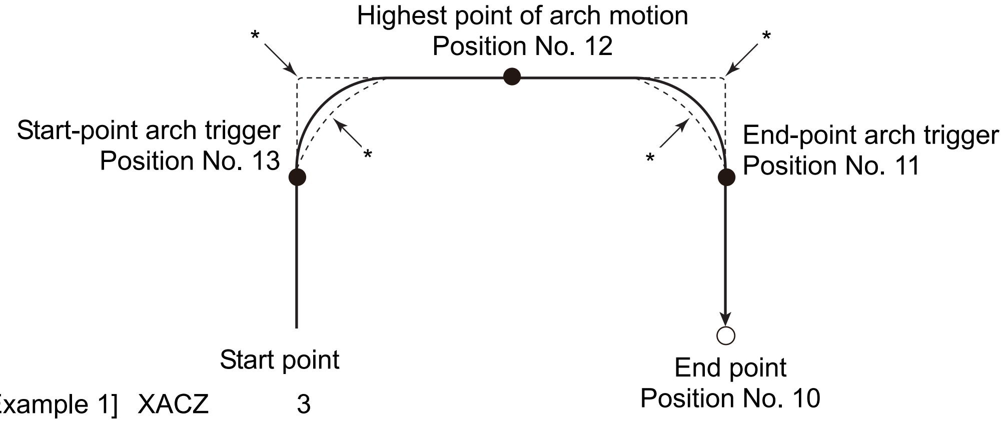
- Movement will be made with arch motion (CP Movement) to the position indicated in Operand 1.
- Movements in directions other than the arch-motion Z-axis direction will begin after rising from the current position to the start-point arch trigger. With Z-axis position in the position indicated in Operand 2 as the highest point, after all the movement except for the direction of the arch motion Z-axis is finished, the actuator passes near the end point arch trigger and reaches the indicated position.
- The setting of the arch motion Z-axis by XACZ command and the arch trigger by XATG command are necessary.

(Note 1) Set axis patterns using the XAXS or XA16 command before executing this command. If the command has not been executed, error No. 4B3 "No extension motion control board axe pattern setting error" occurs. The arch motion Z-axis and arch motion composited axes (when arch motion composition is activated) should be included in the indicated axis patterns. Error No. CB4 "Palletized Z-Axis Non-Declaration Error" will occur if there is no arch motion Z-axis included in the axis patterns.

(Note 2) This command requires indication of the velocity, acceleration and deceleration in VAL, VLMX, ACC, DCL commands before execution. If those commands have not been executed, an error occurs.

(Note 3) The positioning complete band is valid only for the arch motion Z-axis in the position indicated in Operand 1. It should be defined as positioning complete when the actuator goes down along the Z-axis to the target position and reaches the indicated positioning complete band.

(Note 4) It should be CP operation.

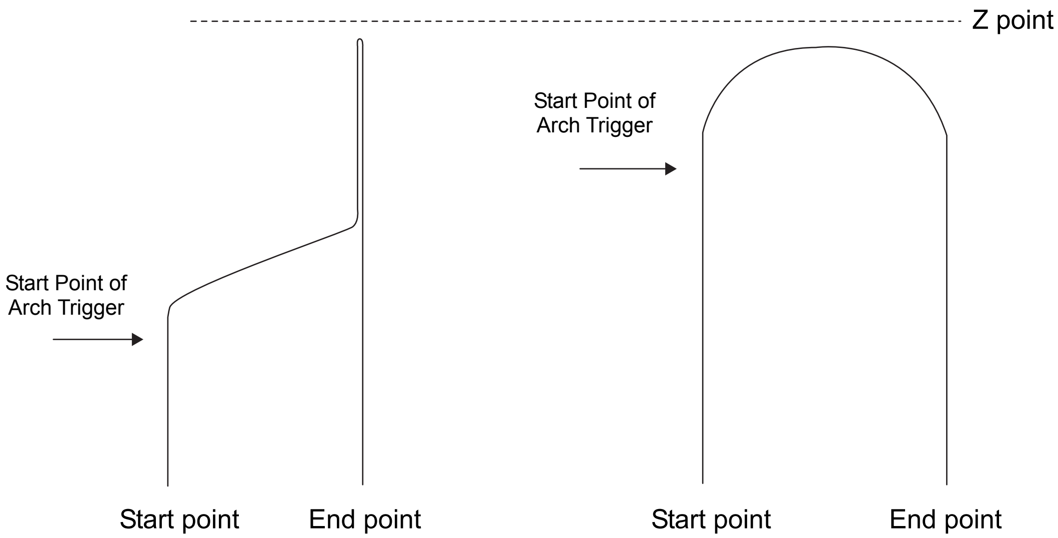



[Example 1] XACZ 3

XATG 13 11
•
•
•
XACH 10 12

* When the operation is resumed after a pause, depending on the position where the operation is resumed the locus may follow the lines (dotted lines) indicated by asterisks in the diagram for the composite section from ascent to horizontal movement or from horizontal movement to descent. Be careful not to cause interference.

- The arch-motion Z-axis coordinate of the end point will become the arch-motion Z-axis component of the position data specified in operand 1, if any, plus the arch-motion Z-axis offset. If there is no arch-motion Z component, the arch-motion Z-axis coordinate of the end point will become the arch-motion Z-axis coordinate of the start point plus the arch-motion Z-axis offset.
- An error will generate if the start-point arch trigger is set below the start point or the end-point arch trigger is set below the end point. (Note: Up/down has nothing to do with +/- on the coordinate system.)
- The arch-motion Z-axis up direction refers to the direction toward the Z point from the start point (the down direction refers to the opposite direction), and has nothing to do with the size of coordinate value. Therefore, be sure to confirm the actual operating direction when using this command.
- The arch-motion Z-axis will come down after a rise-process command value is output. Therefore, one of the following operations will be performed depending on how the arch-trigger point and Z point are set.
  If the resulting operation is undesirable, change the arch trigger and/or Z point to improve the efficiency of movement.




- As for the end position data, movement also starts/ends above the applicable arch trigger for any effective axis, other than the arch motion Z-axis, if data of such axis is included in the position data.
- If there is an arch motion composition setting, and if there is valid axis data except for the end point data and the arch motion Z-axis, this axis will also make operation. In this case, movement of the axis also starts/ends above the applicable arch trigger.

● XACZ (Extension motion control axis arch motion Z-axis declaration)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | XACZ | Axis number | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | ○ | × | × | × | ○ | × | × | × |

[Function] Indicate the axis number in the arch motion Z-axis when operation is made with XACH command. The axis number specified in operand 1 will be set as the axis number representing the arch-motion Z direction. If the output field is specified, the output will turn ON after this command is executed.

[Example] XACZ 3

● XAEX (Extension motion control axis arch motion composition setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | XAEX | (Position number) | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | ○ | × | × | × | ○ | × | × | × |

[Function] Establish the arch motion composition setting when operation is made with XACH command. In this setting, the axis to be moved at the same time as the horizontal operation should be added. Use the position number specified in operand 1 for setting composite motion. While the arch motion is executed, valid axes in the arch motion end position data and the valid axes except for the arch motion Z-axis in the position data indicated by this command will move as the composited axes to the end point coordinate indicated in the position data.
If nothing is specified in operand 1, the position number already declared for setting composite motion becomes invalid.

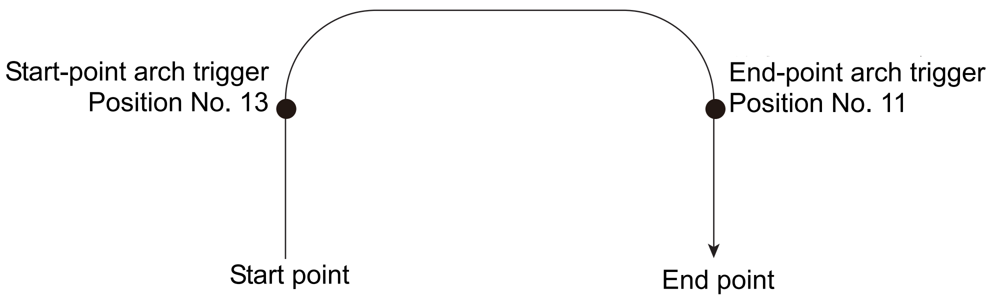
● XATG (Extension motion control axis arch trigger setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | XATG | Position number | Position number | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | ○ | × | × | × | ○ | × | × | × |



[Function] Establish the arch trigger setting when operation is made with XACH command. Set the arch-motion Z-axis position data in the position data specified in operand 1 as the start-point arch trigger, and set the arch-motion Z-axis position data in the position data specified in operand 2 as the end-point arch trigger. For an arch-motion operation, set it so that a horizontal movement will begin when the start-point arch trigger is reached during ascent from the start point, and that the end-point arch trigger will be reached after a horizontal movement is completed during descent. If the output field is specified, the output will turn ON after this command is executed.




[Example] XATG 13 11

● XOAZ (Extension motion control axis arch motion Z-axis offset setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output operation type (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| Optional | Optional | XOAZ | Offset value | Prohibited | CP |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | × | × | ○ | × | × | × | ○ | × | × | × |

[Function] The offset in the arch motion Z-axis direction when operation is made with XACH command is to be set. The value specified in operand 1 will be set as the offset in the arch-motion Z-axis direction. The offset amount is set in mm and the effective resolution is 0.001mm. A negative value can also be specified as the offset, as long as the operation range will not be exceeded. It is the offset valid only to the end point of XACH command. If the output field is specified, the output will turn ON after this command is executed.

## [28] Conveyor Tracking Related Commands

● TRMD (Tracking mode setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TRMD | 0 (Mode OFF)<br>or<br>1 (Mode ON) | Operand 1 = 0<br>Prohibited<br>Operand 1 = 1<br>(TRAC Command timeout time) | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | ○ | ○ | ○ | × | × | × |

[Function] Set the Tracking Mode ON/OFF in Operand 1.
Only when Operand 1 = 1 (Tracking Mode ON), TRAC Command (explained later) timeout time (timeout time until the datum point in the working range exceeds the “minimum work position available for the tracking operation start” after TRAC Command is executed) can be selected in Operand 2. The settable range for the timeout time is settable from 0.00 to 99.00sec. When a selection of no timeout time setting (Operand 2 = not set) is made, TRAC Command defines there is no timeout setting and waits with no time limitation.
Work detection process becomes valid only when Tracking Mode is ON.

• Return Code in TRMD Command (Variable 99 (Local Space))
* When Operand 1 = 0 (Tracking Mode OFF), the return code cannot be returned. (Variable 99 not operated)
* When the return code is the numbers except for 0, Tracking Mode is turned OFF.

0 : Tracking Mode ON (In normal condition)
1 : Vision System initializing incomplete
2 : Ethernet connection incomplete

⚠ Caution : TRMD and TRAC Commands are available only when they are in the same program (task).
Dedicated application software is required when using the conveyor tracking function. Please contact us for the details.

● TRAC (Tracking operation setting & datum point position information obtainment in work)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | TRAC | 0 (Operation OFF) or 1 (Operation (Standby) ON) | Operand 1 = 0 Prohibited Operand 1 = 1 Position number to save the work position information | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | ○ | ○ | ○ | × | × | × |

[Function] Set the Tracking Mode ON/OFF in Operand 1.
It is necessary to specify the work position information storage position number in Operand 2 if Operand 1 = 1 (Tracking operation on setting) is specified.
After the tracking operation command is executed, the datum position information for the identified top work is stored to the position number specified in Operand 2 if the datum point within the work detected and confirmed during the work detection process exceeds “Minimum tracking operation start work position” (if it is already exceeded, at the same time TRAC Command is executed). If the work position information is acquired, move the actuator to the position above the work immediately with MOVL Command with a care to Z-axis (height).

| |
|---|
| Datum Point Position Information in Work Saved in Position Data<br>• X, Y, (rotation) R-axis |

If Tracking Operation ON Command is executed while already in the conveyor tracking operation, the tracking operation will continue and only the datum point position information obtainment in the next work is performed.
When the tracking operation OFF command is executed, the tracking operation is cancelled and it decelerates and stops tracking. If the tracking operation is cancelled by the tracking operation OFF command, etc., the data such as the acquired work datum position information would be invalid (meaningless).

- Return Code in TRAC Command (Variable 99 (Local Space))
  - * When Operand 1 = 0 (Tracking Operation OFF), the return code cannot be returned. (Variable 99 not operated)
  - 0: Tracking operation start & datum point position information obtaining succeeded

1. Datum point in work position information obtaining timeout
   Timeout value should be indicated in Operand 2 of TRMD Command as described previously.
2. Datum point in work position information obtaining timer cancel (Timer cancel by TIMC Command)
3. Reached the maximum work position for tracking operation start (Work reached a position that cannot be tracked)
   Even thought the datum point position information in work can be obtained, the data is no more meaningful, thus ensure not to do the positioning with that position information. The work attribute (it is for the future expansion, currently fixed value = no attribute identification) isaved to the local variable indicated in “All-Axes Parameter No. 93 Tracking Work Attribute Saved Local Variable Number”.
4. Tracking operation stop
   When the work reaches the position to finish the tracking operation, tracking reversed operation workposition, or an error is occurred, the tracking operation gets cancelled (stopped).
5. Tracking Mode Cancelled
   Work detection is set to invalid by Tracking Mode OFF Command or an error and all the existed work information is deleted.

⚠ Caution

1) TRMD and TRAC Commands are available only when they are in the same program (task).
2) Execute the Tracking Operation ON Command on the position where there is no interference to the surroundings with the tracking operation, movement to the point above the datum point in the working range after the tracking, or the combination of both operations.
3) Ensure to use MOVL Command for the movement on the axis during the conveyor tracking operation. If the position data (Datum point position information in Work) is “obtained in normal condition”, perform a positioning quickly with “MOVL” to the point around the position (around the point above the datum point in the work) considering “the target values for the axes that the data is not obtained for, such as Z-axis (height)”. The obtained position data is effective only in “the tracking operation that time” and becomes ineffective after “the tracking operation that time” is complete.
   For SCARA Robot, the robot arm getting close to the area around the peculiar point as the result of conveyor tracking may cause an abnormal acceleration and it is very risky. In the case this abnormal acceleration around the peculiar point is occurred, the arm deceleration angle after the error detection also becomes larger. Do not locate interfering object in the surroundings. The following errors will be detected if the abnormal acceleration around the peculiar point:
   - Error No. B74 CP Operation Limited Area Invasion Error
   - Error No. B91 Main Excess Speed Necessity Error
   - Error No. D09 Driver Excess Speed Error

   To avoid this phenomenon, the work tracking limit can be set in “All-Axes No.75 Tracking Operation Complete Work Position”, however, since it all depends on the work position, there is still a risk that the robot arm reaches the peculiar position unless it is quickly moved to the position (Datum Point in Work) obtained in normal condition by Tracking Operation ON Command.
   - It is also an effective way for the debug test operation when turning the system on to detect in the simple interference check zone before invasion to the peculiar point on the exist side if the operation is under a comparatively low conveyor speed.
   - If reaching to the axis soft limit or interference range due to the positional correlation of “minimum work position for tracking operation start” and “point above datum point for work”, have an appropriate treatment on the sequence to avoid it such as by setting the different start time for the positioning to the point above the work datum point with a timer, etc.
4) Since the right for the servo use is occupied by the TRAC Command execution task during Tracking Operation ON Command, the tracking related servo axis cannot be used from other tasks. (For SCARA Robot, 4 axes are occupied for the purposes of the posture control and other related.)
5) For SCARA Robot, it is operated on the work coordinate system of when the conveyor tracking operation is started during the conveyor tracking operation.
6) Conveyor tracking operation does not stop at the break points of SEL program.
   - The break point only pauses the next program step execution.
7) PUSH Command cannot be used during the conveyor tracking operation.

## [29] Vision System I/F Related Command

● SLVS (Select vision system I/F)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SLVS | Select vision system I/F | (Timeout time) | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | × | × | TT:×, TTA:○ | ○ (PC/PG only) |

[Function] Select whether using Vision System I/F in this command (GTVD Command).

Operand 1 : Select Vision System I/F
0 : Vision System I/F not selected
1 : Vision System I/F selected for use

Operand 2 : Operand 1 = Invalid when set to “0”.· ··· Prohibited
Operand 1 = Except for “0” ··· Timeout time (sec) when GTVD Command is executed
The setting range for the timeout time is from 0.00 to 99.00sec.
When no indication (Operand 2 = blank) is defined, the timeout setting is not established and is set to no limitation.

- Return Code in SLVS Command (Variable 99 (Local Space))

The result in SLVS execution is stored in Variable 99 as a return code.
* No return code will be obtained (Variable 99 not executed) when Operand 1 = 0.
* The return codes not listed below are in common with OPEN Command (for Ethernet connection). Refer to “OPEN Command” in Ethernet Instruction Manual provided separately.

