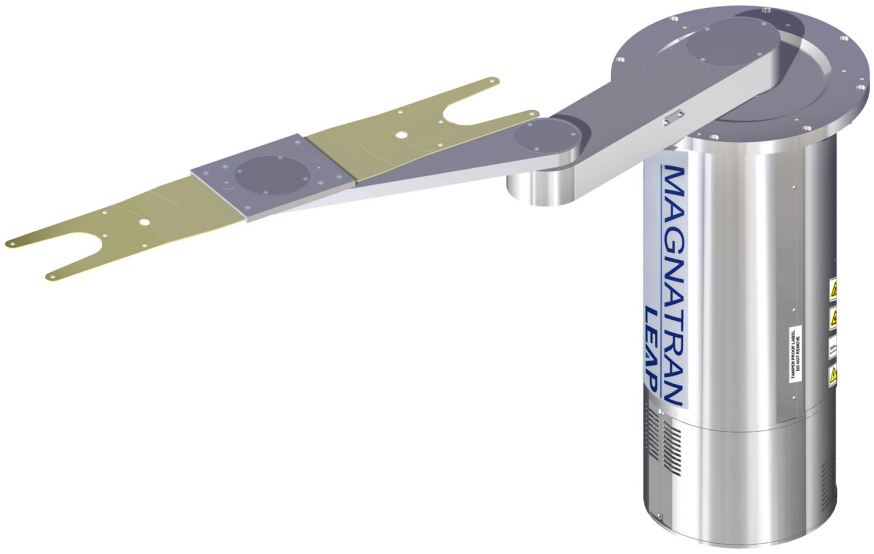


MagnaTran LEAP

Software Manual

Part Number: 605651 Revision A

Brooks Automation, Inc.

Information provided within this document is subject to change without notice, and although believed to be accurate, Brooks Automation, Inc. assumes no responsibility for any errors, omissions, or inaccuracies.

ABF™, AcuLigner™, Advan Tag™, Align™, AutoTeach™, ATR™, AXM™, Basic Blue™, BiSymmetrik™, CenterSmart™, Crate to Operate™, CrossingConnect™, DARTS™, Enerta™, e-RMA™, e-Spares™, e-Volution™, Falcon™, FastRegen™, FIXLOAD™, FrogLeg™, Independent Twin Linear Exchange™, InLigner™, Interface™, Isoport™, ITLX™, Jet™, Jet Engine™, LEAP™, LowProfile™, M2 Nano™, PASIV™, PowerPak™, PerformanceBlue™, Plate Auditor™, PowerPak™, PowerTools™, PuroMaxx™, QuadraFly™, Radius™, Radient™, Radient Express™, Reliance™, Reliance ATR™, RetroEase™, SCARA™, SmartPM™, SPOTLevel™, Sprint™, Synetics™, The New Pathway to Productivity™, Time Optimized Trajectory™, Time Optimal Trajectory™, Time Optimized Path™, TopCooler™, TopLigner™, Ultimate Blue™, VAC-407™, VacuTran™, VersaPort™, and WaferEngine™ are trademarks of Brooks Automation, Inc.

AcuTran®, Asyst®, Crossing Automation®, Fusion®, GOLDLink®, Guardian®, Helix®, Leapfrog®, MagnaTran®, MapTrak®, Marathon®, Marathon 2®, Marathon Express®, Razor®, Spartan®, TrueBlue®, TurboPlus®, Vision®, Zaris®, and the Brooks logo are registered U.S. trademarks of Brooks Automation, Inc.

All other trademarks are properties of their respective owners.

© 2023 Brooks Automation, Inc. All rights reserved. The information included in this manual is proprietary information of Brooks Automation, Inc. and is provided for the use of Brooks customers only and cannot be used for distribution, reproduction, or sale without the express written permission of Brooks Automation, Inc.

This technology is subject to United States export Administration Regulations and authorized to the destination only; diversion contrary to U.S. law is prohibited.

Original manual printed in English. Translation of the original manual from English.

Copyright © 2023, Brooks Automation, Inc.



Corporate Headquarters
15 Elizabeth Drive
Chelmsford, MA 01824 U.S.A.

For Technical Support:

Location	Contact Number	Website
North America	+1-800-447-5007 (Toll Free) +1-978-262-2900 (Local)	http://www.brooks.com/
Europe	+49 800 000 9347 (Toll Free Germany) +49 364 176 9999 6 (Has Toll)	
Japan	+81 120-255-390 (Toll Free) +81 45-330-9005 (Local)	
China	+86 21-5131-7066	
Taiwan	+886 080-003-5556 (Toll Free) +886 3-5525258 (Local)	
Korea	1800-5116 (Toll Free)	
Singapore	+65 1-800-4-276657 (Toll Free) +65 6309 0701 (Local)	

Contact Technical Publications directly: Technical.Publications@brooks.com

Accelerating Innovation

Copyright © 2023, Brooks Automation, Inc.



Brooks Automation, Inc.
15 Elizabeth Drive
Chelmsford, MA
01824-2400
Tel: +1 978-262-2400
Fax: +1 978-262-2500
www.brooks.com

Brooks Locations Worldwide:

Brooks Automation Inc.
46702 Bayside Parkway
Fremont,CA 94538
Tel: +1-510-661-5000
Fax: +1-510-661-5166

Brooks Automation Inc.
AIM Servicios Administrativos
S de RL de CV
Carretera Huinalá km 2.8
Parque Industrial Las Américas
66640 Apodaca, NL Mexico
Tel: +52 81 8863-6363

Brooks Automation (Germany) GmbH
Ernst-Ruska-Ring 11
07745 Jena, Germany
Tel: +49 3641 4821 100
Fax: +49 3641 4821 4100

Brooks Automation (Germany) GmbH
Daimler-Straße 7
78256 Steißlingen, Germany
Tel: +49-7732-9409-0
Fax: +49-7732-9409-200

Brooks Automation, Inc.
9601 Dessau Road, Suite 301
Austin, TX 78754
Tel: +1 512-912-2840
Toll-Free: +1 800-367-4887

Brooks Automation (Israel) Ltd.
Mevo Yerach 5
Kiryat-Gat 82000
Israel
Tel: +972 8672 2988
Fax: +972 8672 2966

Brooks Technology (Shanghai) Limited
2nd Floor, No. 72,
887 Zuchongzhi Road
Zhangjiang Hi-Tech
Park Pudong, Shanghai
China 201203
Tel: +86-21-5131-7070
Fax: +86-21-5131-7068

Brooks Japan K.K.
HEADQUARTERS
Nisso Bldg. No 16, 9F
3-8-8 ShinYokohama, Kohoku-ku
Yokohama, Kanagawa 222-0033
Tel: +81-45-477-5570
Fax: +81-45-477-5571

Brooks Japan K.K.
YOKOHAMA TECHNICAL CENTER
852-1 Kawamuko-cho, Tsuzuki-ku
Yokohama, Kanagawa 224-0044
Tel: +81-45-477-5250
Fax: +81-45-470-6800

Brooks Japan K.K.
KUMAMOTO SERVICE OFFICE
202 Mirai Office II
312-1 Tatsudamachi Yuge
Tatsuda, Kumamoto 861-8002
Tel: +81-96-327-9720
Fax: +81-96-327-9721

Brooks CCS Japan K.K.
CONTAMINATION CONTROL SOLUTIONS
Nisso Bldg. No 16, 9F
3-8-8 ShinYokohama, Kohoku-ku
Yokohama, Kanagawa 222-0033
Tel: +81-45-477-5570
Fax: +81-45-477-5571

Brooks Automation Ltd.
TAIWANHEADQUARTERS
5F-5, No.32, Tai-Yuen Street
Chu-Pei City
Hsinchu County 302, Taiwan, R.O.C.
Tel: +886-3-552 5258
Fax (G&A): +886-3-552 5255
Fax (Sales): +886-3-552 5200

Brooks Automation Korea, Inc.
#35 Giheungdanji-Ro 121Beon-Gil
Giheung-Gu, Yongin-Si
Gyeonggi-Do, 17086
Korea
Tel : +82-31-288-2500
Fax: +82-31-287-2111

Brooks Automation CCS RS AG
Lohstampfestrasse 11
CH-8274 Tagerwilen, Switzerland
Tel: + 41 71-666-72-10
Fax: + 41 71-666-72-11

Brooks Automation Korea, Inc.
#35 Giheungdanji-Ro 121Beon-Gil
Giheung-Gu, Yongin-Si
Gyeonggi-Do, 17086
Korea
Tel : +82-31-288-2500
Fax: +82-31-287-2111

Brooks Automation (S) Pte Ltd
51-18-C1 Menara BHL,
57 Jalan Ahmad Shah,
10050, Penang,
Malaysia
Tel: +60 4 3701012
Fax: +60 4 3701015

Brooks Automation (Singapore) Pte Ltd
Blk 5008 Ang Mo Kio Avenue 5
#05-08, Techplace II
Singapore 569874
Tel: +65-6836-3168
Fax: +65-6836-3177

Brooks Automation Ltd.
TAINAN OFFICE
3F., No.11, Nanke 3rd Rd., Xinshi Dist.
Tainan Science Park
Tainan City 74147, Taiwan (R.O.C.)
TEL: +886-6-505-0268
FAX: +886-6-505-5228

Brooks Automation Precise Collaborative Robotics
201 Lindbergh Drive
Livermore, CA 94551
Tel: +1-978-262-2400

Copyright © 2023, Brooks Automation, Inc.

Revision History

Revision	ECO	Date	Action	Author
A	EC145579	5/30/2023	Initial Release	K. Forscher

Copyright © 2023, Brooks Automation, Inc.

1. Safety

DANGER

Read the Safety Chapter

Failure to review the *Safety* chapter and follow the safety warnings can result in death or serious injury.

- All personnel involved with the operation or maintenance of this product must read and understand the information in this safety chapter.
- Follow all applicable safety codes of the facility as well as national and international safety codes.
- Know the facility safety procedures, safety equipment, and contact information.
- Read and understand each procedure before performing it.



Refer to the Safety Chapter of the *MagnaTran LEAP User Manual*.

Copyright © 2023, Brooks Automation, Inc.

2. Communication

Set/Request/Store Communication

The communication command sets, requests, and stores the serial communications and command execution modes. The communication command sets the specified serial I/O configuration in RAM only. A description of the various communication settings are provided below.The factory set values for communication are listed on the QR.

M/B

The robot provides two modes for main communications with the robot. There is a “user friendly” mode referred to as “Monitor Mode” and a “computer friendly” mode referred to as “Packet Mode”. Sequential or background operation can be selected when using either communications mode.

Monitor Mode

Monitor mode is a “user friendly” communications mode. All responses from the robot are descriptive and easy to understand. This mode is best used when communicating with the robot through a terminal and is recognized by the “:” prompt. A typical sequence of communications in Monitor Mode appears below.

```
:RQ COMM M/B FLOW LF ECHO
COMM
M/B --------------- MON
FLOW ------------- SEQ
LF ---------------- -ON
ECHO -------------- ON
:
```

Errors are reported as complete error messages and status messages.

Packet Mode

Packet mode is a computer-based communications mode. All responses from the robot are short with minimal descriptive information provided. This mode is best used when a host controller is

Copyright © 2023, Brooks Automation, Inc.

communicating with the robot and is recognized by the "_RDY" prompt. A typical sequence of communications in Packet Mode appears below.

```
_RDY
RQ COMM M/B FLOW LF ECHO
COMM MON PKT BKG ON ON
_RDY
```

Errors are reported as codes without associated messages.

Packet Mode does not always respond in the order in which requested.

Copyright © 2023, Brooks Automation, Inc.

FLOW

The basic software command sequence consists of an interplay between Commands from the Host Controller to the robot and Responses from the robot to the Host over the main communications line. Software communications may be configured in one of four modes of interaction: Sequential, Background, Background Plus, and Multi-mode. In Sequential Mode, the software commands and responses occur one at a time. In the Background Modes, certain types of commands may proceed in the “background” while other types of commands may be processed in the “foreground”.

Sequential Mode

In sequential mode, the robot executes the command completely before returning a READY signal indicating that the robot is ready for another command. This mode allows execution of only one command at a time. A typical sequence of communications in Sequential Mode appears below.

