

6 ORCA System Command Set

The ORCA system command set consists of simple two-letter mnemonics, some of which require additional parameters. These commands move the robot, set states that can modify operation of the robot, or allow access to status and error information. These commands are sent directly to the controller through the MDS Dictionary Manager command line.

The general form of the command SYNTAX is:

```
ROBOTSTR$ ("XX parameter[, ]parameter[, ]...")
```

Parameters are separated by a space or an optional comma, and no separators are allowed between the two letters of the mnemonic. Commands are case insensitive.

Default parameters are re-established whenever the robot is ~~initialized~~ through MDS.

Action Commands

ET (Enable Teach mode) activates the teach pendant to move the robot in a rectilinear fashion. ~~The robot speed is dependent on the deflection of the joystick at the last speed specified.~~

NOTE! Wait 2 seconds for the teach pendant to calibrate before attempting to deflect the joystick. A warning message does *not* appear (as in MDS) when this command is executed.

DT (Disable Teach mode) disables the teach pendant.

LO x (LOCate) initializes the robot on side A (x=0) or side B (x= 1). If the parameter is omitted, the arm initializes as if it is on side A.

MA x,y,z,bend,twist,grip (Move Absolute) moves the robot to the absolute position specified by:

x = rail position in cm from the null frame (calibration point)

y = reach position in cm from the null frame (calibration point)

z = height position in cm from the null frame (calibration point)

bend = angle of wrist in degrees from the null frame (calibration point)

twist = angle of pinch rack in degrees from the null frame (calibration point)

grip = distance between fingers in cm

NOTE! All axes must be specified. If any axis is to remain unchanged (in the same position), its current coordinate must be used in the command.

NOTE! [The robot will move at last speed specified.](#)

Example: `ROBOTSTR$ ("MA -10 35 15 - 90 90 3")` will cause the arm to move to the absolute coordinates specified after the MA mnemonic. In other words, the arm will go to a rail position that is 10 cm negative (to the left of) the calibration point, the reach will extend to 35 cm from the calibration point and so on. The robot moves all joints simultaneously so the tool point moves in a straight line.

SD (Shut Down) deactivates the relay that supplies 30 volts to the ORCA bus (powers down the arm).

NOTE! Do not use this command if the robot is holding something. When the arm loses power, it will drop the object.

SF *tw gr* (Set Force) sets the twist torque and grip force. The range is 0–50 and the default value is 50 for both axes. Both parameters must be specified.

NOTE! The command will not yield an error if the twist torque and/or grip force are set to a value greater than 50. However, the force will never exceed 50 (firmware limited). All values greater than 50 are truncated to 50.

Example: `ROBOTSTR$ ("SF 0 50")` will set the twist torque to 0 and the grip force to 50. `ROBOTSTR$ ("SF 50 100")` will set both the twist torque and grip force to 50 (grip force gets truncated)

SS (Set Speed) sets the speed multiplier (1–100) to control the speed of the robot; default setting is 25.

Example: `ROBOTSTR$ ("SS 60")` will set the current robot speed to 60.

SU (Start Up) actuates the relay that supplies 30 volts to the ORCA bus (powers up the arm).

TO *x,y,z* (Tool Offset) sets the current tool offset to the specified values (coordinates) in cm relative to the null tool. This command is similar to the “Send To Robot Now” menu item in the MDS robot module tool editor screen. The tool offset is calculated into the current position of the arm but the arm does not move.

Example: `ROBOTSTR$ ("TO 0 4.15 -3.5")` will download the tool offset coordinates to the controller. The arm will now move about a point 4.15 cm in reach and -3.5 cm in height from the null tool.

Status Commands

When using the **ROBOTSTR\$()** function, responses can be stored in a variable [e.g., `reply$=ROBOTSTR$ ("SR 3")`] which can then be parsed to obtain relevant information not available through an MDS command or function, such as the current tool offset, status of the arm, or wrist torque.

OE (Output Error) returns an error number; if the error bit is set in the status byte, use this command for more information as to the exact error encountered.

OI (Output Identify) returns the installed version of the controller firmware, controller communication processor, and each of the servos [three versions per board: first three for torso board (rail and shoulder), middle three for elbow/wrist board (elbow and wrist), last three for hand board (bend and twist)].

Example: `ROBOTSTR$ ("OI")` might return: `UTB(6.32) COM(0.00F)
SRV(3.07 2.04 1.09 3.07 2.04 1.09 3.07 2.04 1.04).`

OS (Output Status) returns the current status byte as a number between 0 and 255 followed by the arm status, arm mode, robot speed, tool offset, and rail side.

Example: `ROBOTSTR$ ("OS")` might return: `02 ARM(ON) MODE(RDY)
SPD(70) TO(0.00 0.00 0.00) SIDE(A)`

RA (Report Actual position) returns the actual position of each of the six axes and the side of the rail on which the arm is currently operating. The coordinates may not match the commanded positions since each axis has a pre-defined error window. The arm will move to a position within the error window.

Example: `ROBOTSTR$ ("MA 80 25 0 0 0 0")` followed by `ROBOTSTR$ ("RA")` might return:
`79.99 25.00 -0.02 -0.12 0.04 0.00 A.`

RF (Report Force) returns the actual twist torque and grip force, respectively, currently sensed on the twist and grip servos.

Example: `ROBOTSTR$ ("RF")` might return: `3 8.`

RP (Report current Position) returns the coordinates of all the axes from the last command position followed by the side of the rail on which the arm is currently operating.

Example: `ROBOTSTR$ ("MA 80 25 0 0 0 0")` followed by `ROBOTSTR$ ("RP")` would return:
`80.00 25.00 0.00 0.00 0.00 0.00 A.`

RS (Report Servo status) outputs the current status of all six axes:

`RL{status} SH{status} EL{status} BE{status} TW{status}
GR{status}`

If:

status = 00, the servo is operating correctly.

status = 90, communication to that servo has been interrupted.

status = 81, a servo error has occurred.

Example: `ROBOTSTR$ ("RS")` would return: `RL{90} SH{90} EL{90}
BE{90} TW{90} GR{90}` if the ORCA bus cable between the controller and the rail is loose or disconnected.

SR servo# (Servo Request) returns the status of an individual servo (axis) in the following format: error number (in decimal), actual encoder counts, commanded encoder counts, torque drive. Servo# = 0–5: 0=rail, 1=shoulder, 2=elbow, 3=wrists, 4=twist, 5=grip. This command is useful for determining the torque being applied to any axis. A common usage might be to sense the torque on the wrist joint when applying a pipette tip to a pipettor being held by the robot.

Example: `ROBOTSTR$ ("RS 0")` might return: `0 154811 154800 0` which indicates that no error has occurred and a torque of zero has been detected on the rail axis.