0 : Completed in normal condition
1 : Timeout
(Related Parameters: I/O Parameter No. 127, Network Attribute 8, Bits 0 to 7)
2 : Timer cancelled (condition that the waiting status is cancelled by TIMC Command)
6 : Task Complete (Program complete request, etc.)
(Unable to identify from SEL Command)
23 : Vision System Initializing Incomplete Error

(Note 1) SLVS and GTVD Commands can be executed only on the same program (task).

(Note 2) Executing SLVS Command with Operand 1 = 1 is indicated opens the communication channel that is specified in I/O Parameter No. 351, Bits 4 to 7. And also, executing SLVS Command with Operand 1 = 0 is indicated closes the communication channel that is specified in I/O Parameter No. 351, Bits 4 to 7.

(Note 3) When the Vision System I/F is used with Ethernet, message communication attribute is fixed to client.

[Example 1]

| | | | |
|---|---|---|---|
| SLVS | 1 | | Select Vision System I/F Usage |
| • | | | (GTVD Command Timeout Value = None) |
| • | | | |
| SLVS | 0 | | Cancel Vision System I/F Selection |

[Example 2]

| | | | |
|---|---|---|---|
| SLVS | 1 | 60 | Select Vision System I/F Usage |
| • | | | (GTVD Command Timeout Value = 60sec is indicated) |
| • | | | |
| SLVS | 0 | | Cancel Vision System I/F Selection |

● GTVD (Vision system I/F image-capture data acquirement)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | GTVD | Capturing trigger classification | Variable number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | × | × | × | TT:×, TTA:○ | ○ (PC/PG only) |



[Function] This outputs the image-capture command to the Vision System I/F selected by SLVS Command and stores the received image data to the variables and specification data. With one time of execution of this command, one image data can be obtained.

Operand 1 : Capturing Trigger Classification
0 : Reserved
1 : Immediate Image-Capture Command Output
2 : Image-Capture Command Output when Image-Capture Trigger Port (I/O Port and Flag) is ON
3 to 6 : Reserved

Operand 2 : Variable number*1
Assuming the variable number selected in Operand 2 is n, the contents are stored in the variables of quantity 8 in a row starting from n.
Variable No. n : Top data number for image data work coordinates storage*3
Variable No. n+1 : Variable number for image data work attribute storage
(Note) Make sure the continuous 12 variables after the top variable number are not in use. *2
Variable No. n+2 : Variable number for image data work quantity storage
Variable No. n+3 : Image-capture trigger port number
(Valid only when Operand 1 = 2 is input)
Variable No. n+4 : Data Classification for Image Data Storage
0 : Position data
1 : Work Coordinate System Offset Data*4
Variable No. n+5 : Reserved (to be fixed to 0)
Variable No. n+6 : Reserved (to be fixed to 0)
Variable No. n+7 : Reserved (to be fixed to 0)

*1 Select from the range of integer variables in the local and global domains.
Local domain : 1 to 91, 1001 to 1092
Grobal domain : 200 to 292, 1200 to 1292

*2 Select from the range of integer variables in the local and global domains.
Local domain : 1 to 87, 1001 to 1088
Grobal domain : 200 to 288, 1200 to 1288
(Note) For XSEL-P/Q and some others, there are global domains 20000 to 2788.
[Refer to 4.1 Each Type of Data Available to Handle on the Program and its Range]

*3 The variable set to this data number should be as shown below in response to the

indication of Variable No. n+4.
Variable No. n+4 = 0: Top position No.
1st to 12nd･･Center of Work Piece Gravity Posiiton 1 to 12
1: Top work coordinate system No.
No. 1 to 12 …1 to 12 sets of center of work piece gravity offset
(Note) In either case, confirm 12 sets of data from the top data number are kept unused in a row.

*4 Supported only by TTA or MSEL-PC/PG Application V2.00 and later

- Return Code in GTVD Command (Variable 99 (Local Space))
The result in GTVD execution is stored in Variable 99 as a return code.
0 : Completed in normal condition
1 : Work Information Acquirement WAIT Timeout
2 : GTVD Timer cancelled (condition that the waiting status is cancelled by TIMC Command)
3 : Vision System Unset Detection (SLVS Command not executed, etc.))
4 : Work Detection Cancel Status Detection (errors, etc.)

(Note 1) SLVS and GTVD Commands can be executed only on the same program (task).

(Note 2) Receivable communication formats can be switched in I/O Parameter No. 352, Bits 0 to 7.

(Note 3) The system is capable to obtain the work data (coordinates and attributes) of 12 work pieces in 1 shot of image capturing.
Error No. 417 is issued when 13 or more work pieces are detected in 1 shot, and “4: Work Detection Cancel Status Detection (error, etc.)” is set to the return code.

(Note 4) Error No. 416 (Received Message Error) is issued when there is an error in the received message during SLVS Command execution.
Check the communication format selection parameter (I/O Parameter No. 352 or Bits 0 to 7) settings and the output communication format on Vision System side.

(Note 5) There will be no change in the variables for work attribute storage and the position data when the quantity of detected work piece in the received image data is 0.

(Note 6) It is prohibited to capture an image during the movement of the robot if the camera is mounted on the robot.
Make sure to capture an image in the stop condition.
An accurate work data cannot be acquired if a capturing is conducted during the robot movement.

(Note 7) When the position data is indicated for the data classification for captured data storage, the coordinate system definition unit axes position data should be converted into the position on the work coordinate system selected while this command is being executed.

(Note 8) When the work coordinate system offset data is indicated for the data classification for captured data storage, “Error No. B73 Coordinate System Data Change Forbidden during Servo Use Error” will occur if a subjected axis is in operation while this command is being executed.

(Note 9) When the work coordinate system offset data is indicated for the data classification for captured data storage, the result is set to the offset data of X, Y and R-axes of the coordinate system definition unit related to X, Y and R-axes of the vision system in “All Axes Parameter No. 121 Vision System I/F 1 Coordinate Axes Definition”.

## [30] Anti-Vibration Control Related Command

● NTCH (Anti-Vibration Control Parameter Set Select)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | NTCH | Axis pattern | Parameter set number | CC |

| Applicable models | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| XSEL -J/K | XSEL -P/Q/ PCT/ QCT | XSEL -R/S | XSEL -RA/SA | XSEL -JX/KX | XSEL -PX/QX | XSEL -RX/SX/ RXD/ SXD | XSEL -RAX/ SAX/ RAXD/ SAXD | ASEL PSEL SSEL | TT/TTA | MSEL |
| × | ○ | ○ | ○ | × | × | ○ (Linear drive axis only) | ○ (Linear drive axis only) | × | × | × |

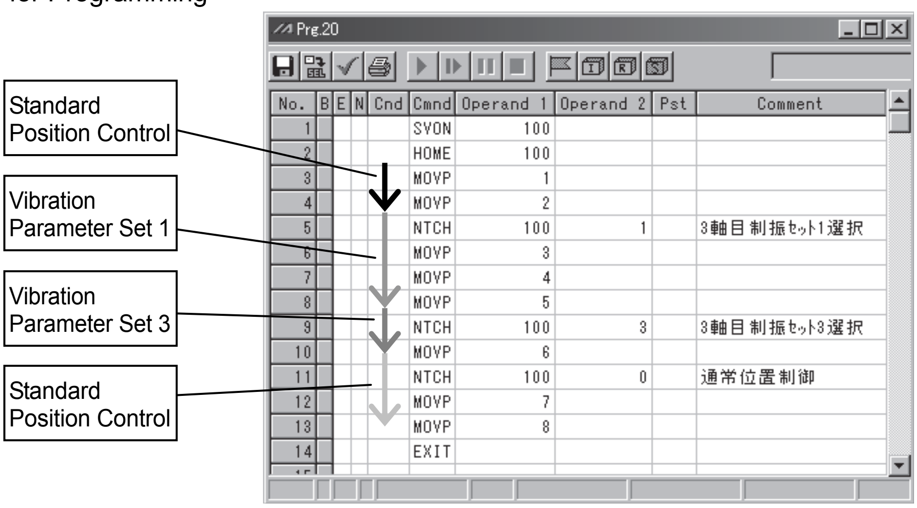
[Function] It declares what, in the specific frequency patterns registered to the parameters, is to be used for the anti-vibration control for the axis pattern set in Operand 1.

Operand 1 : Axis pattern selection
Indicate the axis that the anti-vibration control parameter set selection is conducted as “1” and the one not to be conducted as “0”.

Operand 2 : Parameter set number
Select if use/not to use the anti-vibration control and which specific frequency pattern (parameter set) is to be used.
0 : Standard Position Control (Anti-vibration control is not done)
1 : Vibration Control Parameter Set 1 (Each Axis Parameter No.151 to 154)
2 : Vibration Control Parameter Set 2 (Each Axis Parameter No.156 to 159)
3 : Vibration Control Parameter Set 3 (Each Axis Parameter No.161 to 164)
Except for 0 to 3 : Standard Position Control (Anti-vibration control is not done)

[Example] NTCH 110 2 Setting the anti-vibration control parameter set 2 to the 2nd and 3rd axes

<Example for Programming >



## [31] Compliance Control Related Commands

● COMP (Dedicated SCARA command- Compliance Mode Setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | COMP | Mode Type | (Axis pattern) | CP |

| Applicable models |
|---|
| XSEL-RAX/SAX/RAXD/SAXD(Main application V1.14 or later) |

[Function] The compliance mode (compliance control) specified in Operand 1 should be set to an axis applicable to the axis pattern specified in Operand 2. X and Y-axes should be specified in the base coordinate system for the specification of the axis.
By setting the compliance mode valid, the robot can be controlled softly to the indicated axis direction.
When the compliance mode indicated invalid, the axis pattern in Operand 2 can be skipped. When the axis pattern gets skipped, the compliance mode will get invalid to all the SCARA Robot units.

• Operand 1 Indicated Mode Type

| Setting value | Description |
|---|---|
| 0 | Compliance Mode Invalid<br>* It will be set invalid on all the axes on the SCARA Robot specified in the axis pattern in Operand 2. |
| 1 | Compliance Mode Valid<br>(Base coordinate system compliance mode) |

(Note 1) This command should be available to use only to the axes on IXA SCARA Robot. Error No. C7A “Servo Unsupported Feature Error” will be issued when COMP1 Command is executed to SCARA Robot which does not support the compliance control feature.

(Note 2) It is necessary that X-axis and Y-axis are specified together as a set for the compliance mode. When axes are specified individually, Error no. C30 “Axis Pattern Error” will generate.

(Note 3) Z-axis and R-axis on SCARA Robot will not be able to activate the compliance mode. Error No. B80 “Indication Forbidden Axis Error” will be issued when COMP1 Command is executed to the Z-axis or R-axis on SCARA Robot.

(Note 4) The compliance mode needs to be used in the tool coordinate system No. 0. Error No. B71 “Coordinate System Number Error” will be issued when COMP1 Command is executed in the coordinate system other than the tool coordinate system No. 0 which is currently selected.

(Note 5) The compliance mode cannot be set to the following axes.
- Under Operation
- During Servo being OFF
- Collision Detection Feature in Use
- During Conveyor Tracking Operation (TRAC1 Command being executed)

(Note 6) The compliance mode set with COMP Command remains valid even after the program is finished. When using the compliance mode, make sure to clarify if you would like to activate or inactivate the compliance mode before starting the operation in order to avoid operation in an unexpected mode.

(Note 7) The compliance mode can be inactivated with COMP0 Command or will become invalid when the servo gets turned off, when in emergency stop, when an axis error has been occurred and so on.

(Note 8) When an error gets occurred on an axis which is in the compliance mode (compliance control), Error No. 4C7 “Compliance Control Axis Error Stop Detected” should occur on other SCARA Robot axes and the operation will stop.

(Note 9) It is not available to have an operation command executed on an axis on which the compliance mode is activated. When the compliance mode is valid on the X and Y-axes, no operation command can be executed on the X, Y and R-axes. Having an operation command executed on axes on which the compliance mode is activated should cause Error No. CC0 “Axis Mode Error”.

(Note 10) The compliance mode cannot be activated near the singularity of SCARA Robot. Having COMP1 Command executed near the singularity should cause Error No. B74 “CP Operation Limit Band Violation Error”.

| | | | | |
|---|---|---|---|---|
| [Example 1] | COMP | 1 | 11 | Make the compliance mode on the X-axis and Y-axis valid |
| [Example 2] | COMP | 0 | | Make the compliance mode on all the SCARA Robot axes invalid |
| [Example 3] | COMP | 1 | 110000 | Make the compliance mode on the X-axis and Y-axis valid |

● SCLO0 (Dedicated SCARA command-Compliance Mode Option Feature Setting (Searching Operation Setting))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SCLO | 0 | Integer variable number | CP |

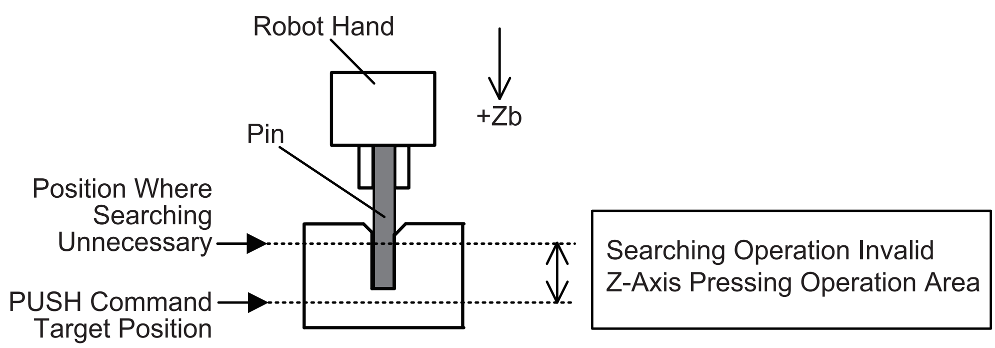
| Applicable models |
|---|
| XSEL-RAX/SAX/RAXD/SAXD(Main application V1.14 or later) |

[Function] The searching operation setting in the compliance mode (compliance control) can be established.
The data stored in the variable numbers (2 numbers in a row) indicated in Operand 2 should be set as the searching operation conditions.

• Details of Integer Variables Indicated in Operand 2

| Variable No. | Description |
|---|---|
| n | Searching Operation Valid Axis Patterns<br>Searching operation should be performed on each joint axis unit that was indicated.<br>The 1st axis or the 2nd axis (5th or 6th axis for the 2nd unit) should be available for indication. |
| n+1 | Searching Operation Invalid Z-Axis Pressing Operation Area [0.01mm unit]<br>Distance in front of the target position for Z-axis pressing operation (PUSH Command) should be indicated. The searching operation will get inactivated when the actuator reaches the indicated distance area.<br>This should be invalid when 0 or any number smaller than 0 is indicated. |

Searching Operation Invalid Z-Axis Pressing Operation Area is a feature that makes the searching operation invalid in the indicated area.
In such a case as to insert a pin to +Zb direction as shown in the figure below, after the pin gets inserted for a certain amount, it is not necessary to have the searching operation after that. In such cases, have an appropriate setting established for Searching Operation Invalid Z-Axis Pressing Operation Area.



(Note 1) Input an integer variable number in Operand 2.
It should be available to indicate from;
- Local area : 1 to 98, 1001 to 1098,
- Global area : 200 to 298, 1200 to 1298, 2000 to 2798

(Note 2) This command should be available to use only to the axes on IXA SCARA Robot. Executing this command to SCARA Robot which is not applicable for the compliance control feature will not cause error but the command will be ignored.

(Note 3) SCARA Robot subject to the compliance mode option change should be indicated with GRP Command. Indicate all the axes on SCARA Robot that you would like to change with GRP Command. Error No. C30 "Axis Pattern Error" will occur if there is no axis indicated.

(Note 4) SCLO Command cannot be executed during the compliance mode (compliance control). Error No. B47 "Compliance Mode Operation Error" will occur if SCLO Command is executed during the compliance control.

(Note 5) At the controller startup, All Axes Common Parameter No. 154/242 "Compliance Searching Operation Valid Axis Pattern Initial Values" and No. 155/243 "Compliance Searching Operation Invalid Z-Axis Pressing Operation Area Initial Values" should be the default.