- The Host sends a command string to the robot.
- The robot responds with information if the command was a Request, or with an Error string if the command is incorrect or an error occurs during processing.
- The robot sends a READY string to the Host, regardless of whether an error has occurred.

> ***NOTE:*** In Sequential Mode, the Host should not attempt to send another command string until it receives the READY string from the robot.

Background Mode

In Background (BKG) mode, for certain commands, the robot will return a READY string immediately after it has received the command and typically before the command has been completed. This command task is then placed in the “background” and other “foreground” commands may be executed sequentially while the background command is in progress.

Only certain commands have been assigned to the “background” and “foreground” categories. All Action commands can be placed in the “background”; all Request commands and the HALT command can be executed in the “foreground” while a command is executing in the “background”. A typical sequence of communications in Background (BKG) Mode appears below.

a. Host sends an Action (motion) command to the robot.
b. The robot sends a READY signal immediately, while beginning the requested action.
c. Host requests information.
d. Robot returns information.
e. Robot completes background action.
f. Host requests operational status.

Copyright © 2023, Brooks Automation, Inc.

g. Robot sends message that the Action operation has been completed, including an error code if an error occurred during the operation.

NOTE: Background tasks do not stack. If the Host Controller sends a second background command before the first background command is complete, the robot sends an error message. Background Mode should not be used if the robot is installed in the Brooks Automation FabExpress system or if the robot is attached to a peripheral device. This mode is mainly used in stand-along applications.

While a command is in background, foreground commands are handled in a normal, sequential manner. To determine when the background command is complete, use the RQ BG command. This command returns the busy status of the background command (Y|N) and (if RDY) any errors that may have occurred.

Background Plus Mode

The Background Plus (BKG+) mode works exactly like the background mode except for one addition: when the action command is done, the prompt _BKGRDY is returned. If an error occurred during the background action command, then a _BKGERR response with the error number or error string (depending on packet or monitor mode) is returned along with the prompt _BKGRDY on the next line. Thus the robot does not need to be polled with RQ BG to determine if an action has been completed. A CDM warning will be displayed as _ERR instead of _BKGERR.

```
 _BKGRDY
 goto n 1
_RDY
_BKGERR #### (if an error occurs)
_BKGRDY (when the action completes)
```

NOTE: Background Plus (BKG+) tasks do not stack. If the Host Controller sends a second background command before the first background command is complete, the robot sends an error message. Background Plus (BKG+) Mode should not be used if the robot is installed in the Brooks Automation FabExpress system or if the robot is to be attached to peripheral device. This mode is mainly used in stand-along applications:. (MultiTran ®5, VacuTran™ 5, MagnaTran 7)

Copyright © 2023, Brooks Automation, Inc.

Multi-Mode

Multi-Mode allows multiple actions to be executed simultaneously. When in Multi-Mode, all commands are treated equally and up to nine action commands can be executed simultaneously as long as they are not mutually exclusive. As each command is issued, it is acknowledged and assigned a two-digit tracking number. When that command is complete, a _DONE message is returned with the same tracking number. If an error occurs, it will also be reported with the same tracking number. Tracking numbers go to 99 then begin again at 01 (00 is reserved for boot-up or Ctrl-C errors). The controller will verify that each command can be performed without conflict. For example, a PICK cannot be performed at a station if the cassette is moving.
At power-up, the following prompt will be displayed:

```
 _DONE 00
```

The following sequence is an example of the command flow:

```
02 ldpt a load
_ack 02
01 home all
_ack 01
_done 01
_done 02

00 LDPT A LOAD
_ACKERR 659

1 LDPT A LOAD
_ACKERR 659

LDPT A LOAD
_ACKERR 659

100 LDPT A LOAD
_ACKERR 659

13 HOME ALL
_ACK 13
<CTRL>C
_ACK 00
_DONE 00
_ERR 13 603

11 HOME ALL
_ACK 11
11 RQ CONFIG
_ACKERR 660
12 RQ CONFIG
_RSP 12 ROBOT CONFIG:SHAFT 7Z
_DONE 12
_DONE 11
```

Copyright © 2023, Brooks Automation, Inc.

```
02 LDPT A LOAD
_ACK 02
03 LDPT B LOAD
_ACK 03
04 LDPT C LOAD
_ACK 04
05 LDPT D LOAD
_ACK 05
06 HOME ALL
_ACKERR 661
_DONE 03
_DONE 05
_DONE 02
_DONE 0411 HOME ALL
_ACK 11
11 RQ CONFIG
_ACKERR 660
12 RQ CONFIG
_RSP 12 ROBOT CONFIG:SHAFT 7Z
_DONE 12
_DONE 11

02 LDPT A LOAD
_ACK 02
03 LDPT B LOAD
_ACK 03
04 LDPT C LOAD
_ACK 04
05 LDPT D LOAD
_ACK 05
06 HOME ALL
_ACKERR 661
_DONE 03
_DONE 05
_DONE 02
_DONE 04
```


Copyright © 2023, Brooks Automation, Inc.

LINEFEED

Specifies the linefeed execution mode as enabled or disabled. Linefeed mode is set to disabled by default.

CHECKSUM

The checksum option sets the communications to checksum for each received command and can be used to validate communication. The checksum is calculated for each character of the command string. There should be no space between the last character of the command string and the checksum value. The checksum is case-sensitive. The response similarly is supplied with a checksum. The checksum algorithm starts with a zero value and adds the ASCII value of each character of the string to be checked. During the summation process, any overflow over 255 is ignored. The checksum option sets the communications to checksum each received command and can be used to validate communication. The checksum is calculated for each character of the command string. There should be no space between the last character of the command string and the checksum value. The checksum is case-sensitive. The response similarly is supplied with a checksum. The checksum algorithm starts with a zero value and adds the ASCII value of each character of the string to be checked. During the summation process, any overflow over 255 is ignored.

CHECKSUM CALCULATION

To calculate the checksum for a LEAP command, it is helpful to have a hexadecimal calculator. One option is to use the calculator provided by Windows which has a hexadecimal mode. Note that the Windows calculator defaults to decimal mode and you must chooses hexadecimal mode under the *View* menu pick. You can tell that hex mode has been selected when you can see calculator keys marked “A” through “E” on the keypad as shown:

Copyright © 2023, Brooks Automation, Inc.



The second item required is an ASCII character chart that shows the equivalent hexadecimal value of each ASCII character. The table shows the hexadecimal value of the character ‘S’ to be 5x + 3 = 0x53. The table is shown below:

	2x	3x	4x	5x	6x	7x
0	SP	0	@	P	'	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7		7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[	k	{
C	,	<	L	\	l	|
D	-	=	M	]	m	}
E	.	>	N	^	n	~
F	/	?	O	_	o	

Copyright © 2023, Brooks Automation, Inc.

Here is an example of calculating the checksum for a LEAP command:

```
SET COMM CHECKSUM OFF
```

Take each character in the command string and find its ASCII equivalent in hexadecimal. Space characters (‘ ‘) count and have a hex value of 0x20. Add up the hex values of the command string as shown:

S	0x53
E	0x45
T	0x54
' '	0x20
C	0x43
O	0x4F
M	0x4D
M	0x4D
' '	0x20
C	0x43
H	0x48
E	0x45
C	0x43
K	0x4B
S	0x53
U	0x55
M	0x4D
' '	0x20
O	0x4F
F	0x46
F	0x46
0x5A6 = Checksum	

Copyright © 2023, Brooks Automation, Inc.

The final step is to truncate the checksum to the lowest two digits. In our example, the usable part of the checksum changes from 0x5A6 to just 0xA6. The protocol also accepts this in lowercase, so the final form can also be displayed as: 0xa6.

Put this together with the command string so that the LEAP robot will accept it:

```
SET COMM CHECKSUM OFFa6
```

> ***NOTE:*** That there is no intervening space between the command string and the checksum value.

DATA REPORTING

Automatic data reporting applies to the robot's movement in the station coordinates. In AUT mode, the robot automatically reports the station position for each PICK, PLACE or XFER command. The condition for the auto response generation is a change in the state of the robot arm location (in terms of station coordinates) during the commands.

The AUT response returns a string consisting of 14 characters (including spaces) on every change in state of the robot arm location in terms of station coordinates as follows:

EX|RE stn ssss slot UP|DN

where,
EX|RE= EX (Extend) or RE (Retract)
stn= decimal station number
ssss= slot number
UP|DN= Z location

Set Communication	
Command Syntax	`SET COMM [M/B (MON|PKT)] [FLOW (SEQ|BKG|BKG+|MULTI)] [LF (ON|OFF)] [ECHO (ON|OFF)] [CHECKSUM (ON|OFF)][ERRLVL err_level] [DREP (AUT|REQ)] [ERR_REP (AUT|REQ)] [RR (ON|OFF)]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Communication	
FLOW:	Specifies the command execution type. `SEQ`: Sequential mode `BKG`: Background mode `BKG+`: Background Plus mode `MULTI`: Multiple action mode ***NOTE:*** *BKG and BKG+ are for backwards compatibility. These options were replaced with MULTI mode.*
LINEFEED:	Specifies the linefeed state. `ON`: Linefeed enabled. `OFF`: Linefeed disabled (default).
ECHO:	Specifies the ECHO state. `ON`: Echo enabled. `OFF`: Echo disabled. ***NOTE:*** *No checksum is displayed when echo is enabled.*
CHECKSUM:	Specifies the checksum option state. `ON`: Checksum enabled. `OFF`: Checksum disabled.
ERR_LEVEL:	`err_level`: Specifies the level of reported messages. Least number of messages = 1 Maximum number of messages = 5 Default = 2
DREP:	Specifies the automatic station position data reporting flag. `AUT`: Automatic mode. Automatic station position data reporting is enabled. In this mode the robot automatically reports the station position for each section of a PICK, PLACE or XFER command. The auto response returns a string of 14 characters on every change in state of the robot arm location in terms of station coordinates. The data string consists of the R location, followed by the station number, followed by the slot number, followed by the Z location. `REQ`: Request mode. Automatic station position data reporting is disabled.
ERR_REP:	Specifies the error reporting state. `AUT`: Automatic reporting of error. `REQ`: No reporting of errors.
RR:	Response resend. `ON`: When the ^E character is sent, the host re-transmits all data characters that were sent to the host after the last command was received, including the prompts. If the checksum mode is ON, the response also includes checksum information. `OFF`: No re-transmission of data after the ^E character is entered.
Compatibility	No known difference with Wave II Software.

Copyright © 2023, Brooks Automation, Inc.

Set Communication	

Example	The following example sets the command execution mode in RAM to Background.
	`SET COMM FLOW BKG`
	
	The following example sets both the serial I/O communications mode and the command execution mode using the ALL specifier.
	`SET COMM ALL MON BKG`
	
	The following example displays the Automatic Data Reporting string after a PICK 1 is performed:
	`PICK 1`
	`RE 01 0001 DN`
	`EX 01 0001 DN`
	`EX 01 0001 UP`
	`RE 01 0001 UP`

Request Communication	
Command Syntax	`RQ COMM [ALL]|` `[(M/B|FLOW|LF|ECHO|CHECKSUM|ERRLVL|DREP|ERR_REP)] [RR (ON|OFF)]`
Response Syntax	`COMM` *(M/B|FLOW|LF|ECHO|CHECKSUM|DREP|ERRLVL|BAUD RATE|ERR_REP|RR)* ***NOTE:*** *Baud Rate cannot be Set, Requested, or Stored using serial commands.*

Request Communication	
M/B:	Specifies the serial I/O communications mode. `MON`: Monitor mode `PKT`: Packet mode (_RDY)
FLOW:	Requests the command execution type. `SEQ`: Sequential mode. `BKG`: Background mode. `BKG+`: Background Plus mode. `MULTI`: Multiple action mode. ***NOTE:*** *BKG and BKG+ are for backwards compatibility. These options were replaced with MULTI mode.*
LINEFEED:	Requests the linefeed state. `ON`: Linefeed enabled. `OFF`: Linefeed disabled (default).

Copyright © 2023, Brooks Automation, Inc.