(Note 6) After the compliance mode option has been set with SCLO Command, it will be remained until the software reset is conducted or the power is turned OFF.

| | | | | |
|---|---|---|---|---|
| [Example 1] | LET | 21 | &B0001 | Indicate J1-axis for searching operation valid axis pattern |
| | LET | 22 | 100000 | Indicate 100.000mm to Compliance Searching Operation Invalid Z-Axis Pressing Operation Area |
| | GRP | 1111 | | Set the axis pattern 1111b |
| | SCLO | 0 | 21 | Set the compliance searching conditions axes from 1st to 4th |
| [Example 2] | LET | 21 | &B10000 | Indicate J1-axis for searching operation valid axis pattern |
| | LET | 22 | 100000 | Indicate 100.000mm to Compliance Searching Operation Invalid Z-Axis Pressing Operation Area |
| | GRP | 11110000 | | Set the axis pattern 11110000b |
| | SCLO | 0 | 21 | Set the compliance searching conditions axes from 5th to 8th |

● SCLO1 (Dedicated SCARA command-Compliance Mode Option Feature Setting (J1 & J2-Axes Torque Limit Mode Setting))

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SCLO | 1 | Integer variable number | CP |

| Applicable models |
|---|
| XSEL-RAX/SAX/RAXD/SAXD(Main application V1.14 or later) |

[Function] The setting for J1 & J2-Axes Torque Limit Mode in the compliance mode (compliance control) can be established.
The type of torque limit mode valid/invalid stored in the variable numbers indicated in Operand 2 should be set.

• Details of Integer Variables Indicated in Operand 2

| Variable No. | Description |
|---|---|
| n | J1 & J2-Axes Torque Limit Mode Valid Setting<br>0: Torque limit invalid<br>1: Torque limit valid<br><br>* Use the product with the J1 and J2-axes torque limit set valid in usual condition. |

(Note 1) Input an integer variable number in Operand 2.
It should be available to indicate from;
- Local area : 1 to 99, 1001 to 1099,
- Global area : 200 to 299, 1200 to 1299, 2000 to 2799

(Note 2) This command should be available to use only to the axes on IXA SCARA Robot. Executing this command to SCARA Robot which is not applicable for the compliance control feature will not cause error but the command will be ignored.

(Note 3) SCARA Robot subject to the compliance mode option change should be indicated with GRP Command. Indicate all the axes on SCARA Robot that you would like to change with GRP Command. Error No. C30 “Axis Pattern Error” will occur if there is no axis indicated.

(Note 4) SCLO Command cannot be executed during the compliance mode (compliance control). Error No. B47 “Compliance Mode Operation Error” will occur if SCLO Command is executed during the compliance control.

(Note 5) At the controller startup, bits from 20 to 23 in All Axes Common Parameter No. 51/228 “SCARA Axis Control 1” should be the default.

(Note 6) After the compliance mode option has been set with SCLO Command, it will be remained until the software reset is conducted or the power is turned OFF.

| | | | | |
|---|---|---|---|---|
| [Example 1] | LET | 21 | 1 | Indicate J1 & J2-Axes Torque Limit Mode valid |
| | GRP | 1111 | | Set the axis pattern 1111b |
| | SCLO | 1 | 21 | Set the compliance torque limit mode axes from 1st to 4th |
| [Example 2] | LET | 21 | 1 | Indicate J1 & J2-Axes Torque Limit Mode valid |
| | GRP | 11110000 | | Set the axis pattern 1111b |
| | SCLO | 1 | 21 | Set the compliance torque limit mode axes from 5th to 8th |

● SCLG (Dedicated SCARA command-Compliance Gain Setting)

| Extension condition (LD, A, O, AB, OB) | Input condition (I/O, flag) | Command, declaration | | | Output (Output, flag) |
|---|---|---|---|---|---|
| | | Command, declaration | Operand 1 | Operand 2 | |
| E | N, Cnd | Cmnd | Operand 1 | Operand 2 | Pst |
| Optional | Optional | SCLG | Integer variable number | Prohibited | CP |

| Applicable models |
|---|
| XSEL-RAX/SAX/RAXD/SAXD(Main application V1.14 or later) |



[Function] The data stored in the variable numbers (6 numbers in a row) indicated in Operand 1 should be set as the compliance gain. The compliance gain should be set in the range from 0 to 10000. With 0 being the softest, it gets harder as the number goes up.

• Details of Integer Variables Indicated in Operand 1

| Variable No. | Description |
|---|---|
| n | X-Axis Gain |
| n+1 | Y-Axis Gain |
| n+2 | System Reservation (Set to 0) |
| n+3 | System Reservation (Set to 0) |
| n+4 | System Reservation (Set to 0) |
| n+5 | System Reservation (Set to 0) |

(Note 1) Input an integer variable number in Operand 1.
It should be available to indicate from;
- Local area : 1 to 94, 1001 to 1094,
- Global area : 200 to 294, 1200 to 1294, 2000 to 2794

(Note 2) This command should be available to use only to the axes on IXA SCARA Robot. Executing this command to SCARA Robot which is not applicable for the compliance control feature will not cause error but the command will be ignored.

(Note 3) SCARA Robot subject to the compliance gain option change should be indicated with GRP Command. Indicate all the axes on SCARA Robot that you would like to change with GRP Command. Error No. C30 “Axis Pattern Error” will occur if there is no axis indicated.

(Note 4) At the controller startup, Axis-Specific Parameter No. 202 “Compliance Gain Viscous Rate Initial Value” should be the default.

(Note 5) After the compliance gain has been set with SCLG Command, it will be remained until the software reset is conducted or the power is turned OFF.

[Example 1]

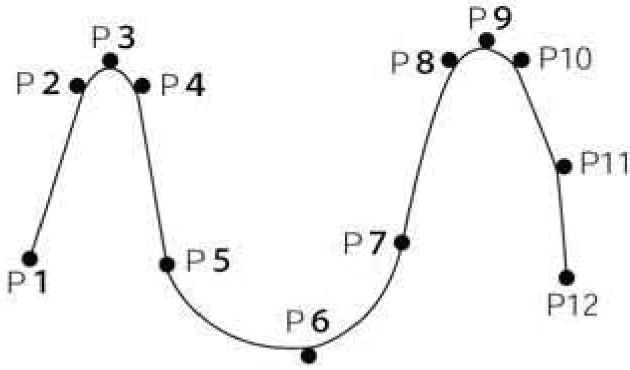
```
LET   21    100    X-axis gain set to 100
LET   22    200    Y-axis gain set to 200
LET   23    0      Set to 0
LET   24    0      Set to 0
LET   25    0      Set to 0
LET   26    0      Set to 0
GRP   1111         Set the axis pattern 1111b
SCLG  21           Set compliance gain using six numbers of integer
                   variables in a row from Variable No. 21
```

## 5.4 Key Characteristics of Actuator Control Commands and Points to Note

### 5.4.1 Continuous Movement Commands
### [PATH, CIR, ARC, PSPL, CIR2, ARC2, ARCD, ARCC, CIRS, ARCS and CNTP]

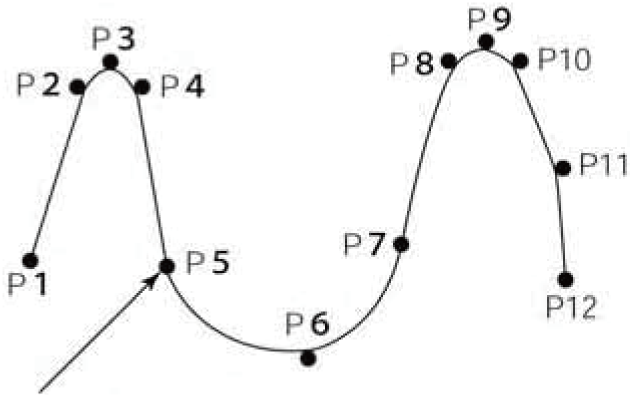
1) By running a program with continuous movement commands input in a series of continuous program
   steps, you can allow the actuators to perform operations continuously without stopping between steps.

| | | | |
|---|---|---|---|
| | PATH | 1 | 5 |
| | ARC2 | 6 | 7 |
| | PATH | 8 | 12 |



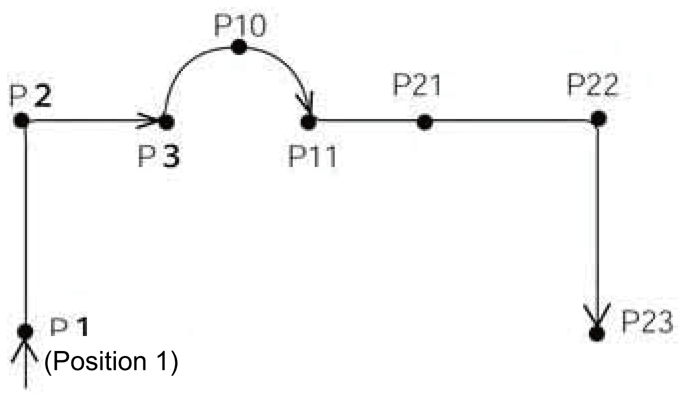

2) Continuous movement will not be achieved if an input condition is specified for any continuous movement command.

| | | | |
|---|---|---|---|
| | PATH | 1 | 5 |
| 20 | ARC2 | 6 | 7 |
| | PATH | 8 | 12 |




Stops momentarily.

3) The output field of each command will turn ON as the end position of that command approaches.
   Only with the last command in a series of continuous movement commands, the output will turn ON upon completion of operation (if there is no input condition).



[Example 1] (POTP = 1)

```
        POTP    1
         :

         :
        PATH    1     3     308
        ARC2    10    11    311
        PATH    21    23    312
```

| Output | Timing |
|---|---|
| 308 | Turn ON as P1 approaches. |
| 309 | Turn ON as P2 approaches. |
| 310 | Turn ON as P3 approaches. |
| 311 | Turn ON as P11 approaches. |
| 312 | Turn ON as P21 approaches. |
| 313 | Turn ON as P22 approaches. |
| 314 | Turn ON when P23 operation is complete. |

[Example 2] (POTP = 0)

```
        PATH    1     3     308
        ARC2    10    11    311
        PATH    21    23    312
```

| Output | Timing |
|---|---|
| 308 | Turn ON as P3 approaches. |
| 311 | Turn ON as P11 approaches. |
| 312 | Turn ON when P23 operation is complete. |

[Example 3] If an input condition is specified, the output will turn ON upon completion of operation inthe step before the one in which the input condition is specified.

```
        POTP    1
         :

         :
        PATH    1     3     308
   20   ARC2    10    11    311
        PATH    21    23    312
```

| Output | Timing |
|---|---|
| 308 | Turn ON as P1 approaches. |
| 309 | Turn ON as P2 approaches. |
| 310 | Turn ON when P3 operation is complete. |
| 311 | Turn ON as P11 approaches. |
| 312 | Turn ON as P21 approaches. |
| 313 | Turn ON as P22 approaches. |
| 314 | Turn ON when P23 operation is complete. |

4) When executing continuous movement commands sequentially, the controller is calculating approx.
100 positions ahead. This is why the steps are displayed continuously on the PC screen or teachingpendant screen, regardless of the actual operation. The last step in the continuous operation section executed by continuous movement commands will wait for the applicable operation to complete.

```
         :

         :                         ←  Actuator operation
        PATH    1     5     308
        ARC     6     7     311    ←  Step displayed on the PC software or
        PATH    8     12    312       teaching pendant
        BTON    310
         :

         :
```

5) Do not allow the output fields to duplicate in the continuous operation section executed by continuous movement commands.
Duplicating output fields in the continuous operation section will not achieve the expected result.
The output field will turn OFF at the start of processing of each command.

```
POTP    1

PATH    1     5     305
 :
 :
PATH    11    15    308
```

Do not let outputs 305 through 308 to duplicate, as in the example shown at left.

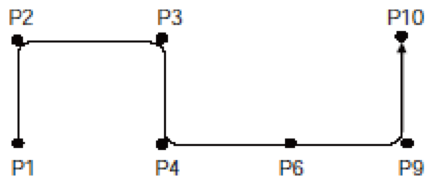
Continuous operation section executed by continuous movement commands

The final output status of duplicate 305 through 308 is indeterminable, because it is affected by the positioning calculation time and the relationship of durations of actual operations.

6) The actuator can be moved continuously along a series of continuous positions including one discontinuous position. Specify the position number corresponding to the discontinuous position as both the start position number and end position number of a PATH command. Position No. 6 is the discontinuous point to be passed in this example.

The actuator will move continuously along the path of position Nos. 1→2→3→4→6→9→10.

```
PATH    1     4
PATH    6     6     Discontinuous position
PATH    9     10
```

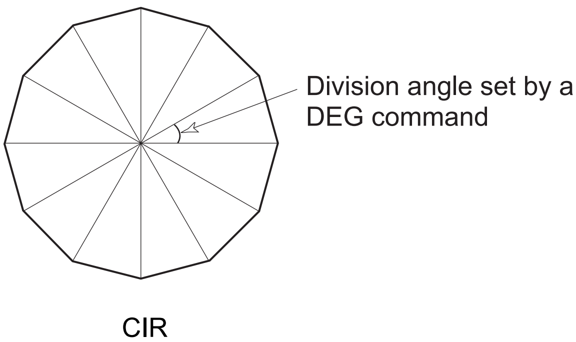


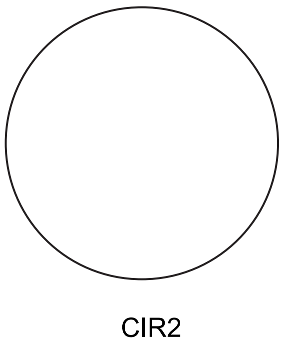
### 5.4.2 PATH/PSPL Commands

When executing a PATH or PSPL command, pay attention to the locus because it will change if the acceleration/deceleration is different between points.
The locus can be fine-tuned by changing the acceleration/deceleration, but different acceleration/deceleration settings between points will prevent smooth transition of speeds when moving from one position to another.
If there is a large difference in deceleration/acceleration between points and the positioning distance is small, the speed may drop. Exercise caution.

### 5.4.3 CIR/ARC Commands

The processing by a CIR or ARC command resembles moving along a polygon with a PATH command.
A small division angle may cause the speed to drop.
CIR2, ARC2, ARCD and ARCC commands actually perform arc interpolation.




### 5.4.4 CIR2/ARC2/ARCD/ARCC Commands

With a CIR2, ARC2, ARCD or ARCC command, the speed can be changed (only in the arc interpolation section) by inputting a speed for the point specified in operand 1. These commands are effective when you must lower the speed partially because the radius is small and the arc locus cannot be maintained inside the allowable range.

The speed and acceleration will take valid values based on the following priorities:

| Priority | Speed | Acceleration (deceleration) |
|---|---|---|
| 1 | Setting in the position data specified in operand 1 | Setting in the position data specified in operand 1 |
| 2 | Setting by VEL command | Setting by ACC (DCL) command |
| 3 | | Default acceleration in all-axis parameter No. 11<br>Default deceleration in all-axis parameter No. 12 |

## 5.5 Position Output Operation Features

Applicable Controllers : TTA, MSEL, XSEL-RA/SA/RAX/SAX/RAXD/SAXD

Applicable Versions : TTA application Part V2.00 and later
MSEL application Part V2.00 and later

: TTA, MSEL • PC software (IA-101-X-MW) V12.03.00.00 and later
• Teaching pendant (TB-01) V1.50 and later
• Teaching pendant (TB-02) first edition and later
: It is applicable from the initial version for XSEL-RA/SA/RAX/SAX/RAXD/SAXD

* XSEL-RA/SA/RAX/SAX/RAXD/SAXD controllers are already applicable for this feature at the delivery.

### 5.5.1 Outline

There was POTP Command (PATH Output Type Setting) in SEL Command as the feature to output with a position in operation by a command such as PATH Command. This is a feature to output the on-signal while incrementing the output port indicated at the output part every time the actuator approaches each position in continuous movement during PATH and PSPL movements. Therefore, it always conducts on operation to different outputs.
On the other hand, the position output operation feature enables on and off operations to the same outputs (control outputs of external devices) as it controls external devices such as a dispenser and torque driver during the axis operation.
The output operation setting should be conducted at each position in the position data.
The set output operation is conducted when the movement command of the position that the output operation is set is executed.
There are six types output operation as;

- To turn the indicated output port or flag ON after movement
- To turn the indicated output port or flag OFF after movement
- To turn the indicated output port or flag ON after passed the indicated distance
- To turn the indicated output port or flag OFF after passed the indicated distance
- To turn the indicated output port or flag ON at the position where the actuator is moved for the indicated ratio minute to the whole movement amount from the movement start position
- To turn the indicated output port or flag OFF at the position where the actuator is moved for the indicated ratio minute to the whole movement amount from the movement start position.

For TTA and MSEL this feature is not activated at delivery. Activate it by conducting 5.5.2 How to Valid Position Output Operation Function.
This feature is already valid at delivery for XSEL-RA/SA/RAX/SAX/RAXD/SAXD. It is not available to make it invalid.

### 5.5.2 How to Valid Position Output Operation Function (TTA, MSEL)

If the output operation data valid/invalid setting is changed, it is necessary to initialize the position data as a change is made to the position data construction. Follow the procedures below to switch the output operation data valid/invalid.

1) Save the position data to a file if necessary.
   * Make sure to have a backup if the current data is necessary when switching the position output operation data valid/invalid as initialization of the position data is conducted.

2) Change Bit 4-7: Position Output Operation Data Valid Select in All Axes Parameter No. 54 “All Axes Setting Bit Pattern 3” (0: Invalid, 1: Valid)

3) Write the parameters to the flash ROM, and conduct the software reset.
   * Error No. 6BD “Position Data Construction Change Error” will occur after rebooting.

4) Initialize the position data and clear up all the position data.
   Conduct “Controller” → “Memory Initialization” → “Position” in the main menu.

5) Write the position data to the flash ROM, and conduct the software reset.

6) As all the position data are cleared, transfer the position data saved in 1) as necessary to the controller and write in the flash ROM.

The procedures above are necessary also when the setting of valid/invalid for the position output operation data is changed by transferring and writing the parameter files. It is not necessary to have the procedures above when all the backup files are to be transferred to the controller as the tool automatically initializes the position data.

### 5.5.3 How to Valid Position Data Output Operation Setting

There are 4 items, OutFn, OutNo., OutPara1 and OutPara2, as the items to set the output operation.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl | OutFn | OutNo. | OutPara1 | OutPara2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10( ) | 100.000 | | | | 100 | 0.30 | 0.30 | ON | 316 | 0.000 | 0.000 |
| 11( ) | | 150.000 | | | 100 | 0.30 | 0.30 | | | | |
| 12( ) | 150.000 | | | | 100 | 0.30 | 0.30 | | | | |
| 13( ) | | 100.000 | | | 100 | 0.30 | 0.30 | OFF | 316 | 0.000 | 0.000 |

<u>OutFn</u>

Set the output functio.
Set a value applicable for the output function of 1 to 6.
The output operation will be invalid if no setting is conducted.
The display is as shown in the table below, and if 1 is set for instance, the display is on.

| Output Function Name | Setting in Number | Display |
|---|---|---|
| Turns on after Moving | 1 | ON |
| Turns off after Moving | 2 | OFF |
| Turns on after passing indicated distance | 3 | OND |
| Turns off after passing indicated distance | 4 | OFFD |
| Turns on after passing indicated ratio | 5 | ONR |
| Turns off after passing indicated ratio | 6 | OFFR |

<u>OutNo.</u>

Set the output port and flag subject to operation.

<u>OutPara1</u>

Set the parameters defined for each output function.

<u>OutPara2</u>

If the output pulse timer time is set, one-shot pulse output is made.

Setting for Each Output Function

<table>
<tr><th>Output Function Name</th><th>OutPara1 :<br>Function Parameter 1</th><th>OutPara2 :<br>Function Parameter 2</th></tr>
<tr><td>Turns on after Moving</td><td rowspan="2">Output delay timer time<br>(0.000 to 999.999 sec)</td><td rowspan="2">Output pulse timer time<br>(0.000 to 999.999 sec)</td></tr>
<tr><td>Turns off after Moving</td></tr>
<tr><td>Turns on after passing indicated distance</td><td rowspan="2">Output operation indicated distance<br>(0.000 to 9999.999 mm)</td><td rowspan="2">Output pulse timer time<br>(0.000 to 999.999 sec)</td></tr>
<tr><td>Turns off after passing indicated distance</td></tr>
<tr><td>Turns on after passing indicated ratio</td><td rowspan="2">Output operation indicated rate<br>(0.000 to 100.000 %)</td><td rowspan="2">Output pulse timer time<br>(0.000 to 999.999 sec)</td></tr>
<tr><td>Turns off after passing indicated ratio</td></tr>
</table>

## 5.5.4 Valid SEL Language Commands for Output Operation of Position Data and Common Notes for Caution

The commands shown in the table below are the valid commands.

| Output Function Code | PC / TP Display | Output Function Name | Valid SEL Commands |
|---|---|---|---|
| 0 | - | Invalid | — |
| 1 | ON | Turns on after Moving | MOVP / MOVL / MVPI / MVLI / TMPI / TMLI / PATH / PSPL / PUSH / CIR / ARC / CIR2 / ARC2 / CIRS / ARCS / ARCD / ARCC / ARCH |
| 2 | OFF | Turns off after Moving | |
| 3 | OND | Turns on after passing indicated distance | MOVL / MVLI / TMLI / PATH / CIR2 / ARC2 / ARCD / ARCC / ARCH |
| 4 | OFFD | Turns off after passing indicated distance | |
| 5 | ONR | Turns on after passing indicated ratio | |
| 6 | OFFR | Turns off after passing indicated ratio | |

### 5.5.5 Explanation of Each Output Function

Below explains each output function.

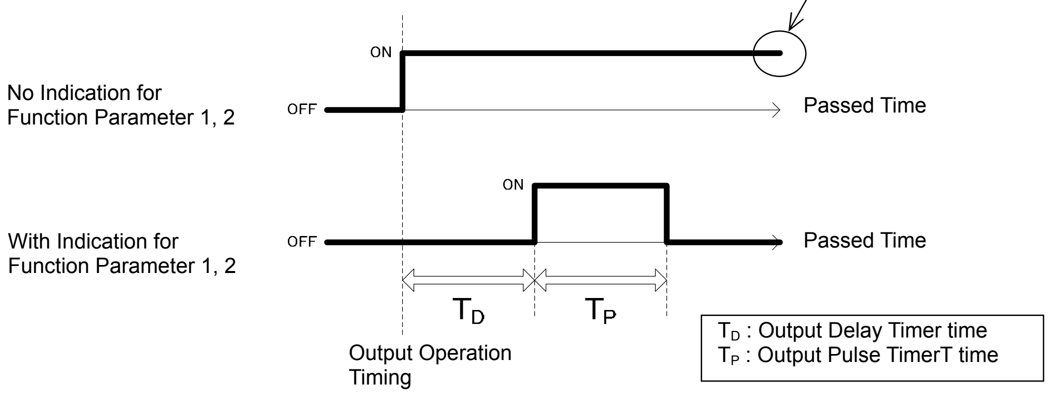
[1] Turns ON/OFF After Moving

| Output Function Code | PC / TP Display | Output Function Name | Operation Output Port Flag | Function Parameter 1 | Function Parameter 2 |
|---|---|---|---|---|---|
| 1 | ON | Turns on after Moving | Output Port Flag No. | Output delay timer time | Output pulse timer time |
| 2 | OFF | Turns off after Moving | | | |

[Function] The indicated output ports and flags are turned on/off after moving to the applicable position. The timing of the output operation is as stated in the table below for each classification of the movement. Indicate each of the output delay timer time and output pulse timer time of the function parameter, and one-shot pulse is output. (No indication when 0)

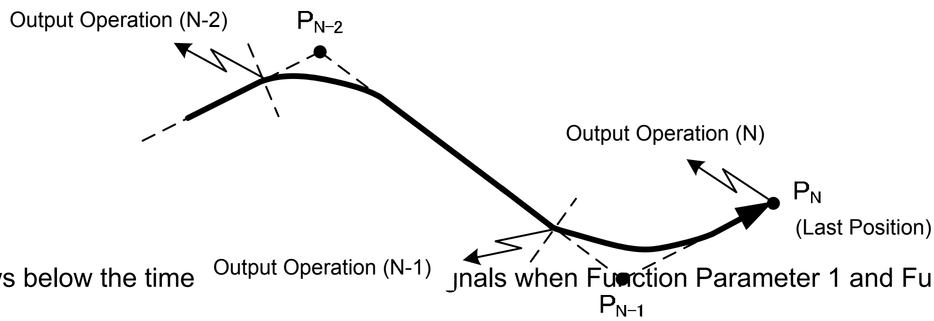
If it is set to be turned off after movement in the position movement to another point, the signal is kept on. If it is required to be compulsorily turned off, turn it off with BTOF Command after movement is completed.

(Example) Output Time Chart of Turning on after Movement




■ Output Operation Timing

| Movement Classification | Output Operation Timing |
|---|---|
| Non-Continuous Movement (MOVP, MOVL, etc.) | At movement complete (positioning complete) |
| Continuous Movement (PATH, CIR2, ARC2, etc.) | At target position approach except for last position and at operation complete (positioning complete) for last position |
| Pressing Movement (PUSH) | At detection of pressing complete |

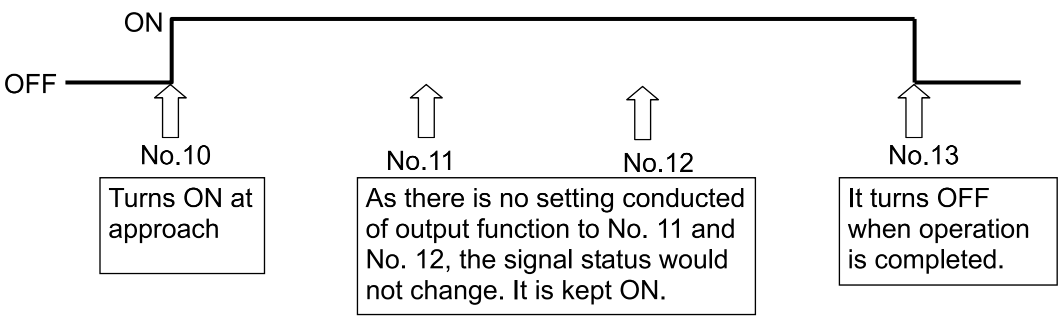



It shows below the time ... gnals when Function Parameter 1 and Function

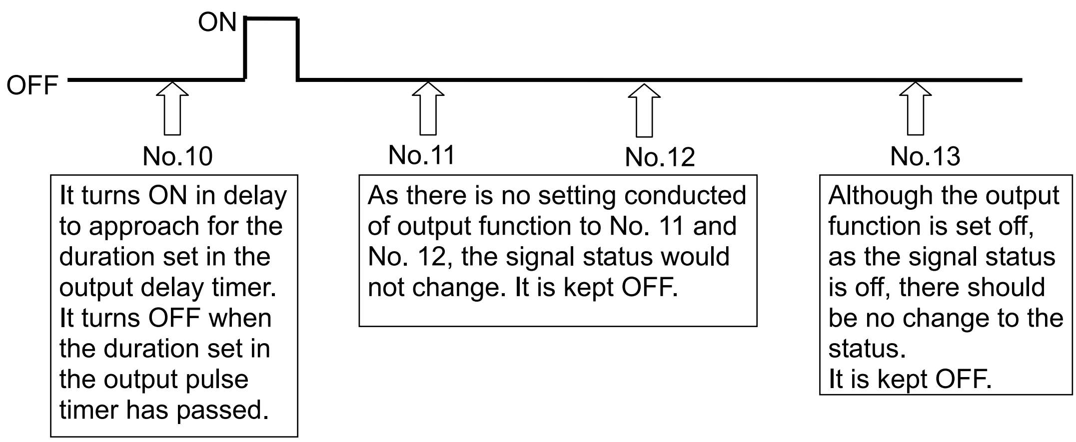
Parameter 2 are indicated and not indicated in the case that PATH Operation in Position No. 10 to No. 13 below is conducted as an example.

| No.(Name) | Axis1 | Axis2 | Axis3 | Axis4 | Vel | Acc | Dcl | OutFn | OutNo. | OutPara1 | OutPara2 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10( ) | 100.000 | | | | 100 | 0.30 | 0.30 | ON | 316 | | |
| 11( ) | | 150.000 | | | 100 | 0.30 | 0.30 | | | | |
| 12( ) | 150.000 | | | | 100 | 0.30 | 0.30 | | | | |
| 13( ) | | 100.000 | | | 100 | 0.30 | 0.30 | OFF | 316 | | |

◎ When Function Parameter 1 and Function Parameter 2 are not indicated



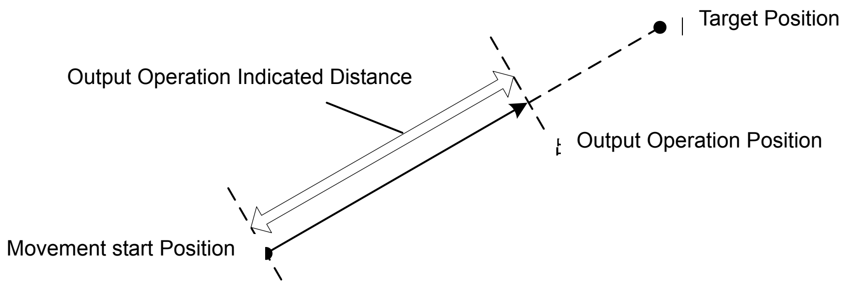

◎ When Function Parameter 1 and Function Parameter 2 are indicated
- Function Parameter 1 Setting Item : Output Delay Timer
- Function Parameter 2 Setting Item : Output Pulse Timer



[2] Turns ON/OFF After Passing Indicated Distance

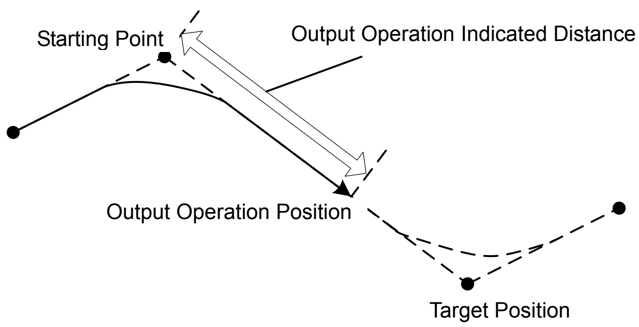
| Output Function Code | PC / TP Display | Output Function Name | Operation Output Port Flag | Function Parameter 1 | Function Parameter 2 |
|---|---|---|---|---|---|
| 3 | OND | Turns on after passing indicated distance | Output Port Flag No. | Output Operation indicated distance | Output pulse timer time |
| 4 | OFFD | Turns off after passing indicated distance | | | |

[Function] The indicated output ports and flags are turned on/off at the position forwarded for the distance from the movement start position to the indicated in Function Parameter 1 during movement to the applicable position. Indicate the output pulse timer times of Function Parameter 2, and one-shot pulse is output. (No indication when 0)
This output function is valid for MOVL, MVLI, TMLI, PATH, CIR2, ARC2, ARCD, ARCC and ARCH Commends. Error No. B97 “Position Output Operation Data Indication Error” will occur if indicated to other commands.
* Even though the output timing of signal ON and OFF differs from that of ON/OFF after movement, the way of ON-OFF operation is the same. Refer to [1] ON /OFF after Movement for the way of ON /OFF operation.

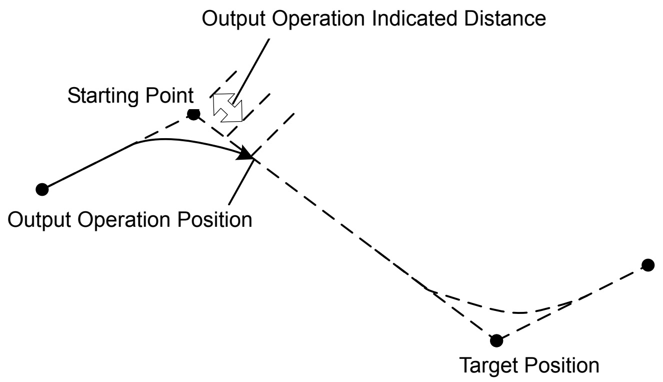



(Note 1) This function cannot be indicated when the tool coordinate system offset is set for X-axis or Y-axis in the movement of 3-Axes Type SCARA. (Error No. B97 “Position Output Operation Data Indication Error”)

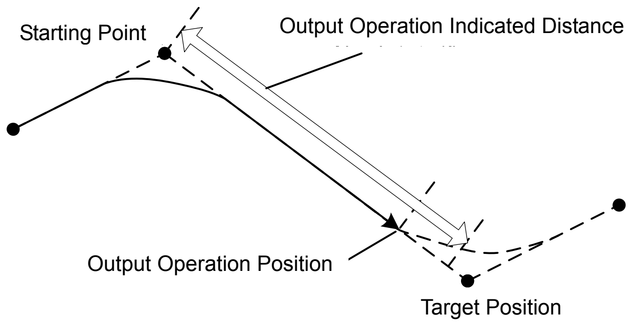
(Note 2) For a position in the continuous movement such as PATH and arc movement, output operation position will be indicated with the distance with the position in front as the starting point.



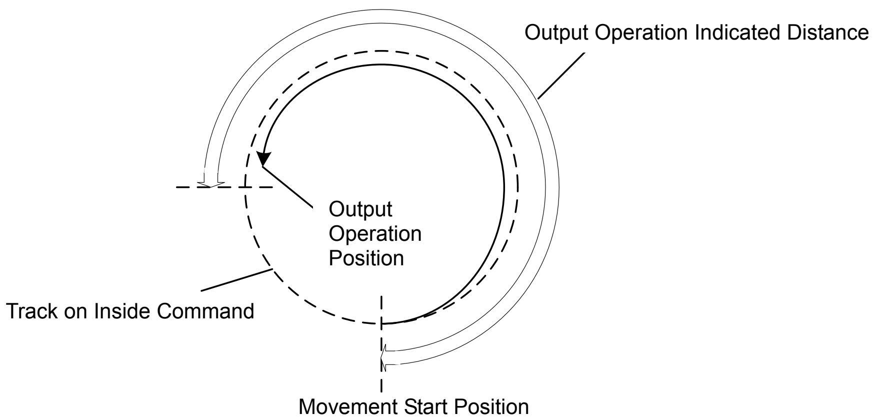
If the output operation indication distance is indicating the duplicating point of the movement to the position in front and the movement to the applicable position (as shown in the figure below), the output operation position will be the position that the movement to the position in front completes.