Request Communication	
ECHO:	Requests the ECHO state. `ON`: Echo enabled. `OFF`: Echo disabled. ***NOTE:*** *No checksum is displayed when echo is enabled.*
CHECKSUM:	Requests the checksum option state. `ON`: Checksum enabled. `OFF`: Checksum disabled.
ERR_LEVEL:	`err_level`: Requests the level of reported messages. Least number of messages = 1 Maximum number of messages = 5 Default = 2
DREP:	Requests the automatic station position data reporting flag. `AUT`: Automatic mode. Automatic station position data reporting is enabled. In this mode the robot automatically reports the station position for each section of a PICK, PLACE or XFER command. The auto response returns a string of 14 characters on every change in state of the robot arm location in terms of station coordinates. The data string consists of the R location, followed by the station number, followed by the slot number, followed by the Z location. `REQ`: Request mode. Automatic station position data reporting is disabled.
ERR_REP:	Requests the error reporting state. `AUT`: Automatic reporting of error. `REQ`: No reporting of errors.
RR:	Response resend. `ON`: When the ^E character is sent, the host re-transmits all data characters that were sent to the host after the last command was received, including the prompts. If the checksum mode is ON, the response also includes checksum information. `OFF`: No re-transmission of data after the ^E character is entered.
Compatibility	No known difference with Wave II Software.
Example	`RQ COMM ALL` `COMM PKT SEQ OFF ON OFF AUT 2 9600 AUT OFF`

Store Communication	
Command Syntax	`STORE COMM [ALL]|` `[(M/B|FLOW|LF|ECHO|CHECKSUM|ERRLVL|DREP|ERR_REP)] [RR (ON|OFF)]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Communication	
M/B:	Specifies the serial I/O communications mode. `MON`: Monitor mode `PKT`: Packet mode (_RDY)
FLOW:	Stores the command execution type. `SEQ`: Sequential mode. `BKG`: Background mode. `BKG+`: Background Plus mode. `MULTI`: Multiple action mode. ***NOTE:*** *BKG and BKG+ are for backwards compatibility. These options were replaced with MULTI mode.*
LINEFEED:	Stores the linefeed state. `ON`: Linefeed enabled. `OFF`: Linefeed disabled (default).
ECHO:	Stores the ECHO state. `ON`: Echo enabled. `OFF`: Echo disabled. ***NOTE:*** *No checksum is displayed when echo is enabled.*
CHECKSUM:	Stores the checksum option state. `ON`: Checksum enabled. `OFF`: Checksum disabled.
ERR_LEVEL:	`err_level`: Stores the level of reported messages. Least number of messages = 1 Maximum number of messages = 5 Default = 2
DREP:	Stores the automatic station position data reporting flag. `AUT`: Automatic mode. Automatic station position data reporting is enabled. In this mode the robot automatically reports the station position for each section of a PICK, PLACE or XFER command. The auto response returns a string of 14 characters on every change in state of the robot arm location in terms of station coordinates. The data string consists of the R location, followed by the station number, followed by the slot number, followed by the Z location. `REQ`: Request mode. Automatic station position data reporting is disabled.
ERR_REP:	Stores the error reporting state. `AUT`: Automatic reporting of error. `REQ`: No reporting of errors.
RR:	Response resend. `ON`: When the ^E character is sent, the host re-transmits all data characters that were sent to the host after the last command was received, including the prompts. If the checksum mode is ON, the response also includes checksum information. `OFF`: No re-transmission of data after the ^E character is entered.
Compatibility	No known difference with WAVE II Software.

Copyright © 2023, Brooks Automation, Inc.

Store Communication	

Example	The following example stores the current serial I/O communications mode in non-volatile memory.
	
	`STORE COMM M/B`
	
	The following example stores the current command execution mode in non-volatile memory.
	
	`STORE COMM FLOW`

Set/Request/Store Communication

The communication command sets, requests, and stores the serial communications and command execution modes. The communication command sets the specified serial I/O configuration in RAM only. A description of the various communication settings are provided below.
The factory set values for communication are listed on the QR.

M/B

The robot provides two modes for main communications with the robot. There is a “user friendly” mode referred to as “Monitor Mode” and a “computer friendly” mode referred to as “Packet Mode”. Sequential or background operation can be selected when using either communications mode.

Monitor Mode

Monitor mode is a “user friendly” communications mode. All responses from the robot are descriptive and easy to understand. This mode is best used when communicating with the robot through a terminal and is recognized by the “:” prompt. A typical sequence of communications in Monitor Mode appears below.

```
:RQ COMM M/B FLOW LF ECHO
COMM
M/B --------------- MON
FLOW ------------- SEQ
LF ---------------- -ON
ECHO -------------- ON
:
```

Errors are reported as complete error messages and status messages.

Packet Mode

Packet mode is a computer-based communications mode. All responses from the robot are short with minimal descriptive information provided. This mode is best used when a host controller is

Copyright © 2023, Brooks Automation, Inc.

communicating with the robot and is recognized by the "_RDY" prompt. A typical sequence of communications in Packet Mode appears below.

```
_RDY
RQ COMM M/B FLOW LF ECHO
COMM MON PKT BKG ON ON
_RDY
```

Errors are reported as codes without associated messages.

Packet Mode does not always respond in the order in which requested.

FLOW

The basic software command sequence consists of an interplay between Commands from the Host Controller to the robot and Responses from the robot to the Host over the main communications line. Software communications may be configured in one of five modes of interaction, Sequential, Background, Background Plus, Multi-mode, and Multi-Device. In Sequential Mode, the software commands and responses occur one at a time. In the Background Modes, certain types of commands may proceed in the "background" while other types of commands may be processed in the "foreground".

Sequential Mode

In sequential mode, the robot executes the command completely before returning a READY signal indicating that the robot is ready for another command. This mode allows execution of only one command at a time. A typical sequence of communications in Sequential Mode appears below.

- The Host sends a command string to the robot.
- The robot responds with information if the command was a Request, or with an Error string if the command is incorrect or an error occurs during processing.
- The robot sends a READY string to the Host, regardless of whether an error has occurred.

NOTE: In Sequential Mode, the Host should not attempt to send another command string until it receives the READY string from the robot.

Background Mode

In Background mode, for certain commands, the robot will return a READY string immediately after it has received the command and typically before the command has been completed. This command task is then placed in the "background" and other "foreground" commands may be executed sequentially while the background command is in progress.

Only certain commands have been assigned to the "background" and "foreground" categories. All Action commands can be placed in the "background"; all Request commands and the HALT command can be executed in the "foreground" while a command is executing in the "background". A typical sequence of communications in Background Mode appears below.

Copyright © 2023, Brooks Automation, Inc.

a. Host sends an Action (motion) command to the robot.
b. The robot sends a READY signal immediately, while beginning the requested action.
c. Host requests information.
d. Robot returns information.
e. Robot completes background action.
f. Host requests operational status.
g. Robot sends message that the Action operation has been completed, including an error code if an error occurred during the operation.

NOTE: Background tasks do not stack. If the Host Controller sends a second background command before the first background command is complete, the robot sends an error message. Background Mode should not be used if the robot is installed in the Brooks Automation FabExpress system or if the robot is attached to a peripheral device. This mode is mainly used in stand-along applications.

While a command is in background, foreground commands are handled in a normal, sequential manner. To determine when the background command is complete, use the RQ BG command. This command returns the busy status of the background command (Y|N) and (if RDY) any errors that may have occurred.

Background Plus Mode

The Background Plus (BKG+) mode works exactly like the background mode except for one addition: when the action command is done, the prompt _BKGRDY is returned. If an error occurred during the background action command, then a _BKGERR response with the error number or error string (depending on packet or monitor mode) is returned along with the prompt _BKGRDY on the next line. Thus the robot does not need to be polled with RQ BG to determine if an action has been completed. A CDM warning will be displayed as _ERR instead of _BKGERR.

e.g.

```
 _BKGRDY
 goto n 1
_RDY
_BKGERR #### (if an error occurs)
_BKGRDY (when the action completes)
```

NOTE: Background Plus tasks do not stack. If the Host Controller sends a second background command before the first background command is complete, the robot sends an error message. This mode is mainly used in stand-alone applications.

Multi-Mode

Copyright © 2023, Brooks Automation, Inc.

Multi-Mode allows multiple actions to be executed simultaneously. When in Multi-Mode, all commands are treated equally and up to nine action commands can be executed simultaneously as long as they are not mutually exclusive. As each command is issued, it is acknowledged and assigned a two-digit tracking number. When that command is complete, a _DONE message is returned with the same tracking number. If an error occurs, it will also be reported with the same tracking number. Tracking numbers go to 99 then begin again at 01 (00 is reserved for boot-up or Ctrl-C errors). The controller will verify that each command can be performed without conflict. For example, a PICK cannot be performed at a station if the cassette is moving.
At power-up, the following prompt will be displayed:

```
 _DONE 00
```

The following sequence is an example of the command flow:

```
02 ldpt a load
_ack 02
01 home all
_ack 01
_done 01
_done 02

00 LDPT A LOAD
_ACKERR 659

1 LDPT A LOAD
_ACKERR 659

LDPT A LOAD
_ACKERR 659

100 LDPT A LOAD
_ACKERR 659

13 HOME ALL
_ACK 13
<CTRL>C
_ACK 00
_DONE 00
_ERR 13 603

11 HOME ALL
_ACK 11
11 RQ CONFIG
_ACKERR 660
12 RQ CONFIG
_RSP 12 ROBOT CONFIG:SHAFT 7Z
_DONE 12
_DONE 11

02 LDPT A LOAD
```


Copyright © 2023, Brooks Automation, Inc.

```
_ACK 02
03 LDPT B LOAD
_ACK 03
04 LDPT C LOAD
_ACK 04
05 LDPT D LOAD
_ACK 05
06 HOME ALL
_ACKERR 661
_DONE 03
_DONE 05
_DONE 02
_DONE 0411 HOME ALL
_ACK 11
11 RQ CONFIG
_ACKERR 660
12 RQ CONFIG
_RSP 12 ROBOT CONFIG:SHAFT 7Z
_DONE 12
_DONE 11

02 LDPT A LOAD
_ACK 02
03 LDPT B LOAD
_ACK 03
04 LDPT C LOAD
_ACK 04
05 LDPT D LOAD
_ACK 05
06 HOME ALL
_ACKERR 661
_DONE 03
_DONE 05
_DONE 02
_DONE 04
```

Multi-Device Mode

Multi-Device mode is similar to Multi-Mode in that commands are associated with tracking numbers or process identifiers. In Multi-Device mode, the host will also specify which device is to execute the command by including a device identifier. Multi-Device mode allows the host controller to route commands asynchronously to any device by setting up a device table. The device table is used by the controller to determine which link to use to communicate with the device. As with Multi-Mode, Multi-Device responses, errors, and prompts are returned at the completion of commands.

The following example demonstrates the normal responses after a command is issued. 01 01 HOME ALL (<tracking number> <device ID> <command>)
_ACK 01 (acknowledge <tracking number>)
_DONE 01 (status <tracking number>)

Copyright © 2023, Brooks Automation, Inc.

In the following example, a command is sent to device 2 while device 1 is still performing a command:

```
01 01 HOME ALL
_ACK 01
02 02 RQ CONFIG
_ACK 02
RSP 02 Robot Confide: Shaft 7z
_DONE 02
_DONE 01
```

A process identifier can only be used by one command at a time. When the host issues the next command, it will use a separate identifier even if a different device is involved.

LINEFEED

Specifies the linefeed execution mode as enabled or disabled. Linefeed mode is set to disabled by default.

Copyright © 2023, Brooks Automation, Inc.

CHECKSUM

The checksum option sets the communications to checksum for each received command and can be used to validate communication. The checksum is calculated for each character of the command string. There should be no space between the last character of the command string and the checksum value. The checksum is case-sensitive. The response similarly is supplied with a checksum. The checksum algorithm starts with a zero value and adds the ASCII value of each character of the string to be checked. During the summation process, any overflow over 255 is ignored. The checksum option sets the communications to checksum each received command and can be used to validate communication. The checksum is calculated for each character of the command string. There should be no space between the last character of the command string and the checksum value. The checksum is case-sensitive. The response similarly is supplied with a checksum. The checksum algorithm starts with a zero value and adds the ASCII value of each character of the string to be checked. During the summation process, any overflow over 255 is ignored.