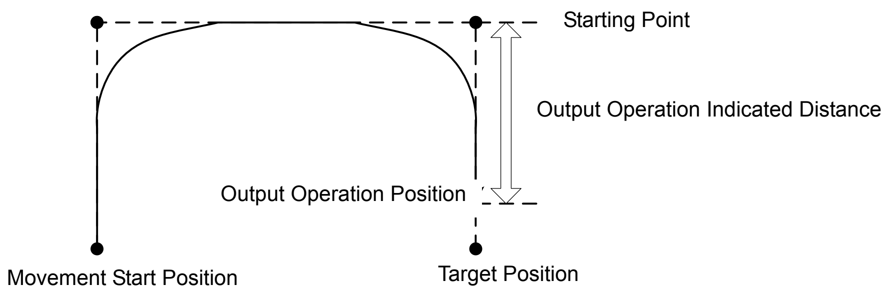

Also, if the output operation indication distance is indicating the duplicating point of the movement to the position in front and the movement to the next position (as shown in the figure below), the output operation position will be the position that the movement to the next position starts.




(Note 3) In the arc movement, the output operation position will be indicated in a distance to the arc track on the inside command. As the actual track may become inside for some velocity settings, adjustment of the indicated distance may be necessary.



(Note 4) In the arch motion movement by ARCH Command, output operation position will be indicated with the distance with the arch motion Z-axis downward movement position as the starting point. Just as PATH, if the output operation indication distance is indicating the duplicating point of the horizontal movement and the Z-axis downward movement, the output operation position will be the position that the horizontal movement completes.




(Note 5) Variance in the output operation timing may become extremely large when an indication is conducted to move in a very small distance to Arm 1, Arm 2 of IXP PowerCon SCARA or an axis with small resolutions compared to other axes. Revise the movement position in such a case.

(Example) When moving PowerCon SCARA Robot from the current position (-50.000, 300.000, 110.000, 0.000) to the target position (-49.900, 299.800, 109.000, 180.000);
→Change the target position to (-, -, -, 180.000) (no indication to 1st to 3rd arms (not to move))

(Note 6) The relation between the distance of the output operation indication and the output position is determined by the movement amount of each axis. When movement is made with the linear movement axes of three axes or less, the output operation indication distance can be a distance in the actual space, but when movement is made with the linear movement axes and rotary movement axes mixed together (for such a reason as to avoid interference), or when four or more linear movement axes are to move, the output operation indication distance cannot be in the actual space as shown in the example below. In such a case, it is easier to establish the setting with using the output function indicating in rate.

Example) The position to conduct the output operation with indication of 150 for output operation indication distance is as shown below when X-axis and Y-axis that move linearly and R-axis that moves rotationally makes the linear interpolation movement from (0, 0, 0) to (100, 200, 90).

$$\text{X-axis : } 150 \times \frac{100}{\sqrt{100^2+200^2+90^2}} = 62.230$$

$$\text{Y-axis : } 150 \times \frac{200}{\sqrt{100^2+200^2+90^2}} = 124.461$$

$$\text{R-axis : } 150 \times \frac{90}{\sqrt{100^2+200^2+90^2}} = 56.007$$

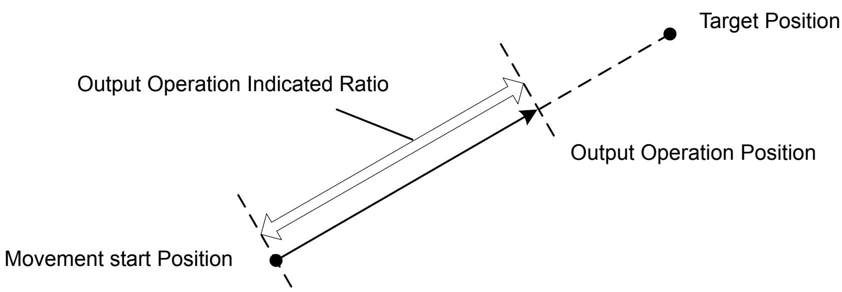
(Note 7) For the tracking operation of XSEL-RA/SA/RAX/SAX/RAXD/SAXD controllers, the output function of indicated distance on/off is not to be used. (Error No. B97 “Position Output Operation Data Indication Error”)

[3] Turns ON/OFF After Passing Indicated Ratio

| Output Function Code | PC / TP Display | Output Function Name | Operation Output Port Flag | Function Parameter 1 | Function Parameter 2 |
|---|---|---|---|---|---|
| 5 | ONR | Turns on after passing indicated ratio | Output Port Flag No. | Output Operation indicated ratio | Output pulse timer time |
| 6 | OFFR | Turns off after passing indicated ratio | | | |

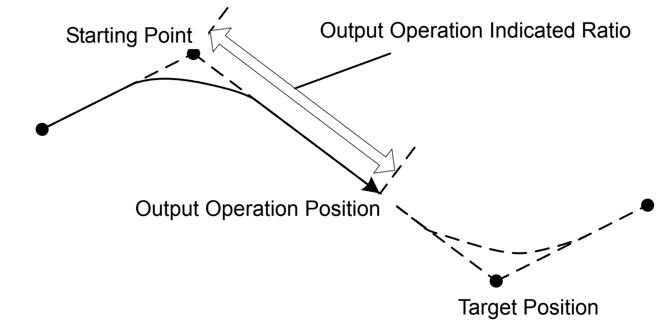
[Function] The indicated output ports and flags are turned on/off at the position forwarded for the distance from the movement start position to the indicated in Function Parameter 1 during movement to the applicable position. Indicate the output pulse timer times of Function Parameter 2, and one-shot pulse is output. (No indication when 0)
This output function is valid for MOVL, MVLI, TMLI, PATH, CIR2, ARC2, ARCD, ARCC and ARCH Commends. Error No. B97 “Position Output Operation Data Indication Error” will occur if indicated to other commands.

* Even though the output timing of signal ON and OFF differs from that of ON/OFF after movement, the way of ON-OFF operation is the same. Refer to [1] ON /OFF after Movement for the way of ON /OFF operation.

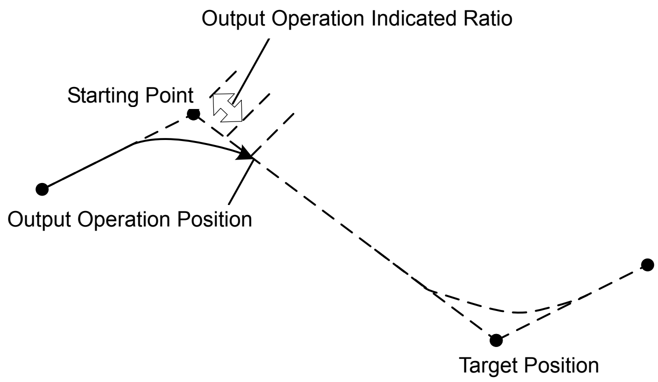



(Note 1) This function cannot be indicated when the tool coordinate system offset is set for X-axis or Y-axis in the movement of 3-Axes Type SCARA. (Error No. B97 “Position Output Operation Data Indication Error”)

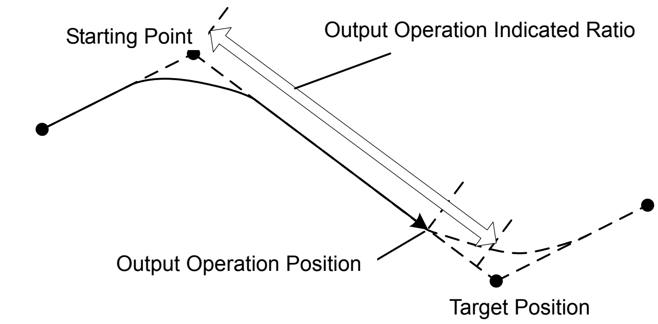
(Note 2) For a position in the continuous movement such as PATH and arc movement, output operation position will be indicated with the ratio with the position in front as the starting point.



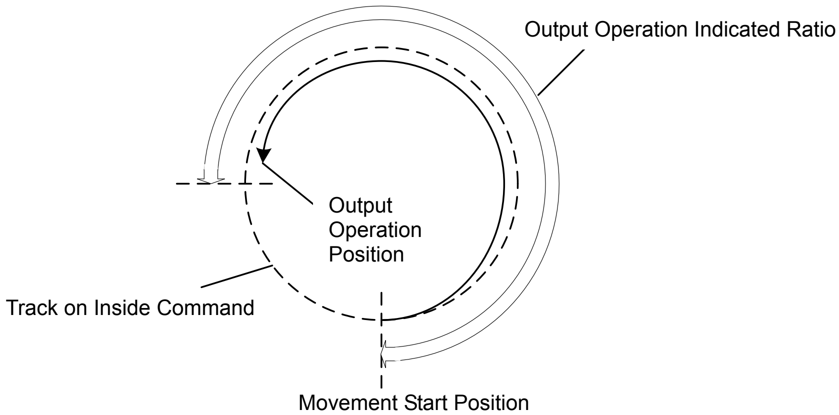

If the output operation indication ratio is indicating the duplicating point of the movement to the position in front and the movement to the applicable position (as shown in the figure below), the output operation position will be the position that the movement to the position in front completes.



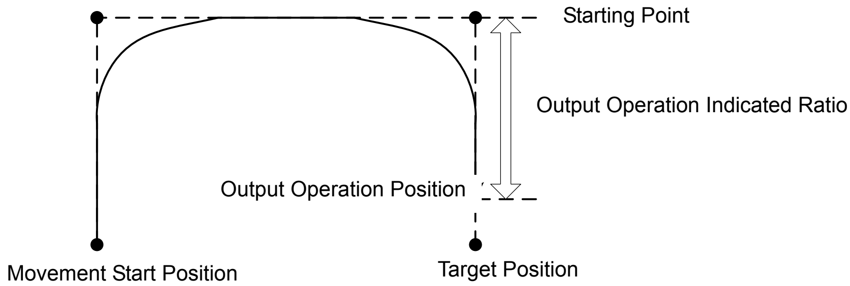

Also, if the output operation indication ratio is indicating the duplicating point of the movement to the position in front and the movement to the next position (as shown in the figure below), the output operation position will be the position that the movement to the next position starts.



(Note 3) In the arc movement, the output operation position will be indicated in a ratio to the arc track on the inside command. As the actual track may become inside for some velocity settings, adjustment of the indicated ratio may be necessary.




(Note 4) In the arch motion movement by ARCH Command, output operation position will be indicated with the distance with the arch motion Z-axis downward movement position as the starting point. Just as PATH, if the output operation indication ratio is indicating the duplicating point of the horizontal movement and the Z-axis downward movement, the output operation position will be the position that the horizontal movement completes.




(Note 5) Variance in the output operation timing may become extremely large when an indication is conducted to move in a very small distance to Arm 1, Arm 2 of IXP PowerCon SCARA or an axis with small resolutions compared to other axes. Revise the movement position in such a case.

(Example) When moving IXP PowerCon SCARA Robot from the current position (-50.000, 300.000, 110.000, 0.000) to the target position (-49.900, 299.800, 109.000, 180.000);
→Change the target position to (-, -, -, 180.000) (no indication to 1st to 3rd arms (not to move))

(Note 6) For the tracking operation of XSEL-RA/SA/RAX/SAX/RAXD/SAXD controllers, the output function of indicated distance on/off is not to be used. (Error No. B97 “Position Output Operation Data Indication Error”)

### 5.5.6 Common Notes for Caution

Stated below are the caution notes in common for each output function.

· The position data setting indicated in Operation 1 is to be valid for the output operation of the arc movement commands (CIR, ARC, CIR2, ARC2, CIRS, ARCS, ARCD and ARCC) and the arch motion command (ARCH). In these commands, only one output operation is valid in one motion. In case of a command to set the position data also to Operation 2, the output operation data gets ignored.

· In the output operation for PATH and PSPL Commands, settings in all the position data from the start position in Operation 1 to the end position in Operation 2 are valid.

· The output operation by the position output operation data and the output operation by the output portion in SEL Command are processed individually. Therefore, indicating both in one command is also available.

· 16 units are available at maximum at the same time in total for all the programs for the output delay timers in the position output operation and the shot pulse timers in Output 1. Error No. B04 “Simultaneous Excess Use in 1-Shot Pulse Output Error” occurs when timers more than that are used.

· At the end of the program (including when cancelling an error), the output delay timers of the position output operation and the shot pulse timers for Output 1 executed in the applicable program are inactivated, and the output operation after that will all be cancelled.

· At the pause of the program, the output delay timers and Output 1 shot pulse timers will not be paused. Therefore, pay attention as the output operation will be conducted after the time of the timer has been passed even in a pause.

· In case that, before completing the delay and one-shot pulse output to the position output “A”, a certain position output, the position output “B” is conducted from the same program to the same output port and flag numbers, the output operation of A will be overwritten by the output operation of B.

Example)

```
Prg 1    : • • •
          MOVP   1          ON    316   2.000   0.000      • • (1)
          MOVP   2          OFF   316   0.500   0.000      • • (2)
                          ・ ・ ・
```

⇒ If Step (2) is finished before passing 2.000 sec. after Step (1) is finished, the position output in Step (1) will be cancelled and Output Port No. 316 will not turn on.

· The actual output timing of the position output operation has tolerance of 3 to 4msec.

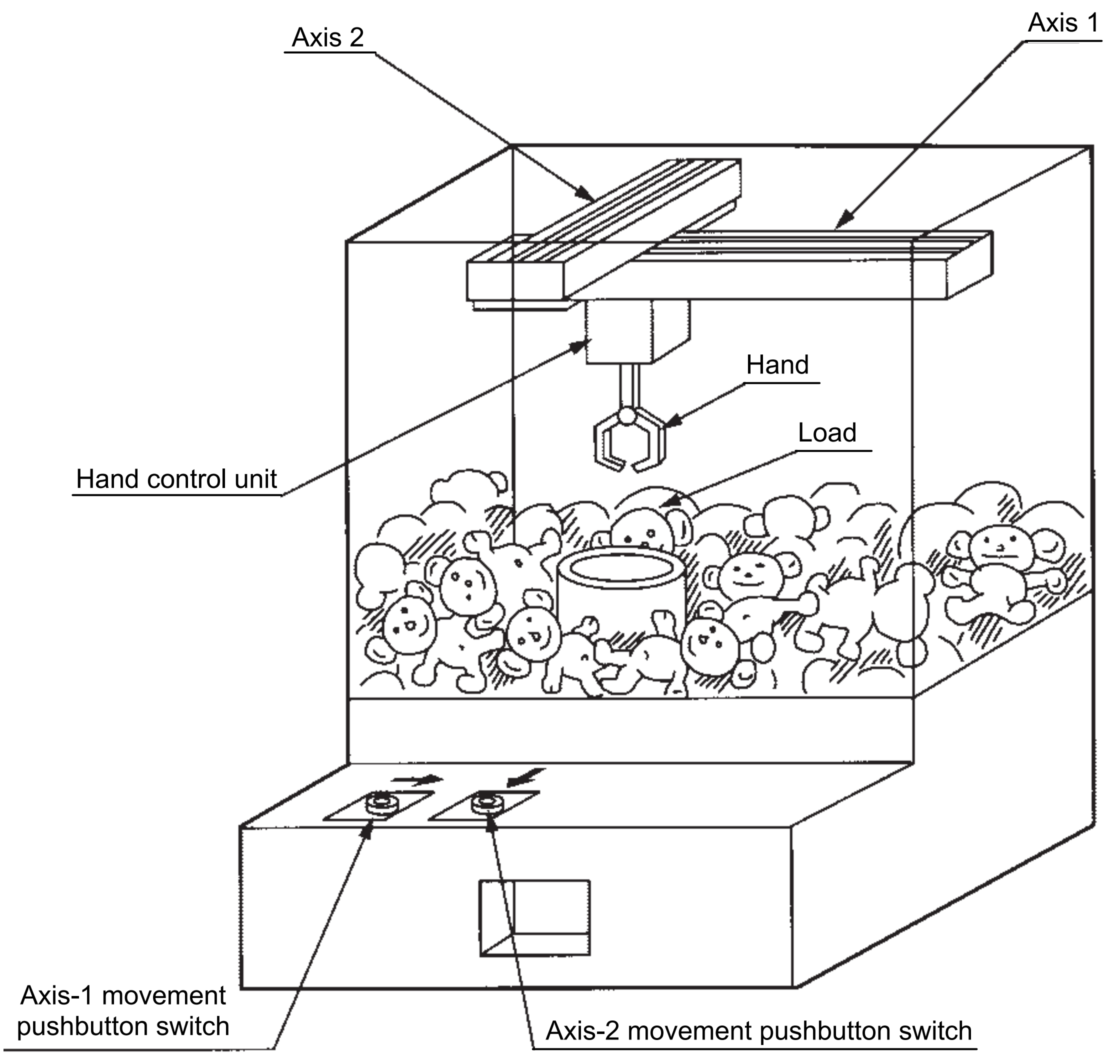
### 5.5.7 Other Caution Notes

- The position output operation feature will not respond when moving an actuator by operating such as MV Button in the position data edit window in the PC software.
- Make sure to always use tools applicable to the feature when using the position output operation feature. In case a tool not applicable for the feature is connected and the position data is edited, the output operation data will be deleted.