CHECKSUM CALCULATION

To calculate the checksum for a LEAP command, it is helpful to have a hexadecimal calculator. One option is to use the calculator provided by Windows which has a hexadecimal mode. Note that the Windows calculator defaults to decimal mode and you must chooses hexadecimal mode under the *View* menu pick. You can tell that hex mode has been selected when you can see calculator keys marked “A” through “E” on the keypad as shown:



Copyright © 2023, Brooks Automation, Inc.

The second item required is an ASCII character chart that shows the equivalent hexadecimal value of each ASCII character. The table shows the hexadecimal value of the character ‘S’ to be 5x + 3 = 0x53. The table is shown below:

	2x	3x	4x	5x	6x	7x
0	SP	0	@	P	'	p
1	!	1	A	Q	a	q
2	"	2	B	R	b	r
3	#	3	C	S	c	s
4	$	4	D	T	d	t
5	%	5	E	U	e	u
6	&	6	F	V	f	v
7		7	G	W	g	w
8	(	8	H	X	h	x
9	)	9	I	Y	i	y
A	*	:	J	Z	j	z
B	+	;	K	[	k	{
C	,	<	L	\	l	|
D	-	=	M	]	m	}
E	.	>	N	^	n	~
F	/	?	O	_	o	

Copyright © 2023, Brooks Automation, Inc.

Here is an example of calculating the checksum for a LEAP command:

```
SET COMM CHECKSUM OFF
```

Take each character in the command string and find its ASCII equivalent in hexadecimal. Space characters (‘ ‘) count and have a hex value of 0x20. Add up the hex values of the command string as shown:

S	0x53
E	0x45
T	0x54
' '	0x20
C	0x43
O	0x4F
M	0x4D
M	0x4D
' '	0x20
C	0x43
H	0x48
E	0x45
C	0x43
K	0x4B
S	0x53
U	0x55
M	0x4D
' '	0x20
O	0x4F
F	0x46
F	0x46
0x5A6 = Checksum	

Copyright © 2023, Brooks Automation, Inc.

The final step is to truncate the checksum to the lowest two digits. In our example, the usable part of the checksum changes from 0x5A6 to just 0xA6. The protocol also accepts this in lowercase, so the final form can also be displayed as: 0xa6.

Put this together with the command string so that the LEAP robot will accept it:

```
SET COMM CHECKSUM OFFa6
```

> ***NOTE:*** That there is no intervening space between the command string and the checksum value.

DATA REPORTING

Automatic data reporting applies to the robot's movement in the station coordinates. In AUT mode, the robot automatically reports the station position for each PICK, PLACE or XFER command. The condition for the auto response generation is a change in the state of the robot arm location (in terms of station coordinates) during the commands.

The AUT response returns a string consisting of 14 characters (including spaces) on every change in state of the robot arm location in terms of station coordinates as follows:

EX|RE stn ssss slot UP|DN

where,
EX|RE= EX (Extend) or RE (Retract)
stn= decimal station number
ssss= slot number
UP|DN= Z location

Set Communication	
Command Syntax	`SET COMM [M/B (MON|PKT)] [FLOW (SEQ|BKG|BKG+|MULTI|MULTI_DEV)] [LF (ON|OFF)] [ECHO (ON|OFF)] [CHECKSUM (ON|OFF)][ERRLVL err_level] [DREP (AUT|REQ)] [ERR_REP (AUT|REQ)] [RR (ON|OFF)]`
Response Syntax	There is no response from this command.

Copyright © 2023, Brooks Automation, Inc.

Set Communication	
FLOW:	Specifies the command execution type. `SEQ`: Sequential mode `BKG`: Background mode `BKG+`: Background Plus mode `MULTI`: Multiple action mode `MULTI_DEV`: Multiple Device mode ***NOTE:*** *BKG and BKG+ are for backwards compatibility. These options were replaced with MULTI mode.*
LINEFEED:	Specifies the linefeed state. `ON`: Linefeed enabled. `OFF`: Linefeed disabled (default).
ECHO:	Specifies the ECHO state. `ON`: Echo enabled. `OFF`: Echo disabled. ***NOTE:*** *No checksum is displayed when echo is enabled.*
CHECKSUM:	Specifies the checksum option state. `ON`: Checksum enabled. `OFF`: Checksum disabled.
ERR_LEVEL:	`err_level`: Specifies the level of reported messages. Least number of messages = 1 Maximum number of messages = 5 Default = 2
DREP:	Specifies the automatic station position data reporting flag. `AUT`: Automatic mode. Automatic station position data reporting is enabled. In this mode the robot automatically reports the station position for each section of a PICK, PLACE or XFER command. The auto response returns a string of 14 characters on every change in state of the robot arm location in terms of station coordinates. The data string consists of the R location, followed by the station number, followed by the slot number, followed by the Z location. `REQ`: Request mode. Automatic station position data reporting is disabled.
ERR_REP:	Specifies the error reporting state. `AUT`: Automatic reporting of error. `REQ`: No reporting of errors (errors can be retrieved using "Request History" on page 54).
RR:	Response resend. `ON`: When the ^E character is sent, the host re-transmits all data characters that were sent to the host after the last command was received, including the prompts. If the checksum mode is ON, the response also includes checksum information. `OFF`: No re-transmission of data after the ^E character is entered.
Supported Devices	SEQ, MULTI, and MULTI_DEV are used with the FabExpress. BKG and BKG+ are NOT used with the Reliance ATR in the FabExpress system.
Compatibility	Compatible with Wave II Software/No known difference with Wave II Software.

Copyright © 2023, Brooks Automation, Inc.

Set Communication	

Example	The following example sets the command execution mode in RAM to Background.
	`SET COMM FLOW BKG`
	
	The following example sets both the serial I/O communications mode and the command execution mode using the ALL specifier.
	`SET COMM ALL MON BKG`
	
	The following example displays the Automatic Data Reporting string after a PICK 1 is performed:
	`PICK 1`
	`RE 01 0001 DN`
	`EX 01 0001 DN`
	`EX 01 0001 UP`
	`RE 01 0001 UP`

Request Communication	
Command Syntax	`RQ COMM [ALL]|` `[(M/B|FLOW|LF|ECHO|CHECKSUM|ERRLVL|DREP|ERR_REP)] [RR (ON|OFF)]`
Response Syntax	`COMM` *(M/B|FLOW|LF|ECHO|CHECKSUM|DREP|ERRLVL|BAUD RATE|ERR_REP|RR)* ***NOTE:*** *Baud Rate cannot be Set, Requested, or Stored using serial commands.*

Request Communication	
M/B:	Specifies the serial I/O communications mode. `MON`: Monitor mode `PKT`: Packet mode (_RDY)
FLOW:	Requests the command execution type. `SEQ`: Sequential mode. `BKG`: Background mode. `BKG+`: Background Plus mode. `MULTI`: Multiple action mode. `MULTI_DEV`: Multiple Device mode. ***NOTE:*** *BKG and BKG+ are for backwards compatibility. These options were replaced with MULTI mode.*
LINEFEED:	Requests the linefeed state. `ON`: Linefeed enabled. `OFF`: Linefeed disabled (default).

Copyright © 2023, Brooks Automation, Inc.

Request Communication	
ECHO:	Requests the ECHO state. `ON`: Echo enabled. `OFF`: Echo disabled. ***NOTE:*** *No checksum is displayed when echo is enabled.*
CHECKSUM:	Requests the checksum option state. `ON`: Checksum enabled. `OFF`: Checksum disabled.
ERR_LEVEL:	`err_level`: Requests the level of reported messages. Least number of messages = 1 Maximum number of messages = 5 Default = 2
DREP:	Requests the automatic station position data reporting flag. `AUT`: Automatic mode. Automatic station position data reporting is enabled. In this mode the robot automatically reports the station position for each section of a PICK, PLACE or XFER command. The auto response returns a string of 14 characters on every change in state of the robot arm location in terms of station coordinates. The data string consists of the R location, followed by the station number, followed by the slot number, followed by the Z location. `REQ`: Request mode. Automatic station position data reporting is disabled.
ERR_REP:	Requests the error reporting state. `AUT`: Automatic reporting of error. `REQ`: No reporting of errors (errors can be retrieved using "Request History" on page 54).
RR:	Response resend. `ON`: When the ^E character is sent, the host re-transmits all data characters that were sent to the host after the last command was received, including the prompts. If the checksum mode is ON, the response also includes checksum information. `OFF`: No re-transmission of data after the ^E character is entered.
Supported Devices	SEQ, MULTI, and MULTI_DEV are used with the FabExpress. BKG and BKG+ are NOT used with the Reliance ATR in the FabExpress system.
Compatibility	Compatible with Wave II Software/No known difference with Wave II Software.
Example	`RQ COMM ALL` `COMM PKT SEQ OFF ON OFF AUT 2 9600 AUT OFF`

Store Communication	
Command Syntax	`STORE COMM [ALL]|` `[(M/B|FLOW|LF|ECHO|CHECKSUM|ERRLVL|DREP|ERR_REP)] [RR (ON|OFF)]`
Response Syntax	There is no response from this command.

Copyright © 2023, Brooks Automation, Inc.

Store Communication	
M/B:	Specifies the serial I/O communications mode. `MON`: Monitor mode `PKT`: Packet mode (_RDY)
FLOW:	Stores the command execution type. `SEQ`: Sequential mode. `BKG`: Background mode. `BKG+`: Background Plus mode. `MULTI`: Multiple action mode. `MULTI_DEV`: Multiple Device mode. ***NOTE:*** *BKG and BKG+ are for backwards compatibility. These options were replaced with MULTI mode.*
LINEFEED:	Stores the linefeed state. `ON`: Linefeed enabled. `OFF`: Linefeed disabled (default).
ECHO:	Stores the ECHO state. `ON`: Echo enabled. `OFF`: Echo disabled. ***NOTE:*** *No checksum is displayed when echo is enabled.*
CHECKSUM:	Stores the checksum option state. `ON`: Checksum enabled. `OFF`: Checksum disabled.
ERR_LEVEL:	`err_level`: Stores the level of reported messages. Least number of messages = 1 Maximum number of messages = 5 Default = 2
DREP:	Stores the automatic station position data reporting flag. `AUT`: Automatic mode. Automatic station position data reporting is enabled. In this mode the robot automatically reports the station position for each section of a PICK, PLACE or XFER command. The auto response returns a string of 14 characters on every change in state of the robot arm location in terms of station coordinates. The data string consists of the R location, followed by the station number, followed by the slot number, followed by the Z location. `REQ`: Request mode. Automatic station position data reporting is disabled.
ERR_REP:	Stores the error reporting state. `AUT`: Automatic reporting of error. `REQ`: No reporting of errors (errors can be retrieved using "Request History" on page 54).
RR:	Response resend. `ON`: When the ^E character is sent, the host re-transmits all data characters that were sent to the host after the last command was received, including the prompts. If the checksum mode is ON, the response also includes checksum information. `OFF`: No re-transmission of data after the ^E character is entered.
Supported Devices	SEQ, MULTI, and MULTI_DEV are used with the FabExpress. BKG and BKG+ are NOT used with the Reliance ATR in the FabExpress system.
Compatibility	Compatible with WAVE II Software/No known difference with WAVE II Software.

Copyright © 2023, Brooks Automation, Inc.

Store Communication	

Example	The following example stores the current serial I/O communications mode in non-volatile memory.
	
	`STORE COMM M/B`
	
	The following example stores the current command execution mode in non-volatile memory.
	
	`STORE COMM FLOW`

Set/Request I/O Echo

The I/O ECHO command sets and requests the serial communications echo option. The I/O echo option is used to set full or half duplex communications. If the terminal or terminal emulator displays double characters for all user-entered text, IO ECHO should be set off (N). If the terminal or terminal emulator displays no characters for all user entered text, IO ECHO should be set on (Y).

To store the echo state persistently over power cycles, use the command STORE COMM ALL.