# 6. Program Examples

## 6.1 Operation by Jog Command Doll-Picking Game Machine

(1) Overview of the system
This system is a doll-picking game machine consisting of axis-1 and axis-2 actuators. Pushbutton switches corresponding to the two axes are provided on an external operation switch box, and these switches are used to move the actuators to a desired position to grab and pick up dolls inside the case.

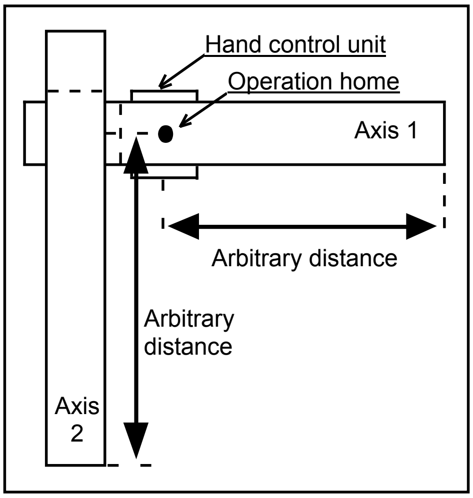


(2) Explanation of the operation
How this system operates is explained.

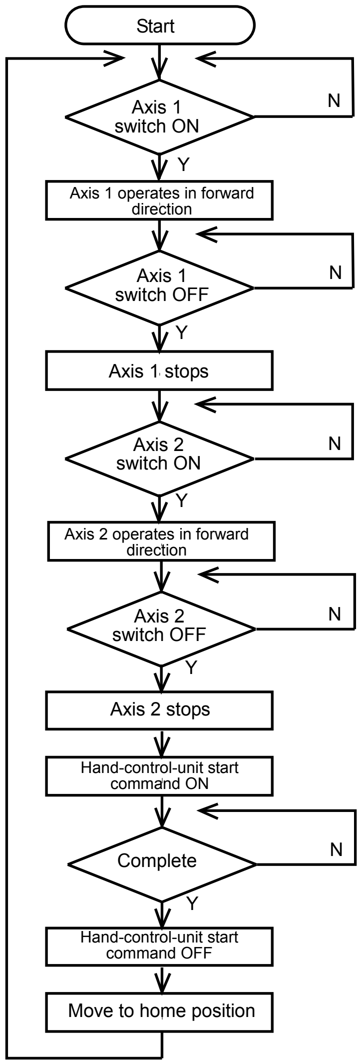
1) Wait for the axis-1 movement pushbutton switch to turn ON.
2) The X-axis moves while the pushbutton switch is ON, and stops when the switch turns OFF.
3) Wait for the axis-2 movement pushbutton switch to turn ON.
4) The Y-axis moves while the pushbutton switch is ON, and stops when the switch turns OFF.
5) Output a start command to the hand control unit.
6) Wait for an operation completion input from the hand control unit.
7) Move to the home after the input is received.

The above operation will be repeated. The operation position, external I/O assignments and operation flow chart of this operation are shown below:

Operation Position




Operation Flow Chart




I/O Assignments

| Category | | I/O No. | Signal name | Specification |
|---|---|---|---|---|
| XSEL | Input | 16 | Axis-1 movement command | Pushbutton switch |
| | | 17 | Axis-2 movement command | Pushbutton switch |
| | | 18 | Hand operation completion | External control unit |
| | Output | 309 | Hand start command | 24V DC |
| | * Flag is not used. | | | |

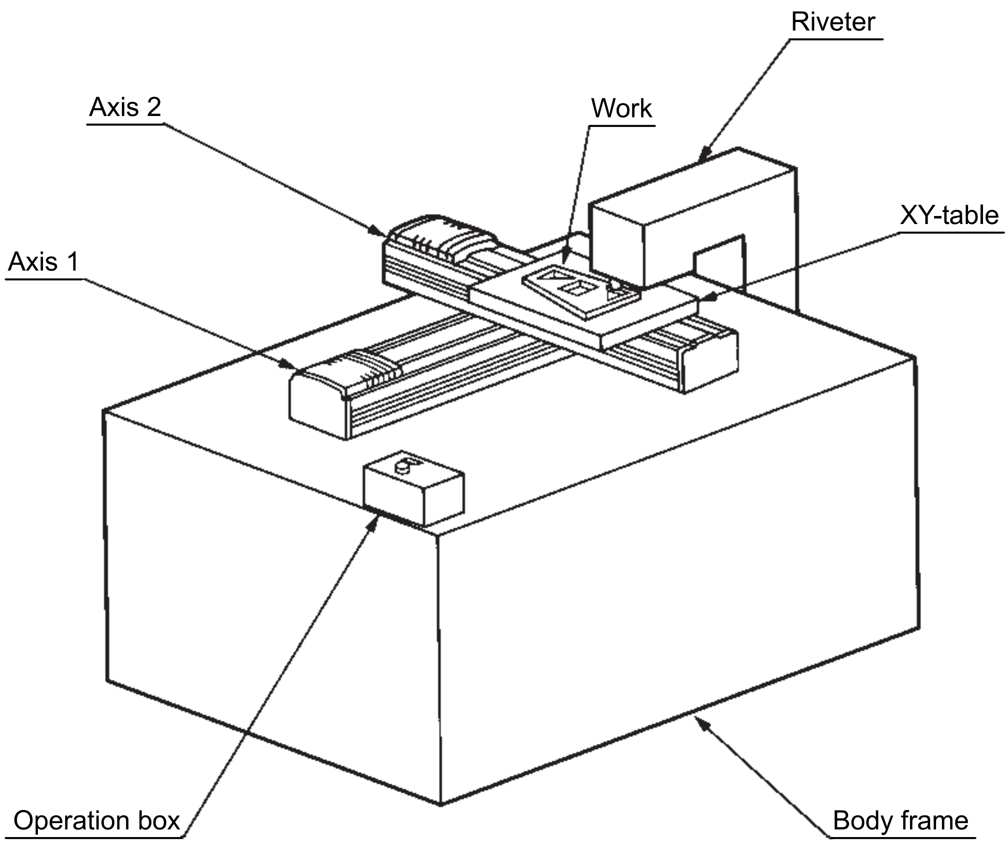
(3) XSEL Controller application program

| Step | E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | HOME | 11 | | | Axes 1 and 2 return to home (servo ON). |
| 2 | | | | VEL | 400 | | | Set speed to 400mm/s. |
| 3 | | | | TAG | 1 | | | |
| 4 | | | | WTON | 16 | | | Wait for input from axis-1 movement switch. |
| 5 | | | | JFWN | 1 | 16 | | Move forward while axis-1 movement switch is ON. |
| 6 | | | | WTON | 17 | | | Wait for input from axis-2 movement switch. |
| 7 | | | | JFWN | 10 | 17 | | Move forward while axis-2 movement switch is ON. |
| 8 | | | | BTON | 309 | | | Start command for external control unit turns ON. |
| 9 | | | | WTON | 18 | | | Wait for external control unit to complete operation. |
| 10 | | | | BTOF | 309 | | | Start command for external control unit turns OFF. |
| 11 | | | | JBWF | 11 | 18 | | Axes 1 and 2 move backward while 18 is ON. |
| 12 | | | | GOTO | 1 | | | Jump to TAG1. |
| 13 | | | | | | | | |
| 14 | | | | | | | | |
| 15 | | | | | | | | |
| 16 | | | | | | | | |
| 17 | | | | | | | | |
| 18 | | | | | | | | |
| 19 | | | | | | | | |
| 20 | | | | | | | | |
| 21 | | | | | | | | |
| 22 | | | | | | | | |
| 23 | | | | | | | | |
| 24 | | | | | | | | |
| 25 | | | | | | | | |
| 26 | | | | | | | | |
| 27 | | | | | | | | |
| 28 | | | | | | | | |
| 29 | | | | | | | | |
| 30 | | | | | | | | |
| 31 | | | | | | | | |
| 32 | | | | | | | | |

## 6.2 Operation by Point Movement Command Riveting System

(1) Overview of the system

This system is a riveting system consisting of an XY-table operated by axis-1 and axis-2 actuators and a riveter. By setting a work on the XY-table at the operation home and turning ON the start switch, rivets will be driven at the three points specified on the work.

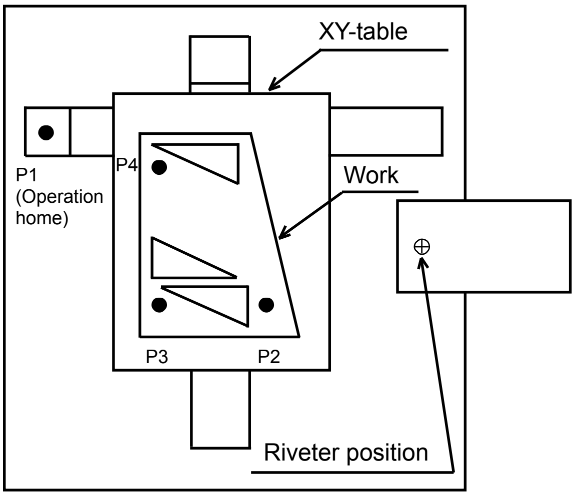


(2) Explanation of the operation
How this system operates is explained.

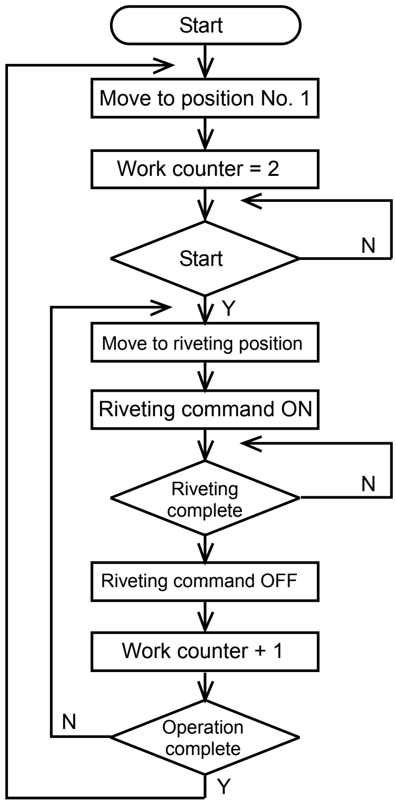
1) The XY-table moves to the operation home (P1) and waits.
2) The operator sets a work on the XY-table and turns ON the start switch.
3) The XY-table moves to riveting position No. 1 (P2) on the work and a riveting command is output to the riveter.
4) When the riveter completes the riveting operation and a completion signal is input, the table will move to riveting position No. 2 (P3) and then No. 3 (P4), in the same manner.
5) When all three points have been riveted, the table will return to the operation home (P1).

The above operation will be repeated. The operation position, external I/O assignments and operation flow chart of this operation are shown below:

Operation Position




Operation Flow Chart




I/O Assignments

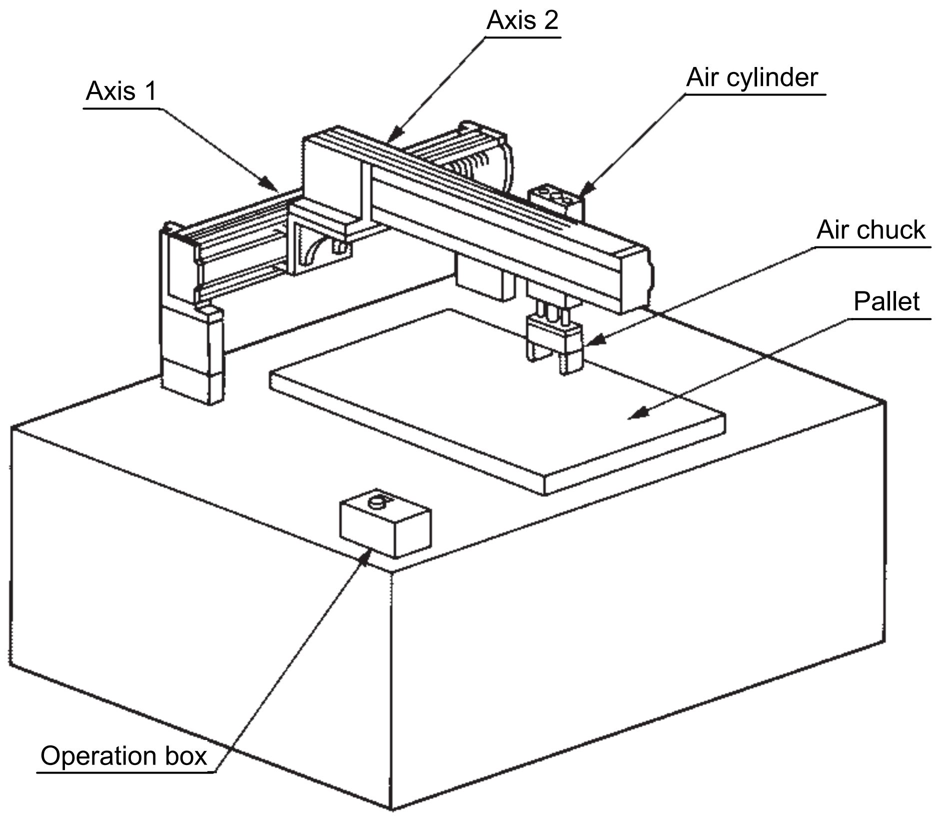
| Category | | I/O No. | Signal name | Specification |
|---|---|---|---|---|
| XSEL | Input | 16 | Start command | Pushbutton switch |
| | | 17 | Riveting completion | Contact signal |
| | Output | 309 | Riveting command | 24V DC |
| | * Flag is used from 600. | | | |

(3) XSEL Controller application program

| Step | E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | HOME | 11 | | | XY-table returns to home (servo ON). |
| 2 | | | | VEL | 400 | | | Set speed to 400mm/s. |
| 3 | | | | TAG | 1 | | | |
| 4 | | | | MOVL | 1 | | | Move to position No. 1 (home of work). |
| 5 | | | | LET | 1 | 2 | | Set 2 in work counter. |
| 6 | | | | BTOF | 600 | | | Clear completion flag. |
| 7 | | | | WTON | 16 | | | Wait for start command. |
| 8 | | | | TAG | 2 | | | |
| 9 | | | | MOVL | *1 | | | Move to work counter position. |
| 10 | | | | BTON | 309 | | | Riveting command turns ON. |
| 11 | | | | WTON | 17 | | | Wait for riveting to complete. |
| 12 | | | | BTOF | 309 | | | Riveting command turns OFF. |
| 13 | | | | ADD | 1 | 1 | | Increment work counter by 1. |
| 14 | | | | CPEQ | 1 | 5 | 600 | Turns ON flag if operation is complete. |
| 15 | | N | 600 | GOTO | 2 | | | Jump to TAG2 if not complete. |
| 16 | | | | GOTO | 1 | | | Jump to TAG1 if complete. |
| 17 | | | | | | | | |
| 18 | | | | | | | | |
| 19 | | | | | | | | |
| 20 | | | | | | | | |
| 21 | | | | | | | | |
| 22 | | | | | | | | |
| 23 | | | | | | | | |
| 24 | | | | | | | | |
| 25 | | | | | | | | |
| 26 | | | | | | | | |
| 27 | | | | | | | | |
| 28 | | | | | | | | |
| 29 | | | | | | | | |
| 30 | | | | | | | | |
| 31 | | | | | | | | |
| 32 | | | | | | | | |

## 6.3 Palletizing Operation Palletizing System

(1) Overview of the system
This system is a palletizing system consisting of axis-1 and axis-2 actuators and a Z-axis air cylinder. It clamps a work at the work feed point and transfers it onto a pallet, and repeats this operation in a sequence.
(Operation is implemented by an offset command without using a palletizing function.)