Set I/O Echo		
Command Syntax	`SET IO ECHO (Y|N)`	
Response Syntax	There is no response to this command.	
Arguments	ECHO:	`Y`: Sets the communications echo option on (full duplex) `N`: Sets the communications echo option off (half duplex)
Compatibility Notes	This command is for backwards compatibility only.	
Description	Sets the serial communications echo option.	

Example	The following example sets the communications mode in RAM to full duplex.
	`SET IO ECHO Y`
	
	The following example sets the communications mode in RAM to half duplex.
	`SET IO ECHO N`	

Request I/O Echo	
Command Syntax	`RQ IO ECHO`
Response Syntax	`IO ECHO (Y|N)`

Copyright © 2023, Brooks Automation, Inc.

Request I/O Echo		
Arguments	ECHO:	`Y`:The communications echo option is on (full duplex). `N`: The communications echo option is off (half duplex).
Compatibility Notes	This command is for backwards compatibility only.	
Description	Requests the serial communications echo option.	
Example	`RQ IO ECHO Y`	

Copyright © 2023, Brooks Automation, Inc.

I/O Echo Comment

The robot returns any text string that is entered after the ECHO command. This can be used in loops and scripts to provide annotation in the command logs. The ECHO command does not perform any operations on the robot, but rather acts as a diagnostic feature for the user.

Example	
Command Syntax	`ECHO PROCESS NUMBER 7 HAS STARTED`
Response Syntax	`PROCESS NUMBER 7 HAS STARTED`

Example	
Command Syntax	`ECHO The quick brown fox jumps over the lazy dog`
Response Syntax	`The quick brown fox jumps over the lazy dog`

Copyright © 2023, Brooks Automation, Inc.

Configure Ethernet

The Configure Ethernet command configures the address and mask to the robot. The command establishes Ethernet network communication protocol for the robot.The Request Ethernet command requests the Ethernet network communication protocol for the robot.

NOTE: This command does not require STORE; changes will take effect immediately. If you are changing the Ethernet IP address when connected over Ethernet, you will be disconnected using the new IP address.

NOTE: The IP address can also be changed using the CDM when using the serial port for communication.

Configure Ethernet		
Command Syntax	`CONFIG ENET IP address MASK mask [UNIT 0/1]` `RQ ENET IP [UNIT 0/1]`	
Response Syntax	`IP address MASK mask`	
Arguments	address:	IPv4 address assigned to the robot in dot-decimal format.
	mask:	The subnet mask in dot-decimal format.
Description	The optional `UNIT` parameter allow the user to select between the controller's HOST or the DIAG port. 0 – Used to configure or request information for the HOST port. If `UNIT` is not supplied, this is the default. 1 – Used to configure or request information for the DIAG port.	
Supported Devices	All MagnaTran LEAP robots.	
Compatibility Notes	Compatible with WAVE II Software.	
Description	Configures the address and mask to the robot.	

Examples	`RQ ENET IP`
	`IP 20.20.249.2 MASK 255.255.255.0`
	
	`RQ ENET IP UNIT 1`
	`IP 192.168.23.23 MASK 255.255.240.0`
	
	`CONFIG ENET IP 20.20.249.3 MASK 255.255.255.0`	

The default IP address for the HOST port: `20.20.249.2`

The default IP address for the DIAG port: `192.168.23.23`

Copyright © 2023, Brooks Automation, Inc.

Storing Configuration Data

Stores the command parameters for all available commands. Many commands have their own individual store commands, while some commands (as noted) automatically store their data when the assignment command is executed. A small number of items (as noted) are not storable.

Store (Store)		
Command Syntax	`STORE ALL`	
Response Syntax	None	
Arguments	ALL	Stores all available commands.
Description	Stores the command parameters for all available commands.	
Example	`STORE ALL`	

Copyright © 2023, Brooks Automation, Inc.

3. System

Request Who

Requests “who” the robot is by its Software Version number. This command is used to request the current version of the firmware installed in the robot.

Request Who		
Command Syntax	`RQ WHO`	
Response Syntax	`WHO BROOKS AUTOMATION V_version`	
Arguments	version:	The installed software version.
Description	Requests “who” the robot is by its Software Version number.	
Example	`RQ WHO` `WHO BROOKS AUTOMATION V8.2.0.11`	

Request Version

Requests the version and date. This command is supplied for backward compatibility.

Request Version		
Command Syntax	`RQ VERSION`	
Response Syntax	`version date`	
Arguments	version:	V x.x.x.x
	date:	mm dd yyyy

Copyright © 2023, Brooks Automation, Inc.

Request Version	
Description	Requests the version and date.
Example	RQ VERSION V8.2.0.19 10/29/2020

Request Revision

Requests the current Brooks part number and software revision number. This command is used to query the current revision of the software installed on the controller.

Request Revision		
Command Syntax	RQ RVSN	
Response Syntax	RVSN part_num revision	
Arguments	part_num:	The six digit software part number.
	revision:	The revision number of the software.
Description	Requests the current Brooks part number and software revision number.	
Example	RQ RVSN RVSN 355191 V8.2.0.19	

Request Serial Number

This command returns the serial numbers of the robot parts for identification purposes.

Request Serial Number	
Command Syntax	RQ SERIAL NUMBER [ROBOT|DRIVE|ARM]
Response Syntax	Serial number of requested part.
Arguments	ROBOT|DRIVE|ARM
Description	Returns the serial numbers of the robot parts for identification purposes.
Example	RQ SERIAL NUMBER ROBOT NONE

Copyright © 2023, Brooks Automation, Inc.

Request Date and Time

Requests the date or time from the robot. Date and time are always given in standard GMT time with no adjustment for time zones.

Request Date	
Command Syntax	`RQ DATE[YMD|MDY]`
Response Syntax	*Mon/Day/Year*
Arguments	None
Description	Requests the date from the robot.

Example	`RQ DATE`
	`09/18/2017`
	
	`RQ DATE YMD`
	`DATE: 2020-02-06`
	
	`RQ DATE MDY`
	`DATE: 2020-02-06`

Request Time	
Command Syntax	`RQ TIME`
Response Syntax	`time (24 hour clock)`
Arguments	None
Description	Requests the time from the robot.
Example	`RQ TIME` `09/18/2017`

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Compatibility

Sets, requests, and stores assorted parameters to a specific default state and defines the communication and coordinate system for the robot.

NOTE: Changing the Compatibility Mode will change robot behavior. This command is set at the factory and should not be changed.

The following information defines the compatibility mode:

- Table 3-1 compares the commands used in the MAG 6 mode to those used in the VT5 mode and
- notes any behavioral differences.
- Table 3-2 compares the error code differences of the MT5/VT5 to the MagnaTran 7.
- Table 3-3 compares the error code differences of the MagnaTran 6 to the MagnaTran 7.
- Table 3-4 shows the commands used to configure the standard VT5/MT5 compatibility for the MagnaTran 7.

Set Compatibility		
Command Syntax	`SET COMPATIBILITY [ALL (VT5|MAG6)] [ [COORDT (VT5|MAG6)] [ECHO ((VT5|MAG6)] [HALT (VT5|MAG6)] [CPTR (VT5|MAG6)] [RESP (VT5|MAG6)] [SPEED (VT5|MAG6)]]`	
Response Syntax	There is no response to this command.	
Arguments	ALL:	Sets all compatibility parameters to Mag6 or VT5.
	CORDT:	Sets Coordinate T to Mag6 or VT5.
	ECHO:	Sets IO ECHO to Mag6 or VT5.
	HALT:	Sets Halt to Mag6 or VT5.
	CPTR:	Sets Capture to Mag6 or VT5.
	RESP:	Sets Response to Mag6 or VT5.
	SPEED:	Sets Speed to Mag6 or VT5.
Description	Sets assorted parameters to a specific default state and defines the communication and coordinate system for the robot.	

Example	Set all compatibility parameters to Mag6:
	`SET COMPATIBILITY ALL MAG6`
	Set specific compatibility parameters:
	`SET COMPATIBILITY CORDT MAG6 ECHO MAG6 HALT MAG6 CAPT VT5 RESP MAG6 SPEED VT5`	

Copyright © 2023, Brooks Automation, Inc.

Request Compatibility		
Command Syntax	`RQ COMPATIBILITY [ALL]COORDT|ECHO|HALT|CPTR|RESP|SPEED]`	
Response Syntax	`RQ COMPATIBILITY ALL COMPATIBILITY COORDT (MAG6|VT5) ECHO (MAG6|VT5) HALT (MAG6|VT5) CPTR (MAG6|VT5) RESP (MAG6|VT5)SPEED (MAG6|VT5)`	
Arguments	ALL:	Requests all compatible parameters.
	CORDT:	Requests the Coordinate T compatibility.
	ECHO:	Requests the IO ECHO compatibility.
	HALT:	Requests the Halt compatibility.
	CPTR:	Requests the Capture compatibility.
	RESP:	Requests the Response compatibility.
	SPEED:	Requests the Speed compatibility.
Description	Requests assorted parameters to a specific default state and defines the communication and coordinate system for the robot.	
Example	`RQ COMPATIBILITY ALL` `COMPATIBILITY CORDT MAG6 ECHO MAG6 HALT MAG6 CAPT VT5 RESP MAG6 SPEED VT5`	

Store Compatibility		
Command Syntax	`RQ COMPATIBILITY [ALL]COORDT|ECHO|HALT|CPTR|RESP|SPEED]`	
Response Syntax	`RQ COMPATIBILITY ALL` `COMPATIBILITY COORDT (MAG6|VT5) ECHO (MAG6|VT5) HALT (MAG6|VT5) CPTR (MAG6|VT5) RESP (MAG6|VT5)SPEED (MAG6|VT5)`	
Arguments	ALL:	Stores all compatible parameters.
	CORDT:	Stores the Coordinate T compatibility.
	ECHO:	Stores the IO ECHO compatibility.
	HALT:	Stores the Halt compatibility.
	CPTR:	Stores the Capture compatibility.
	RESP:	Stores the Response compatibility.
	SPEED:	Stores the Speed compatibility.
Description	Stores assorted parameters to a specific default state and defines the communication and coordinate system for the robot.	
Example	`STORE COMPATIBILITY ALL`	

Copyright © 2023, Brooks Automation, Inc.

Table 3-1: Command Comparison MAG6 to VT5

Command	MAG6	VT5
FIND PHASE		Must enter options.
GOTO	"ARM" is optional.	"ARM" is required. Wafer speed is used regardless of load status.
HALT	Robot comes to a controlled stop and the arm remains under servo control.	Robot comes to a controlled stop and the armset servos disengages.
HLLO	Returns "HELLO"	Returns ready string.
HOME		
LFTST		
MAP		Not supported.
MOVE	"ARM" is optional.	"ARM" is required. Wafer speed is used regardless of load status.
PICK	"ARM" is optional.	"ARM" is required. SLOT number starts with zero.
PLACE	See PICK	Initial motions of PLACE will be performed at "with wafer" speed.
REMOVE IO		Not supported.
RESET	Returns "Brooks Automation."	Returns different string.
RQ BG		
RQ COMM	[DREP] is not supported. New option [LF]	Has [DREP] option.
RQ CPTR	Command format is the same but the return formats are different. Response: R 1234567 T 1234567 Z 1234567 L	Response: CPTR R 1234567 T 1234567 Z 1234567 L.
RQ IO ECHO	Command format is the same but the return formats are different. In monitor mode, RQ IO ECHO response is IO ECHO Y. In packet mode, RQ IO ECHO response is IO ECHO Y.	In monitor mode, RQ IO ECHO response is Echo status: Y. In packet mode, RQ IO ECHO response is IO Y.
RQ IO MAP		Not supported.
RQ IO STATE		Not supported.

Copyright © 2023, Brooks Automation, Inc.

Command	MAG6	VT5
RQ LOAD		Not supported.
RQ POS ABS	"ARM" is optional. Always returns in RTZ order. Monitor mode reponse to RQ POS ABS ALL: POS ABS RADIAL.....xxxxxx THETA.....xxxxxx Z.....xxxxxx	Does not support "ARM." Return does not include "ARM." Returns RTZ in order requested. Monitor mode response to RQ POS ABS ALL: Radial: xxxxxx Theta: xxxxxx Z: xxxxx
RQ POS STN	"ARM" is optional.	Does not support "ARM."
RQ POS TRG	"ARM" is optional.	Does not support "ARM." Return does not include "ARM."
RQ STN	"ARM" is optional.	Does not support "ARM." Return does not include "ARM."
RQ STN OPTION		
RQ STNSENSOR	"ARM" is optional. Command format is the same but the return formats are different. In monitor mode, the response is: STN 1 ARM A TYPE.....RE ACT.....LO SEN.....1 STATE.....OFF In packet mode, the response is: STN 1 ARM A TYPE RE ACT HI SEN 1 STATE OFF	"ARM" must be typed. In monitor mode, the response is: STN 01 ARM A TYPE RE ACT LO SEN 01 STATE OFF In packet mode, the response is: STN 01 ARM A TYPE RE ACT HI SEN 01 STATE OFF
SET COMM	[DREP] is supported. New option [LF]	Has [DREP] [AUT|REQ] option.
SET HISPD SET LOSPD	Sets the speed for MOVE and GOTO action commands. Pan speed is always used when LOAD is OFF.	Sets the speed for MOVE and GOTO action commands. Action is performed at wafer speed and pan speed is used for PICK and PLACE only.
SET IO ECHO		
SET IO STATE		Not supported.
SET LOAD		Not supported.

Copyright © 2023, Brooks Automation, Inc.

Command	MAG6	VT5
SET STN	"ARM" is optional.	"ARM" is not supported.
SET STN OPTION		Not supported.
SET STNSENSOR	"ARM" is optional. Error codes are different from VT5.	"ARM" is required.
SET COMM	[DREP] is not supported. New option [LF]	Has [DREP] [AUT|REQ] option.
STORE IO ECHO		
STORE STN	"ARM" is optional.	
STORE STN OPTION		Not supported.
STORE STNSENSOR	"ARM" is optional.	"ARM" is required.
XFER	"ARM" is optional.	"ARM" is required.

Copyright © 2023, Brooks Automation, Inc.

Error Code Comparison

Table 3-2 compares the error code differences of the MT5/VT5 to the MagnaTran 7.

Table 3-2: Error Code Comparison

Error Number	Existing MT5/VT5 Error Code	Error Number	Mag 7/70/77 Error Code
411	Interlock Calc Overflow	408	Bad R Position
412	Invalid Arm Locate	No Equivalent	
413	Invalid Wafer Size	No Equivalent	
501	Internal Error	No Equivalent	
502	ESC CDM Error	No Equivalent	
503	Void Error	No Equivalent	
602	Command Overrun	605	Busy
707	Pick Fail	721	Pick Failed
708	Place Fail	722	Place Failed
709	Interlock Calc Overflow	No Equivalent	
710	Interlock Violation		Slot valve closed prior to PICK/PLACE/GOTO/XFER.

Table 3-3: Error Code Comparison Mag 6/60 Mag 7/70/77

Error Number	Existing Mag 6/60 Error Code	Error Number	Mag 7/70/77 Error Code
411	Station Not Initialized	416	Station Not Initialized
412	Offset Too Large	417	Offset Too Large
413	Invalid RTRCT2	418	Invalid RTRCT2
501	MCC COMM Error	527	MCC COMM Error
502	MCC Queue	528	MCC Queue
503	MCC No ID	529	MCC No ID
707	Not Used	No Equivalent	

Copyright © 2023, Brooks Automation, Inc.

Error Number	Existing Mag 6/60 Error Code	Error Number	Mag 7/70/77 Error Code
708	Pick Failed	721	Pick Failed
709	Place Failed	722	Place Failed

Configuration Compatibility Commands

The following commands may be used to ensure the communication protocol of the MagnaTran 7 to be the same as the MT5/VT5 (Arm A and B Theta positions for a given station are 180° apart/servos freewheel at HALT).

Table 3-4 shows the commands used to configure the standard VT5/MT5 compatibility for the MagnaTran 7.

NOTE: The ALL option is also available for each of the COMPATIBILITY commands. For example, to request all of the settings: RQ COMPATIBILITY ALL or to store all settings: STORE COMPATIBILITY ALL or to set all: SET COMPATIBILITY ALL VT5.

Table 3-4: Standard VT5/MT5 Compatibility

Parameter Type	Setting	Command to SET	Command to STORE	Command to REQUEST
Communications Protocol	RS-232	None (with Dip Switch)	None	None
Serial Communications Mode	Monitor	SET COMM M/B MON	STORE COMM M/B	RQ COMM M/B
Command Flow Mode	Sequential	STORE COMM FLOW SEQ	STORE COMM FLOW	RQ COMM FLOW
Terminal Echo Mode	ON	STORE COMM ECHO Y	STORE COMM ECHO	RQ COMM ECHO
Linefeed After Carriage Return	ON	STORE COMM LF ON	STORE COMM LF	RQ COMM LF
Error Reporting Level	2	STORE COMM ERRLVL 2	STORE COMM ERRLVL	RQ COMM ERRLVL
Data Reporting	REQ	STORE COMM DREP REQ	STORE COMM DREP	RQ COMM DREP
Terminal Echo Compatibility	VT5	STORE COMPATIBILITY ECHO VT5	STORE COMPATIBILITY ECHO	RQ COMPATIBILITY ECHO

Copyright © 2023, Brooks Automation, Inc.

Parameter Type	Setting	Command to SET	Command to STORE	Command to REQUEST
Theta Coordinate Compatibility	VT5	STORE COMPATIBILITY COORDT VT5	STORE COMPATIBILITY COORDT	RQ COMPATIBILITY COORDT
		Theta coordinate system defines theta HOME for Arm A as 0°/Arm B as 180°.		
HALT Compatibility	VT5	SET COMPATIBILITY HALT VT5	STORE COMPATIBILITY HALT	RQ COMPATIBILITY HALT
Capture Response Compatibility	VT5	SET COMPATIBILITY CPTR VT5	STORE COMPATIBILITY CPTR	RQ COMPATIBILITY CPTR
Wafer Speed Compatibility	VT5	SET COMPATIBILITY SPEED VT5	STORE COMPATIBILITY SPEED	RQ COMPATIBILITY SPEED
		Sets the speed for MOVE and GOTO action commands to be compatible with VT5; action is performed at wafer speed, and pan speed is used for PICK and PLACE only.		
Response Compatibility	VT5	SET COMPATIBILITY RESP VT5	STORE COMPATIBILITY RESP	RQ COMPATIBILITY RESP
		Sets the Response to RQ CPTR, RQ POS ABS ALL, and RQ STNSENSOR commands to be compatible with VT5.		
Checksum Mode	OFF	SET COMM CHECKSUM OFFA6	STORE COMM CHECKSUM	RQ COMM CHECKSUM

Table 3-5 shows the commands used to configure the standard MagnaTran 6 compatibility for the MagnaTran 7 (Arm A and B Theta positions are 0° apart/servos hold position at HALT).

NOTE: The ALL option is also available for each of the COMPATIBILITY commands. For example, to request all of the settings: RQ COMPATIBILITY ALL or to store all settings: STORE COMPATIBILITY ALL or to set all: SET COMPATIBILITY ALL MAG6.

Copyright © 2023, Brooks Automation, Inc.

Table 3-5: Standard MagnaTran 6 Compatibility

Parameter Type	Setting	Command to SET	Command to STORE	Command to REQUEST
Communications Protocol	RS-232	None (with Dip Switch)	None	None
Serial Communications Mode	Monitor	SET COMM M/B MON	STORE COMM M/B	RQ COMM M/B
Command Flow Mode	Sequential	SET COMM FLOW SEQ	STORE COMM FLOW	RQ COMM FLOW
Terminal Echo Mode	ON	SET IO ECHO Y	STORE COMM ECHO	RQ COMM ECHO
Linefeed After Carriage Return	ON	SET COMM LF ON	STORE COMM LF	RQ COMM LF
Error Reporting Level	2	SET COMM ERRLVL 2	STORE COMM ERRLVL	RQ COMM ERRLVL
Data Reporting	REQ	SET COMM DREP REQ	STORE COMM DREP	RQ COMM DREP
Terminal Echo Compatibility	MAG6	SET COMPATIBILITY ECHO ,MAG6	STORE COMPATIBILITY ECHO	RQ COMPATIBILITY ECHO
Theta Coordinate Compatibility	MAG6	SET COMPATIBILITY COORDT MAG6	STORE COMPATIBILITY COORDT	RQ COMPATIBILITY COORDT
		Theta coordinate system defines theta HOME for Arm A and Arm B as 0°.		
HALT Compatibility	MAG6	SET COMPATIBILITY HALT MAG6	STORE COMPATIBILITY HALT	RQ COMPATIBILITY HALT
Capture Response Compatibility	MAG6	SET COMPATIBILITY CPTR MAG6	STORE COMPATIBILITY CPTR	RQ COMPATIBILITY CPTR
Wafer Speed Compatibility	MAG6	SET COMPATIBILITY SPEED MAG6	STORE COMPATIBILITY SPEED	RQ COMPATIBILITY SPEED
		Sets the speed for MOVE and GOTO action commands to be compatible with MAG6; pan speed is always used when LOAD is OFF.		
Response Compatibility	MAG6	SET COMMPATIBILITY RESP MAG6	STORE COMMPATIBILITY RESP	RQ COMMPATIBILITY RESP
		Sets the Response to RQ CPTR, RQ POS ABS ALL, and RQ STNSENSOR commands to be compatible with MAG6.		

Copyright © 2023, Brooks Automation, Inc.

Parameter Type	Setting	Command to SET	Command to STORE	Command to REQUEST
Checksum Mode	OFF	SET COMM CHECKSUM offa6 This command should be set last. This command will turn off Checksum mode if it is ON. Default = OFF. Command is case sensitive.	STORE COMM CHECKSUM	RQ COMM CHECKSUM

Reset

Performs a software reset of the robot's firmware.

NOTE: This command is NOT used in the normal operation of the robot. Contact Brooks Technical Support for more information. For more information on contacting Brooks Technical Support, please see "Contact Brooks Automation Technical Support".

RESET: Performs a software reset of the robot's firmware that is functionally equivalent to turning the power off and then back on. All parameters stored in RAM will be replaced by the values stored in nonvolatile memory. After approximately 30 seconds, a "ready" response or "BROOKS AUTOMATION" is returned when reset as complete.

Reset	
Command Syntax	`RESET`
Response Syntax	There is no response to this command - the Controller resets.
Arguments	None
Description	Performs a software reset of the robot's firmware. Resets the robot and loads the user default settings stored in non-volatile memory for all parameters.
Example	`RESET`

Copyright © 2023, Brooks Automation, Inc.

Request Error Message

Requests and reports the text message assigned to the error code number.

Request Error Message		
Command Syntax		`RQ ERRMSG [ERRCODE|ALL]`
Response Syntax		Error Description
Arguments	error_code:	Error number
Description	Requests and reports the text message assigned to the error code number.	

Example	`RQ ERRMSG 0305`
	`ERR 305: _ERR = Unknown command or argument.`
	
	`RQ ERRMSG 10023`
	`ER 10023: _ERR = Arm actual location impossible, check sync zero.`	

NOTE: If all parameters are supplied, a list of all robot error codes will be returned.

Copyright © 2023, Brooks Automation, Inc.

Request History

Requests the history of events performed by the robot. Information requests are treated independently to allow for a maximum amount of information for each type.

The argument for number of records to be displayed is optional. If no value is entered, all available records will be displayed. If a “1” is entered, only the last logged (most recent) record to be displayed.

If both CMD and ERR are selected, they will be sent concurrently by date. Latest information is transmitted last. If TOT is selected, these totals will be transmitted after any CMD and ERR information; TOT information by axis is fixed (not based on date of last error for given axis).

Request History		
Command Syntax	`RQ HISTORY [(CMD|ERR|TOT)][count | ALL]`	

Response Syntax	`CMD: date time command`
	`ERR: date time command`
	`description`
	`error_code error_msg`
	`cycle_count {axis total_travel}`
	`TOT: date time axis error_count`	

Arguments	ALL:	Requests non-action commands, errors, and the total number of motion errors for each axis.
	CMD:	Requests non-action commands (set and store).
	ERR:	Requests errors.
	TOT:	Requests the total number of motion errors for each axis.
	#records:	Number of records to be displayed.
Description	Requests the history of events performed by the robot.	