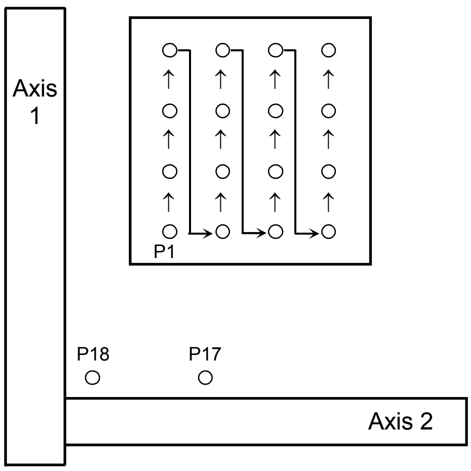


(2) Explanation of the operation
How this system operates is explained.

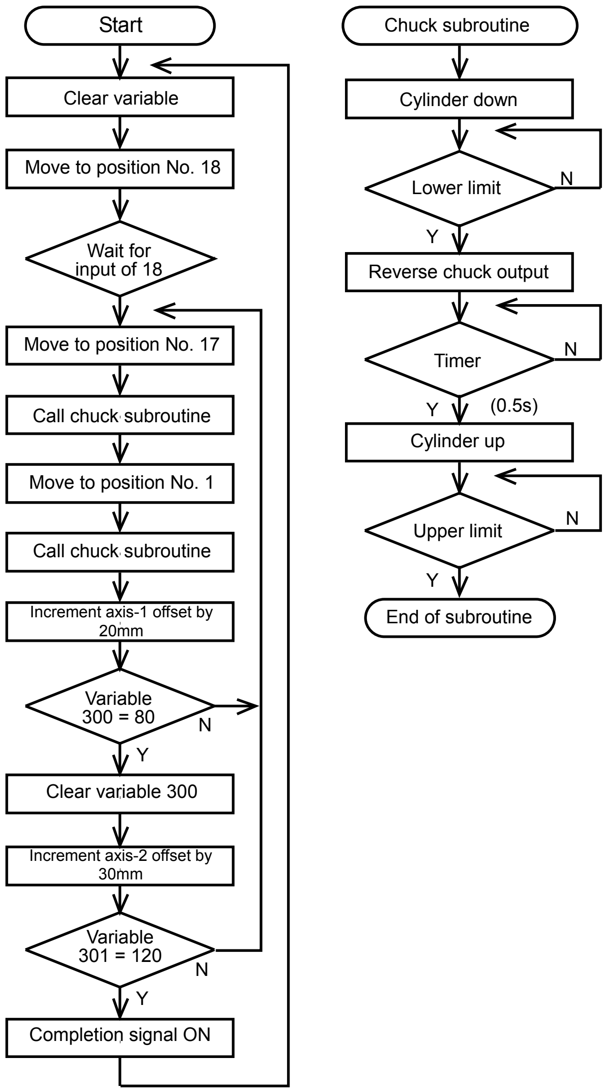
1) Move to the standby point and wait for a start input.
2) Move to the work feed point after a start input is received.
3) The Z-axis comes down and the air chuck clamps the work.
4) The Z-axis rises and moves to above the pallet.
5) The Z-axis comes down and releases the work.
6) The Z-axis rises and moves to above the work feed point.
7) When the pallet becomes full, a pallet-completion indicator signal is output. The axes move to P18 and then wait for restart.

The above operation will be repeated. The operation position, external I/O assignments and operation flow chart of this operation are shown below:

Operation Position




Operation Flow Chart




I/O Assignments

| Category | | I/O No. | Signal name | Specification |
|---|---|---|---|---|
| XSEL | Input | 16 | Z-axis cylinder upper limit | Proximity SW |
| | | 17 | Z-axis cylinder lower limit | Proximity SW |
| | | 18 | Start | Pushbutton switch |
| | Output | 309 | Z-axis cylinder SV | 24V DC |
| | | 310 | Z-axis chuck SV | 24V DC |
| | | 311 | Pallet-completion indicator | 24V DC |
| | * Flag is used from 600. | | | |

Pallet specifications
Axis-1 direction: 20mm pitch
Axis-2 direction: 30mm pitch

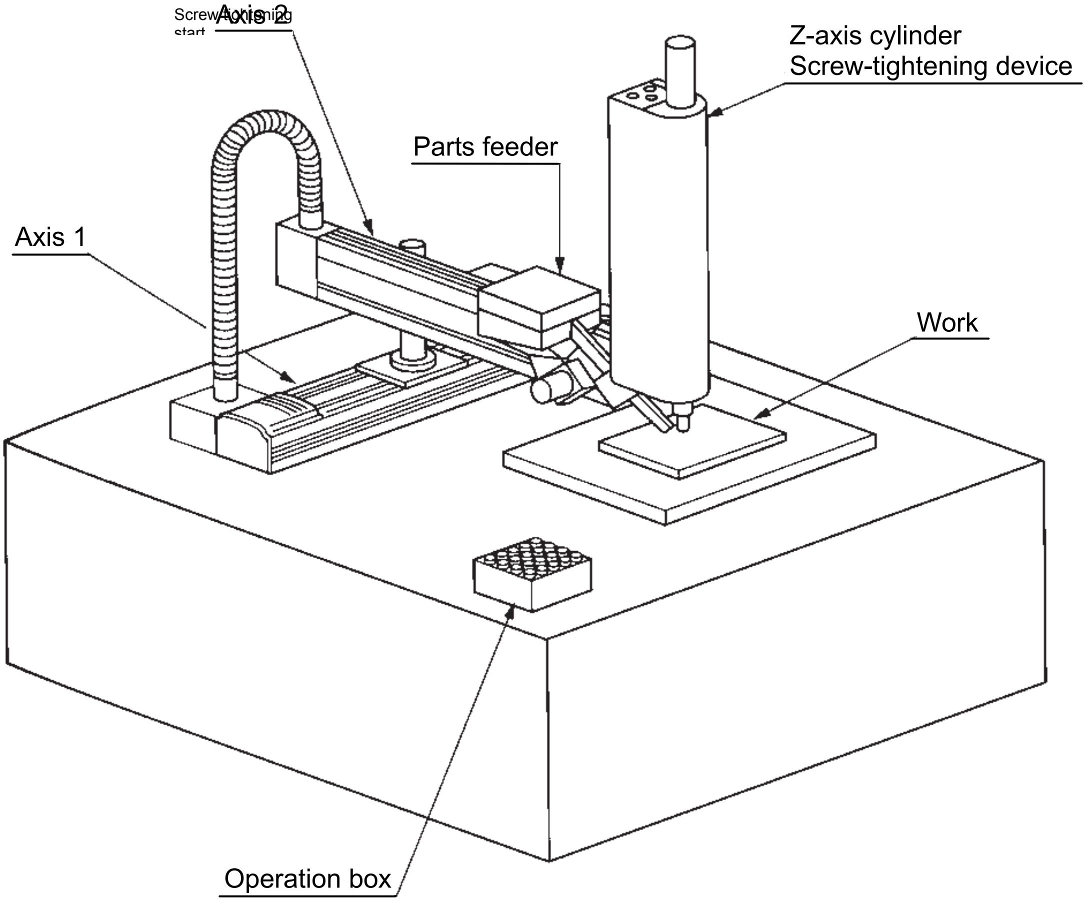
(3) XSEL Controller application program

| Step | E | N | Cnd | Cmnd | Operand 1 | Operand 2 | Pst | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | HOME | 11 | | | Axes 1 and 2 return to home. |
| 2 | | | | VEL | 100 | | | Set speed to 100mm/s. |
| 3 | | | | ACC | 0.2 | | | Acceleration/deceleration: 0.2G |
| 4 | | | | TAG | 1 | | | |
| 5 | | | | LET | 300 | 0 | | Clear variable. |
| 6 | | | | LET | 301 | 0 | | Clear variable. |
| 7 | | | | OFST | 11 | 0 | | Clear offset value. |
| 8 | | | | MOVL | 18 | | | Move to position No. 18. |
| 9 | | | | WTON | 18 | | | Wait for start input. |
| 10 | | | | BTOF | 311 | | | Output 311 turns OFF. |
| 11 | | | | TAG | 2 | | | |
| 12 | | | | OFST | 11 | 0 | | Clear offset value. |
| 13 | | | | MOVL | 17 | | | Move to position No. 17. |
| 14 | | | | EXSR | 1 | | | Call chuck subroutine (chuck). |
| 15 | | | | OFST | 1 | *300 | | Offset axis 1 by value in variable 300. |
| 16 | | | | OFST | 10 | *301 | | Offset axis 2 by value in variable 301. |
| 17 | | | | MOVL | 1 | | | Move to position No. 1 + offset value. |
| 18 | | | | EXSR | 1 | | | Call chuck subroutine (unchuck). |
| 19 | | | | ADD | 300 | 20 | | Add 20 to variable 300. |
| 20 | | | | CPEQ | 300 | 80 | 600 | Turn ON flag 600 if variable 300 = 80. |
| 21 | | N | 600 | GOTO | 2 | | | Jump to TAG2 if flag 600 is OFF. |
| 22 | | | | LET | 300 | 0 | | Clear variable 300. |
| 23 | | | | ADD | 301 | 30 | | Add 30 to variable 301. |
| 24 | | | | CPEQ | 301 | 120 | 601 | Turn ON flag 601 if variable 301 = 120. |
| 25 | | N | 601 | GOTO | 2 | | | Jump to TAG2 if flag 601 is OFF. |
| 26 | | | | BTON | 311 | | | Output 311 turns ON. |
| 27 | | | | GOTO | 1 | | | Jump to TAG1. |
| 28 | | | | BGSR | 1 | | | Start chuck subroutine. |
| 29 | | | | BTON | 309 | | | Z-axis cylinder down |
| 30 | | | | WTON | 17 | | | Wait for lower-limit input. |
| 31 | | | | BTNT | 310 | | | Reverse air-chuck output. |
| 32 | | | | TIMW | 0.5 | | | Timer: 0.5 second |
| 33 | | | | BTOF | 309 | | | Z-axis cylinder up |
| 34 | | | | WTON | 16 | | | Wait for upper-limit input. |
| 35 | | | | EDSR | | | | End of chuck subroutine |
| 36 | | | | | | | | |
| 37 | | | | | | | | |
| 38 | | | | | | | | |
| 39 | | | | | | | | |

## 6.4 Screw-Tightening Machine

(1) Overview of the system

This system consists of axis-1 and axis-2 actuators, Z-axis cylinder, screw-tightening device and parts feeder, and tightens the screws fed by the parts feeder at the specified positions on the work.




(2) Equipment

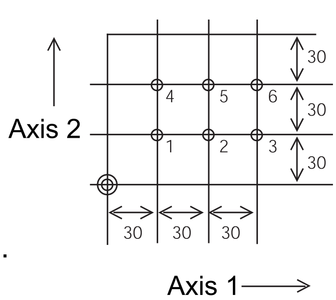
| | |
|---|---|
| Screw-tightening machine (for Z-axis) | |
| Actuators (for axes 1 and 2) | IAI's 60W servo motor/actuator with 300mm stroke × 2 |
| Controller | IAI's XSEL controller |

(3) Explanation of the operation

(1) Tighten six screws at 30mm pitches on axes 1 and 2.
  1) The actuators move to a screw-tightening position.
  2) The Z-axis air cylinder of the screw-tightening machine comes down.
  3) The screw-tightening machine starts operating.
  4) When the screw tightening is complete, the Z-axis air cylinder rises.
  5) The actuators move to the next position.

(2) The parts feeder operates in parallel with the above operation.
  1) The parts feeder starts when screws are short.
  2) The parts feeder stops when the screws are fully loaded.

Coordinates



[Hardware]

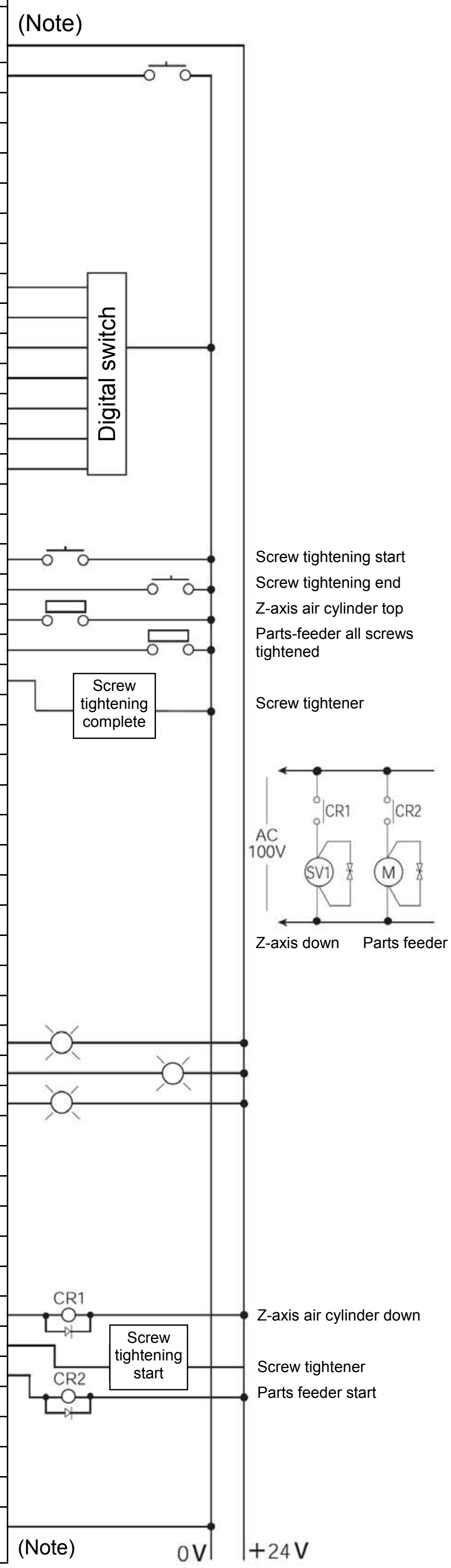
1) I/O assignment

I/O connector (50 pins)

| Pin No. | Category | Port No. | Function | Cable color |
|---|---|---|---|---|
| 1 | Input | – | General-purpose: NC / Compact: +24V input | Brown-1 |
| 2 | | 000 | Program start | Red-1 |
| 3 | | 001 | General-purpose input | Orange-1 |
| 4 | | 002 | General-purpose input | Yellow-1 |
| 5 | | 003 | General-purpose input | Green-1 |
| 6 | | 004 | General-purpose input | Blue-1 |
| 7 | | 005 | General-purpose input | Purple-1 |
| 8 | | 006 | General-purpose input | Gray-1 |
| 9 | | 007 | Program specification (PRG No.1) | White-1 |
| 10 | | 008 | Program specification (PRG No.2) | Black-1 |
| 11 | | 009 | Program specification (PRG No.4) | Brown-2 |
| 12 | | 010 | Program specification (PRG No.8) | Red-2 |
| 13 | | 011 | Program specification (PRG No.10) | Orange-2 |
| 14 | | 012 | Program specification (PRG No.20) | Yellow-2 |
| 15 | | 013 | Program specification (PRG No.40) | Green-2 |
| 16 | | 014 | General-purpose input | Blue-2 |
| 17 | | 015 | General-purpose input | Purple-2 |
| 18 | | 016 | Screw tightening start | Gray-2 |
| 19 | | 017 | Screw tightening end | White-2 |
| 20 | | 018 | Z-axis air cylinder top | Black-2 |
| 21 | | 019 | Parts-feeder all screws tightened | Brown-3 |
| 22 | | 020 | Screw tightening complete | Red-3 |
| 23 | | 021 | General-purpose input | Orange-3 |
| 24 | | 022 | General-purpose input | Yellow-3 |
| 25 | | 023 | General-purpose input | Green-3 |
| 26 | | 024 | General-purpose input | Blue-3 |
| 27 | | 025 | General-purpose input | Purple-3 |
| 28 | | 026 | General-purpose input | Gray-3 |
| 29 | | 027 | General-purpose input | White-3 |
| 30 | | 028 | General-purpose input | Black-3 |
| 31 | | 029 | General-purpose input | Brown-4 |
| 32 | | 030 | General-purpose input | Red-4 |
| 33 | | 031 | General-purpose input | Orange-4 |
| 34 | Output | 300 | Alarm output | Yellow-4 |
| 35 | | 301 | Ready output | Green-4 |
| 36 | | 302 | Emergency stop output | Blue-4 |
| 37 | | 303 | General-purpose output | Purple-4 |
| 38 | | 304 | General-purpose output | Gray-4 |
| 39 | | 305 | General-purpose output | White-4 |
| 40 | | 306 | General-purpose output | Black-4 |
| 41 | | 307 | General-purpose output | Brown-5 |
| 42 | | 308 | General-purpose output | Red-5 |
| 43 | | 309 | Z-axis air cylinder down | Orange-5 |
| 44 | | 310 | Screw tightening start | Yellow-5 |
| 45 | | 311 | Parts feeder start | Green-5 |
| 46 | | 312 | General-purpose output | Blue-5 |
| 47 | | 313 | General-purpose output | Purple-5 |
| 48 | | 314 | General-purpose output | Gray-5 |
| 49 | | 315 | General-purpose output | White-5 |
| 50 | | – | General-purpose: NC / Compact: 0V | Black-5 |