Copyright © 2023, Brooks Automation, Inc.

Request History	
Example	The following example requests all available history information: RQ HISTORY CMD 5 CMD: 02/06/2020 15:28:27.508 SET STN 41 A T 180000.75 CMD: 02/06/2020 15:28:28.283 SET RESOLUTION HIGH CMD: 02/06/2020 15:28:29.364 STORE STN 41 ARM A ALL CMD: 02/06/2020 15:28:29.950 CALL RESET CONTROLLER CMD: 02/06/2020 15:56:32.332 SET IO ECHO Y RQ HISTORY ERR 3 ERR: 02/06/2020 15:28:09.539 HOME ALL N MOVING TO STN N R NAS T NAS Z NAS SLOT N ARM A 10061 _ERR = position impossible to reach. 11530 R 349063 T 5000 Z -20470000 ERR: 02/06/2020 15:28:13.939 HOME ALL N MOVING TO STN N R NAS T NAS Z NAS SLOT N ARM A 10061 _ERR = position impossible to reach. 11530 R 349063 T 5000 Z -20470000 ERR: 02/06/2020 15:28:25.697 GOTO N 42 Z UP R EX N MOVING TO STN N R NAS T NAS Z NAS SLOT N ARM A 10061 _ERR = position impossible to reach. 11530 R 349063 T 5000 Z -20470000 RQ HISTORY TOT 4 TOT: 02/06/2020 14:31:57.851 R 376 TOT: 10/01/2019 12:31:40.559 T 36 TOT: 02/06/2020 15:23:46.021 Z 391

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Warning CDM Status

Sets, requests, and stores the warning feature of the CDM. This command is used to turn the CDM warning feature on or off. For more information on the CDM, refer to the *MagnaTran LEAP User Manual*.

If the feature is enabled, the host will receive an unsolicited error message “CDM has control of the robot” when the CDM is turned on. Additionally, when the CDM is turned on or off, warning messages “CDM has been turned on” and “CDM has been turned off” will be displayed appropriately.

NOTE: This command is not functional in Ethernet mode.

Set Warning CDM Status		
Command Syntax	`SET WARN CDM (Y|N)`	
Response Syntax	There is no response to this command.	
Arguments	status:	`Y`: Enable warning CDM status. `N`: Disable warning CDM status.
Description	Sets the warning feature of the CDM.	
Example	`SET WARN CDM Y`	

Request Warning CDM Status		
Command Syntax	`RQ WARN CDM`	
Response Syntax	There is no response to this command.	
Arguments	status:	`Y`: Warning CDM status is enabled. `N`: Warning CDM status is disabled.
Description	Requests the warning feature of the CDM.	
Example	`RQ WARN CDM` `WARN CMD Y`	

Store Warning CDM Status	
Command Syntax	`STORE WARN CDM`
Response Syntax	There is no response to this command.
Arguments	None

Copyright © 2023, Brooks Automation, Inc.

Store Warning CDM Status	
Description	Stores the warning feature of the CDM.
Example	`STORE WARN CDM`

Backup

The BACKUP command can be used to store and duplicate (with the RESTORE command) the specific selections of a customer's robot. This includes robot configuration, station, I/Os, and options. The BACKUP command creates a backup file called Backup.zip. This file can be accessed on the robot via FTP file transfer.

Backup	
Command Syntax	`BACKUP`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores and duplicates (with the RESTORE command) the specific selections of a customer's robot.
Example	`BACKUP` ***NOTE:*** *(File Backup.zip created in TMP.)*

Copyright © 2023, Brooks Automation, Inc.

Restore

The RESTORE command takes the file created from the BACKUP command and sets all of the values contained in the file on the robot. This is useful in replacing a robot to make sure that the replacement is the same as the original robot.

NOTE: The Backup.zip from the previous backup must be transferred via FTP prior to executing the RESTORE command.

NOTE: If there is no Backup.zip file in the TMP directory, then an error message will be generated (_ERR7005).

Restore	
Command Syntax	`RESTORE`
Response Syntax	There is no response to this command.
Arguments	None
Description	Takes the file created from the BACKUP command and sets all of the values contained in the file on the robot.
Example	`RESTORE`

Copyright © 2023, Brooks Automation, Inc.

4. MISC I/O

LEAP Input/Output (I/O)

LEAP robots support input and output signals to external equipment via the 50-pin Miscellaneous I/O connector on the control panel. These signals can be configured by the user in a wide variety of ways. I/O signals must be configured (i.e. "mapped") before they can be used by the user.

The signals on this connector are industry standard 24 Volt logic where a “1” is approximately 24 volts and a logical “0” is approximately zero volts. Please refer to the *MagnaTran LEAP User Manual* for specific details on the signal levels and loading.

The signals can also be configured as HIGH SIDE where a logic “1” is 24 Volts and a logic “0” is zero Volts or LOW SIDE where a logic “1” is zero Volts and a logic “0” is 24 Volts. Any of the available signals can be individually configured as either HIGH SIDE or LOW SIDE. Please refer to the *MagnaTran LEAP User Manual* for specific details on how to hook up devices to HIGH SIDE or LOW SIDE signals.

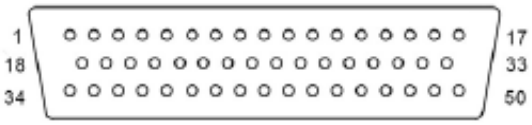


Figure 4-1: 50-Pin Miscellaneous I/O Connector

Figure 4-1 shows the layout of the signals on 50-pin Miscellaneous I/O connector. Note that the signals are generally referred to as IN_0, IN_1, OUT_2, etc.

There are several non-signal connections that must exist on this connector including the following: 1) I/O power and 2) Safety Interlock. For convenience, the required connections include: 1) Connections for IO Power must exist as shown in Figure 4-1 between pins 26 & 30 and 27 & 29 or none of the OUTxx signals will have power; and 2) For the Safety Interlock, pins 23 & 24 must be connected. Once configured as described, I/O signals will be able to be observed on signals OUT_1, OUT_2, etc.

Copyright © 2023, Brooks Automation, Inc.

LEAP I/O Modes

LEAP robots currently support three broad configurations of I/O. These configurations must be selected via robot commands and stored so that they remain configured over power cycles. The different modes invoke different behavior of the I/O signals, so it is important to choose the proper mode. The I/O Modes are:

- "Standard I/O Mode" on page 60 (default)
- "Extended I/O Mode" on page 81
- "DIO I/O Mode" on page 84

Each of these I/O Modes will be explained in detail in the following sections and examples will be provided. Note that these modes are generally mutually exclusive, so all the I/O follow the rules of the currently selected mode. Also, these modes have to be selected and stored by the user to persist over power cycles.

Standard I/O Mode

Standard I/O Mode is the default I/O mode and the default. It is very similar to the I/O provided by WAVE robots.

Standard I/O supports 22 input signals and 20 output signals as shown in Figure 4-2.

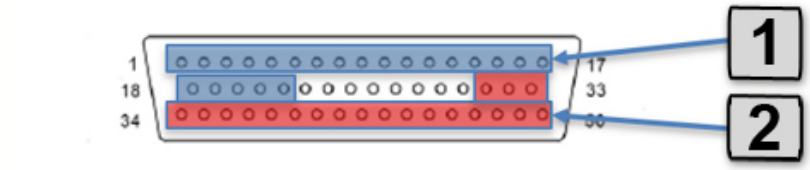


Figure 4-2: Standard I/O Input and Output Signals

1	22 Inputs
2	20 Outputs

Table 4-1: Standard I/O Pinout

Pin #	Signal Name	Signal Description	I/O Mask	Notes
1	IN_0	INPUT 1	0x01	
2	IN_1	INPUT 2	0x02	

Copyright © 2023, Brooks Automation, Inc.

Pin #	Signal Name	Signal Description	I/O Mask	Notes
3	IN_2	INPUT 3	0x04	
4	IN_3	INPUT 4	0x08	
5	IN_4	INPUT 5	0x10	
6	IN_5	INPUT 6	0x20	
7	IN_6	INPUT 7	0x40	
8	IN_7	INPUT 8	0x80	
9	IN_8	INPUT 9	0x100	
10	IN_9	INPUT 10	0x200	
11	IN_10	INPUT 11	0x400	
12	IN_11	INPUT 12	0x800	
13	IN_12	INPUT 13	0x1000	
14	IN_13	INPUT 14	0x2000	
15	IN_14	INPUT 15	0x4000	
16	IN_15	INPUT 16	0x8000	
17	IN_16	INPUT 17	0x10000	
18	IN_17	INPUT 18	0x20000	
19	IN_18	INPUT 19	0x40000	
20	IN_19	INPUT 20	0x80000	
21	IN_20	INPUT 21	0x100000	
22	IN_21	INPUT 22	0x200000	
23	INTLCK_0	INTLCK 0	-	CONNECT TO INTLCK_1 (PIN 24) - See note
24	INTLCK_1	INTLCK 1	-	CONNECT TO INTLCK_0 (PIN 23) - See note
25	+PWR_ ISOL	+PWR_ISOL	-	
26	+PWR_ ISOL	+PWR_ISOL	-	CONNECT TO +24 VDC (PIN 30) - See note
27	PWR RET ISOL	PWR RET ISOL	-	CONNECT TO 24V RET (PIN 29) - See note
28	PWR RET ISOL	PWR RET ISOL	-	
29	24V RET	24V RET	-	CONNECT TO +PWR_ISOL (PIN 27) - See note
30	+24 VDC	+24 VDC	-	CONNECT TO +PWR_ISOL (PIN 26) - See note

Copyright © 2023, Brooks Automation, Inc.

Pin #	Signal Name	Signal Description	I/O Mask	Notes
31	OUT_0	OUTPUT 1	0x01	
32	OUT_1	OUTPUT 2	0x02	
33	OUT_2	OUTPUT 3	0x04	
34	OUT_3	OUTPUT 4	0x08	
35	OUT_4	OUTPUT 5	0x10	
36	OUT_5	OUTPUT 6	0x20	
37	OUT_6	OUTPUT 7	0x40	
38	OUT_7	OUTPUT 8	0x80	
39	OUT_8	OUTPUT 9	0x100	
40	OUT_9	OUTPUT 10	0x200	
41	OUT_10	OUTPUT 11	0x400	
42	OUT_11	OUTPUT 12	0x800	
43	OUT_12	OUTPUT 13	0x1000	
44	OUT_13	OUTPUT 14	0x2000	
45	OUT_14	OUTPUT 15	0x4000	
46	OUT_15	OUTPUT 16	0x8000	
47	OUT_16	OUTPUT 17	0x10000	
48	OUT_17	OUTPUT 18	0x20000	
49	OUT_18	OUTPUT 19	0x40000	
50	OUT_19	OUTPUT 20	0x80000	

NOTE: There are numerous connections possible for these pins - please consult the *MagnaTran LEAP User Manual* for other configurations. The configuration shown is the simplest to support measurable outputs from the robot.

Standard I/O requires the INxx and OUTxx signals to be defined before use. This process is called mapping I/O signals. User signals can be single signal, or a collection of signals grouped into a numeric value (i.e. an 8-bit pressure value). Once a signal is mapped as an input, it can be read in by a user. Once a signal is mapped as an output, it can be set by the user.

In some cases, signals can be mapped to stations to provide coordinated motion of an arm based on an I/O signal.

Copyright © 2023, Brooks Automation, Inc.

Once user defined signals are mapped, they should be stored to persist over power cycles. These signals can now be set using SET IO STATE commands or read using RQ IO STATE commands. The following sections will provide details on how to do this with examples.

LEAP Robot Outputs

LEAP Robot outputs include interlock signals, power indicators, and other various outputs.

Refer to Appendix B. "I/O Input/Output Examples" on page 339 for examples.