2) Layout drawing

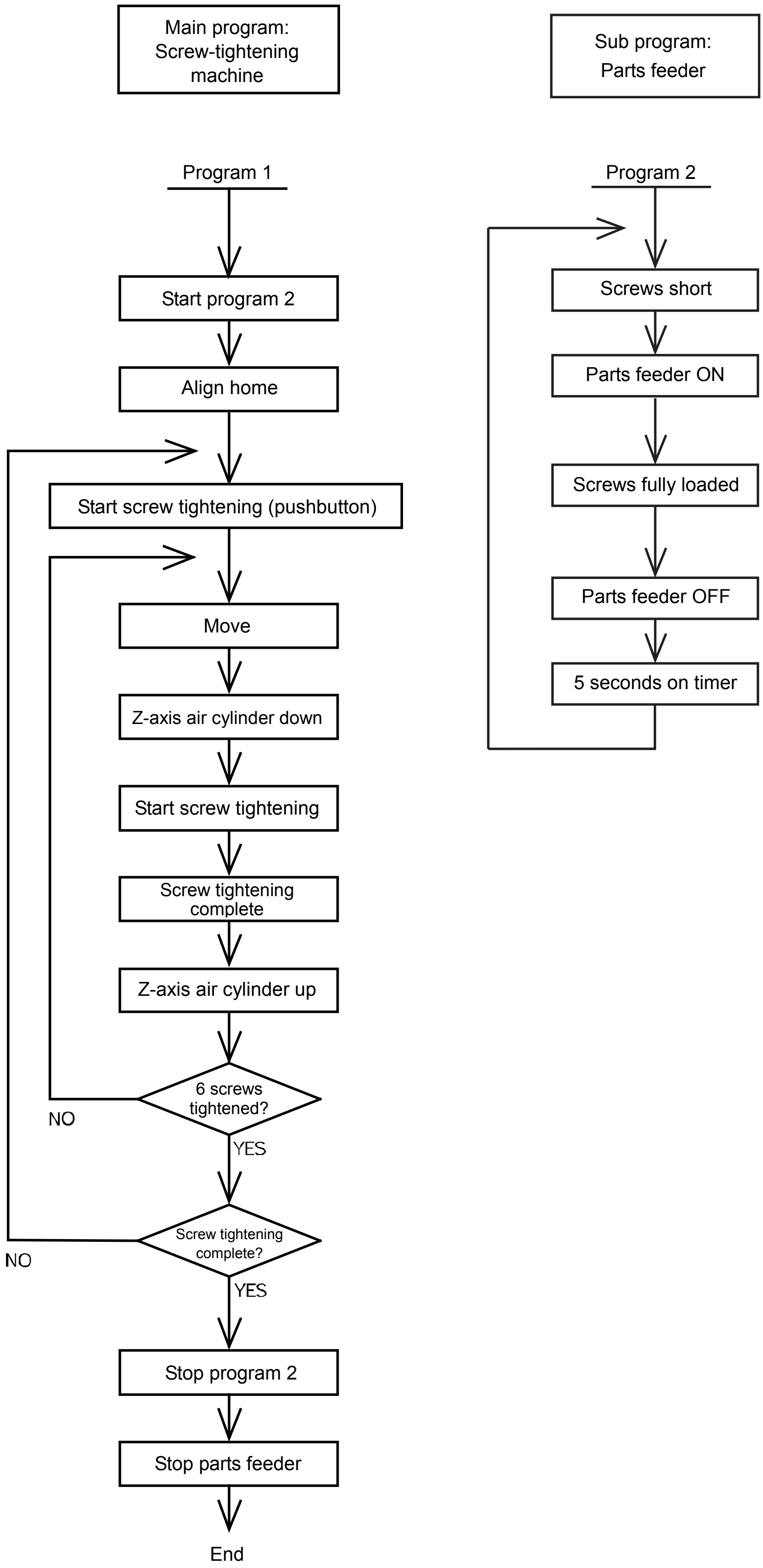
| Pin No. | Category | Port No. | Function |
|---|---|---|---|
| 1 | Input | – | General-purpose: NC / Compact: +24V input |
| 2 | | 000 | Program start |
| 3 | | 001 | General-purpose input |
| 4 | | 002 | General-purpose input |
| 5 | | 003 | General-purpose input |
| 6 | | 004 | General-purpose input |
| 7 | | 005 | General-purpose input |
| 8 | | 006 | General-purpose input |
| 9 | | 007 | Program specification (PRG No.1) |
| 10 | | 008 | Program specification (PRG No.2) |
| 11 | | 009 | Program specification (PRG No.4) |
| 12 | | 010 | Program specification (PRG No.8) |
| 13 | | 011 | Program specification (PRG No.10) |
| 14 | | 012 | Program specification (PRG No.20) |
| 15 | | 013 | Program specification (PRG No.40) |
| 16 | | 014 | General-purpose input |
| 17 | | 015 | General-purpose input |
| 18 | | 016 | Screw tightening start |
| 19 | | 017 | Screw tightening end |
| 20 | | 018 | Z-axis air cylinder top |
| 21 | | 019 | Parts-feeder all screws tightened |
| 22 | | 020 | Screw tightening complete |
| 23 | | 021 | General-purpose input |
| 24 | | 022 | General-purpose input |
| 25 | | 023 | General-purpose input |
| 26 | | 024 | General-purpose input |
| 27 | | 025 | General-purpose input |
| 28 | | 026 | General-purpose input |
| 29 | | 027 | General-purpose input |
| 30 | | 028 | General-purpose input |
| 31 | | 029 | General-purpose input |
| 32 | | 030 | General-purpose input |
| 33 | | 031 | General-purpose input |
| 34 | Output | 300 | Alarm output |
| 35 | | 301 | Ready output |
| 36 | | 302 | Emergency stop output |
| 37 | | 303 | General-purpose output |
| 38 | | 304 | General-purpose output |
| 39 | | 305 | General-purpose output |
| 40 | | 306 | General-purpose output |
| 41 | | 307 | General-purpose output |
| 42 | | 308 | General-purpose output |
| 43 | | 309 | Z-axis air cylinder down |
| 44 | | 310 | Screw tightening start |
| 45 | | 311 | Parts feeder start |
| 46 | | 312 | General-purpose output |
| 47 | | 313 | General-purpose output |
| 48 | | 314 | General-purpose output |
| 49 | | 315 | General-purpose output |
| 50 | | – | General-purpose: NC / Compact: 0V |




Pin No. 1 and 50 are not connected for general-purpose types.
Pin No. 1 is connected to +24V, while Pin No. 50 is connected to 0V, for compact types.

[Software]

1) Control flow chart



2) Main program
Screw-tightening program No. 1

Application program

| Comment | Extension condition | Input condition | Command | | | Output condition | Comment |
|---|---|---|---|---|---|---|---|
| | AND, OR | I/O, flag | Command | Operand 1 | Operand 2 | Output port, flag | |
| 1 | | | EXPG | 2 | | | Start program 2. |
| 2 | | | HOME | 11 | | | Align home. |
| 3 | | | VEL | 100 | | | Speed: 100mm/sec |
| 4 | | | ACC | 0.3 | | | Acceleration: 0.3G |
| 5 | | | TAG | 1 | | | Jump destination at restart |
| 6 | | | WTON | 16 | | | Screw-tightening start pushbutton |
| 7 | | | LET | 1 | 1 | | Set screw counter. |
| 8 | | | TAG | 2 | | | Jump destination after tightening one screw |
| 9 | | | MOVL | *1 | | | Move. |
| 10 | | | BTON | 309 | | | Z-axis air cylinder down |
| 11 | | | BTON | 310 | | | Start screw tightening. |
| 12 | | | WTON | 20 | | | Screw tightening complete. |
| 13 | | | BTOF | 309 | 310 | | Cylinder up, screw tightening stopped. |
| 14 | | | WTON | 18 | | | Check Z-axis air cylinder top position. |
| 15 | | | ADD | 1 | 1 | | Increment screw counter by 1. |
| 16 | | | CPEQ | 1 | 7 | 900 | Compare after tightening six screws. |
| 17 | | N900 | GOTO | 2 | | | Go to next screw-tightening cycle after tightening one screw. |
| 18 | | N17 | GOTO | 1 | | | Restart screw tightening. |
| 19 | | | ABPG | 2 | | | Stop program 2. |
| 20 | | | BTOF | 311 | | | Stop parts feeder. |
| 21 | | | EXIT | | | | End of program 1 |

Position program

| No. | X | Y |
|---|---|---|
| 1 | 30 | 30 |
| 2 | 60 | 30 |
| 3 | 90 | 30 |
| 4 | 30 | 60 |
| 5 | 60 | 60 |
| 6 | 90 | 60 |

3) Sub program
Parts feeder program No. 2

Application program

| Comment | Extension condition | Input condition | Command | | | Output condition | Comment |
|---|---|---|---|---|---|---|---|
| | AND, OR | I/O, flag | Command | Operand 1 | Operand 2 | Output port, flag | |
| 1 | | | TAG | 1 | | | Jump destination for repeating |
| 2 | | | WTOF | 19 | | | Screws short. |
| 3 | | | BTON | 311 | | | Start parts feeder. |
| 4 | | | WTON | 19 | | | Screws fully loaded. |
| 5 | | | BTOF | 311 | | | Stop parts feeder. |
| 6 | | | TIMW | 5 | | | 5 seconds on restart timer |
| 7 | | | GOTO | 1 | | | Repeat. |

# 7. Appendix

## ASCII Code Table

| Upper 3 bits →<br>↓ Lower 4 bits | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | NUL | DLE | SP | 0 | @ | P | ` | p |
| 1 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 2 | STX | DC2 | " | 2 | B | R | b | r |
| 3 | ETX | DC3 | # | 3 | C | S | c | s |
| 4 | EOT | DC4 | $ | 4 | D | T | d | t |
| 5 | ENQ | NAK | % | 5 | E | U | e | u |
| 6 | ACK | SYN | & | 6 | F | V | f | v |
| 7 | BEL | ETB | ' | 7 | G | W | g | w |
| 8 | BS | CAN | ( | 8 | H | X | h | x |
| 9 | HT | EM | ) | 9 | I | Y | i | y |
| A | LF/NL | SUB | * | : | J | Z | j | z |
| B | VT | ESC | + | ; | K | [ | k | { |
| C | FF | FS | , | < | L | \ | l | \| |
| D | CR | GS | - | = | M | ] | m | } |
| E | SO | RS | . | > | N | ^ | n | ~ |
| F | SI | US | / | ? | O | _ | o | DEL |

# Change History

| Revision Date | Description of Revision |
|---|---|
| 2010.11 | First edition<br>1A → 1B (note corrected)<br>Correction of page number for each command in pages 1 to 19, 254 to 260<br>Pages 44 Correction in reference for (3)<br>Pages 103 Correction in right-hand rule<br>Pages 248 Addition of note to state to refer to TT Instruction Manual<br>Pages 295 Correction to the note of continuous operation command reference<br>Pages 489 and 491 Correction to the note of palletizing reference |
| 2011.11 | Second edition<br>Contents changed in Safety Guide<br>Caution notes added for when working with two or more persons<br>Contents deleted regarding Vertical Articulated and Rectangular 6-Axes Robots |
| 2011.12 | Edition 2B<br>Note corrected etc. |
| 2012.04 | Third edition<br>Note added for PCT/QCT Types for CT4 Actuator and explanation changed for related command (SCRV) |
| 2012.09 | Fourth edition<br>Note added for XSEL-R/S/RX/SX/RXD/SXD types and related commands added and changed |
| 2012.10 | Fifth edition<br>Note revised |
| 2013.10 | Sixth edition<br>TTA added |
| 2014.06 | Seventh edition<br>MSEL added |
| 2014.08 | Eighth edition<br>Notes added for MSEL Cartesian Type application |
| 2015.09 | Edition 8C<br>Pages 164 Correction of the explanation for palletizing calculation<br>Pages 299, 300 Correction of the SCRV Command of TT, TTA and MSEL |
| 2016.01 | Ninth edition<br>Contents related to coordinate system definition unit axes set in Coordinate System Definition 1 Constructing Axes Setting for MSEL-PC/PG and TTA added in explanation of commands |
| 2016.02 | Tenth edition<br>Contents added for work coordinate system, support of tool coordinate system (GTVD command) |

| Revision Date | Description of Revision |
|---|---|
| 2016.07 | Tenth edition<br>Contents added for work coordinate system, support of tool coordinate system (GTVD command)<br>XSEL-RA/SA/RAX/SAX/RAXD/SAXD added<br>1.4.5 Coordinate System of TTA, MSEL-PC/PG added<br>5.4 Key Characteristics of Actuator Control Commands and Points to Note added<br>5.5 Position Output Operation Features added |
| 2016.09 | Edition 10B<br>Pages 399, 401 Torque limit setting parameter added for XSEL-J/K, MSEL-PC/PG and TTA |
| 2016.09 | Edition 10C<br>Pages 571 Content partially deleted in Caution [6] for XCAS Command<br>Note such as way of signal on and off added in 5.5 Position Output Operation Feature |
| 2016.09 | Edition 10D<br>Pages 236, 244 Application added for variables (integers) from 2000 to 2799 in XSEL-R/S<br>Pages 337 Correction Made from ACMX Command MSEL ○ → × |
| 2016.10 | Edition 10E<br>Pages 238, 239 SSEL Number of symbol definitions 500 → 1000<br>SSEL Number of symbol used in commands 2500 → 5000<br>Pages 245 "How to Deal with Character String Literals" added<br>Pages 294 Correction Made from Extension condition and Input condition of EXSR Prohibited → Optional |
| 2016.12 | Edition 10F<br>Pages 601 Correction made to tell TTA and MSEL applied to TB-02 are later first edition |
| 2017.03 | Edition 10G<br>ASIN, ACOS, DTOR, RTOD, ABS, SGN, ECMD7, ECMD8 and ECMD9 commands added<br>Pages 124 7051 to 7054 added in TTA Virtual Input/Output Ports<br>Pages 128 7051 to 7054 added in MSEL Virtual Input/Output Ports |
| 2017.06 | Edition 10I<br>ECMD280 to 282, ECMD290 to 292 and ECMD300 commands added<br>XPTH command added<br>XACH, XACZ, XAEX, XATG and XOAZ commands added |
| 2017.08 | Edition 10J<br>QRTN Command: Note added stating applicable for XSEL-RA/SA/RAX/SAX/RAXD/SAXD<br>ECMD250 command: Note added stating applicable for MSEL<br>NOT, LSFT, RSFT, CNTP, IPCN, ECMD10 and ECMD11 commands added |
| 2017.09 | Edition 10K<br>ECMD1 TTA (V1.22 or later) command added |

| Revision Date | Description of Revision |
|---|---|
| 2018.02 | Edition 10L<br>CLLV, COL, GCLX, COMP, SCLO 0, SCLO 1 and SCLG command added |
| 2018.08 | Edition 10M<br>Contents added for SCARA Robot in 3.6.5 How to Use Palletizing Function |

Manual No.: ME0224-10M (August 2018)

IAI
Quality and Innovation


## IAI Corporation

Head Office: 577-1 Obane Shimizu-KU Shizuoka City Shizuoka 424-0103, Japan
TEL +81-54-364-5105 FAX +81-54-364-2589
website: www.iai-robot.co.jp/

Technical Support available in USA, Europe and China

## IAI America, Inc.

Head Office: 2690 W. 237th Street, Torrance, CA 90505
TEL (310) 891-6015 FAX (310) 891-0815
Chicago Office: 110 East State Parkway, Schaumburg, IL 60173
TEL (847) 908-1400 FAX (847) 908-1399
Atlanta Office: 1220 Kennestone Circle, Suite 108, Marietta, GA 30066
TEL (678) 354-9470 FAX (678) 354-9471
website: www.intelligentactuator.com

## IAI Industrieroboter GmbH

Ober der Röth 4, D-65824 Schwalbach am Taunus, Germany
TEL 06196-88950 FAX 06196-889524

## IAI (Shanghai) Co., Ltd.

SHANGHAI JIAHUA BUSINESS CENTER A8-303, 808, Hongqiao Rd. Shanghai 200030, China
TEL 021-6448-4753 FAX 021-6448-3992
website: www.iai-robot.com

## IAI Robot (Thailand) Co., Ltd.

825, PhairojKijja Tower 12th Floor, Bangna-Trad RD., Bangna, Bangna, Bangkok 10260, Thailand
TEL +66-2-361-4458 FAX +66-2-361-4456




18.08.000