- **NUMERIC_OUT**: Allows multiple output pins to be grouped to form a digit. A binary number is converted into decimal and presented on the consecutive grouped output pins.
- **DISCRETE_OUT**: Single, discreet output pin that can be defined by the user to interface with the process controller or other external device.
- **RETRACT_PIN**: Output that indicates when the robot arm is in the retracted state. The RETRACT_PIN signal is controlled by the robot motion controller and should not be set by the user.
- **RETRACT_SEN**: Individual sensor monitoring of robot retract.
- **EXTEND_SEN**: The IO event 9005 will be sent when the IO is set to EXTENDED. The IO event 9006 will be sent when the IO is set to NOT_EXTENDED. The RQ STN n OPTION ALL response will contain this new station option.
- **CDM_CTRL**: Output signal that indicates that the CDM is in control of the robot and that the Host inputs are being ignored.
- **POWER_IND**: Output signal that indicates whether the robot is powered up. Output will be HI or LOW.
- **MOTOR_POWER_IND**: Output signal that indicates that the robot arm is powered.
- **STOP_FUNC_IND**: Output signal that indicates an E-Stop or Interlock condition.
- **LIGHT_TOWER**: Output signals that can be grouped to provide control of a light tower, including blinking.
- **STN_FACET**: Output bits are usually grouped as three bits to provide a numeric indication of the facet of the current arm motion (values usually 0 to 7).
- **SYS_CTRL_ON**: Output signal used with the Safety Hub.
- **HEART_BEAT**: Output signal that oscillates at 10 Hz to indicate that the robot is powered and the CPU is operating. In order to map the heart beat output, then the user must define SYS_CTRL_ON and SAFE_SPEED_CTRL. Then the HEART_BEAT must be mapped. Alternatively, the user can perform the CONFIG SAFETY_SETTING to perform the mapping of all three.
  NOTE: This configuration is compatible with the Safety Hub.
- **NEAR_STN**: Output signal that indicates that the robot arm's end effector is near the station (soon to be at station).
- **EXCL_REQ**: Output signal used with Exclusion Zones, robot asserts this signal to ask for permission to move into the Exclusion Zone and waits for the associated input to execute move.
- **HANDOFF_OUTPUT**: Used with the active handoff feature described in "Set/Request/Store Active Handoff Option" on page 167.
- **VIBRATION_IND**: Signal that indicates that excessive vibration has occurred and stopped the arm motion.

Command	Bit	Events	I/O State
NUMERIC_OUT	MULTIPLE	NULL	N/A
DISCRETE_OUT	SINGLE	3000 (INACTIVE) 3001 (ACTIVE)	INACTIVE ACTIVE

Copyright © 2023, Brooks Automation, Inc.

Command	Bit	Events	I/O State
RETRACT_PIN	SINGLE	3015 (IN) 3016 (OUT)	IN OUT
RETRACT_SEN	SINGLE	3019 (RETRACTED) 3020 (NOT_RETRACTED)	RETRACTED NOT_RETRACTED
EXTEND_SEN	SINGLE	9005 (EXTENDED) 9006 (NOT_EXTENDED)	EXTENDED NOT_EXTENDED
CDM_CTRL	SINGLE	N/A	N/A
POWER_IND	SINGLE	N/A	ON OFF
MOTOR_POWER_ IND	SINGLE	N/A	ON OFF
STOP_FUNC_IND	SINGLE	3017 (ON) 3018 (OFF)	ON OFF
LIGHT_TOWER	SINGLE	3034 3035	LIGHT_TOWER ON LIGHT_TOWER OFF
STN_FACET	MULTIPLE	N/A	N/A
SYS_CTRL_ON	SINGLE	N/A	N/A
HEART_BEAT	SINGLE	N/A	N/A
NEAR_STN	SINGLE	3025 3026	NEAR NOT_NEAR
EXCL_REQ	SINGLE	N/A	ENABLED DISABLED
HANDOFF_OUTPUT	SINGLE	N/A	ENABLED DISABLED
VIBRATION_IND	SINGLE	N/A	EXIST NOT EXIST

Copyright © 2023, Brooks Automation, Inc.

LEAP Robot Inputs

Refer to Appendix B. "I/O Input/Output Examples" on page 339 for examples.

- **NUMERIC_IN**: Allows monitoring of Input Signals as a group. Must be consecutive Input bits. Individual signals are aggregated into a digit for reading.
- **DISCRETE_IN**: Allows monitoring of individual Input Signals from the Miscellaneous I/O connector.
- **WAF_SEN**: Input signal bit for monitoring wafer sensor inputs.
- **SBIT_SVLV_SEN**: Allows single bit monitoring of the slot valve when the robot is extending. If the slot valve is closed, an error message will occur and the robot will stop. A fault response will be given.
- **EX_ENABLE**: Allows the process module to allow/prohibit the robot arm from extending.
- **VLV_SEN**: Input signal bit that allows monitoring of the poppet valve when the robot is extending. If the poppet valve is not closed, an error message will occur and the robot will stop.
- **HALT_CTRL**: Input signal bit that allows an external device to cause the robot to go into the Halt state.
- **EMER_STOP**: Input signal bit that allows an external device to cause the robot to go into an E-Stop state.
- **SAFE_SPEED_CTRL**:
- **FAN_PRESSURE**: Allows monitoring of sensors as a group. A binary number is converted into decimal and presented on multiple consecutive INPUT channels.
- **EXCL_CTRL**: Input signal bit used for Exclusion Zone control. This input is supplied by the external Exclusion Zone controller to tell the robot that it is safe to move into the zone.
- **HANDOFF_INPUT**: Used with the active handoff feature described in "Set/Request/Store Active Handoff Option" on page 167.

Command	Bit	Events	I/O State
NUMERIC_IN	MULTIPLE	NULL	N/A
DISCRETE_IN	SINGLE	3000 (INACTIVE) 3001 (ACTIVE)	INACTIVE ACTIVE
WAF_SEN	SINGLE	3002 (BLOCKED) 3003 (NOT_BLOCKED)	BLOCKED NOT_BLOCKED
SBIT_SVLV_SEN	SINGLE	3005 (NOT OPEN) 3006 (OPEN)	NOT OPEN OPEN
EX_ENABLE	SINGLE	3009 (DISABLED) 3010 (ENABLED)	DISABLED ENABLED
VLV_SEN	SINGLE	3013 (CLOSED) 3014 (NOT_CLOSED)	CLOSED NOT_CLOSED
HALT_CTRL	SINGLE	3017 (OFF) 3018 (ON)	OFF ON

Copyright © 2023, Brooks Automation, Inc.

Command	Bit	Events	I/O State
EMER_STOP	SINGLE	3017 (ON) 3018 (OFF)	ON OFF
SAFE_ SPEED_CTRL	DOUBLE	N/A	AUTO MANUAL HALT ERROR
FAN_ PRESSURE	SINGLE	N/A	NORMAL LOW
EXCL_CTRL	SINGLE	N/A	ON OFF
HANDOFF_ INPUT	SINGLE	N/A	ACTIVE INACTIVE

Map IO

The MAP command allows the flexibility to create interlocks and/or control digital devices by assigning a name to physical I/O bits. This command also assigns the use of the I/O and defines the specific Active State of the I/O.

> ***NOTE:*** Sensors must be configured using the MAP command before they can be assigned using the CDM.

The set command is used to assign an internal reference name, type or use, and, if required, a characteristic to physical I/O. Referring to the I/O by its reference name automatically references the I/O by its type, therefore defining the nature of the reference.

For I/O types defined as active high:

- Setting that I/O to the ACTIVE state will cause the signal to go HI
- Setting that I/O to the INACTIVE state will cause the signal to go LOW.
- Reading that I/O when the signal is HI will cause an ACTIVE response.
- Reading that I/O when the signal is LOW will cause an INACTIVE response

For I/O types defined as active low:

- Setting that I/O to the ACTIVE state will cause the signal to go LOW.
- Setting that I/O to the INACTIVE state will cause the signal to go HI.
- Reading that I/O when the signal is LOW will cause an ACTIVE response
- Reading that I/O when the signal is HI will cause an INACTIVE response.

Copyright © 2023, Brooks Automation, Inc.

NOTE: The actual terms used by the I/O instead of “ACTIVE” and “INACTIVE” are defined by the I/O type and are specified in the descriptions that reference the I/O. For example, OPEN or NOT_OPEN; RETRACTED or NOT_RETRACTED.

The I/O types are defined as performing specific functions with the settings and responses defined by those functions.

NOTE: When defining the NUMERIC I/O type, the I/O channels being specified must be consecutive and the low order channel must be the least significant bit.

It is possible to define a specific I/O channel using multiple MAP commands where the function of that I/O is defined differently in the different MAP commands. This allows an I/O channel to be referenced in different ways depending upon the function being performed.

For example; it may be convenient to identify the Wafer Sensors as both WAF_SEN and as NUMERIC allowing individual monitoring of the sensors (using RQ IO STATE WAF_SEN) and monitoring of the sensors as a group (using RQ IO NUMERIC).

Map I/O	
Command Syntax	`MAP [io_name] [io_type] [HI|LOW] (TO|NOT) [io_name] [io_num]` *The MAP command will automatically store the new parameters.*
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Map I/O		
Arguments	name:	User reference name to be assigned to the specified I/O Range = 20 characters maximum.
	type:	Specifies the type of I/O by its specific function.
	HI|LO:	Defines the active state of the I/O device being assigned as defined by the type of argument. To allow flexibility, the user may define the characteristic as active HI or active LOW depending on the hardware functionality.
	TO:	Assigns the same polarity of the input bit to the output bit.
	NOT:	Assigns the opposite polarity of the input bit to the output bit.
	io_name:	Name assigned to physical I/O for device being mapped or assigned. The I/O devices available to the robot are: DIGITAL_IN DIGITAL_OUT
	io_num:	Specifies the I/O channels being mapped or assigned. This variable is an eight digit (maximum) hex number of the form 0x12345678 representing the specific I/O channel(s). See the examples that follow for assistance in designating the io_num. ***NOTE:*** *That leading zeros may be dropped from this number.*
	ALL:	All defined I/O points.
	NOTE: *At least one argument must be specified.*	
Description	Allows the flexibility to create interlocks and/or control digital devices by assigning a name to physical I/O bits. This command also assigns the use of the I/O and defines the specific Active State of the I/O.	

Copyright © 2023, Brooks Automation, Inc.

Map I/O

Example

The following examples provide an overview of the usage of I/O types and how to assign the bits. In each example provided, the Least Significant Bit is considered to be Bit-0. The hex numbering scheme is:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), F (15)

Mapping an Individual Input:

```
MAP P_GAUGE_1 DISCRETE_IN HI TO DIGITAL_IN 0X8
```

This example command maps the name P_GAUGE_1 as a DISCRETE_IN input, Active HI to I/O DIGITAL_IN #3 represented by hex #8. The input is wired to connector MISC I/O EXT_IN3 pin #4. The io_num (0X8) is determined by the example in the figure below.

SUP PI/O 16-23								SUP PI/O 8-15								SUP PI/O 0-7								
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	I/O #
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	MASK
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	Bit Weight
0				0				0				0				0				8				HEX #

Mapping an Individual Wafer Sensor:

```
MAP STN_1_WFR WAF_SEN HI TO DIGITAL_IN 0X40
```

This example command maps STN_1_WFR as a WAF_SEN input, Active HI to I/O DIGITAL_IN #6 represented by hex #40. The input is wired to connector MISC I/O EXT_IN6 pin #7. The io_num (0X40) is determined by the example in the figure below.

SUP PI/O 16-23								SUP PI/O 8-15								SUP PI/O 0-7								
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	I/O #
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	MASK
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	Bit Weight
0				0				0				0				4				0				HEX #

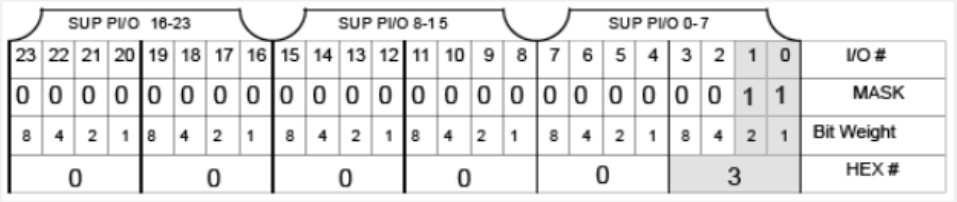
Mapping a Slot Valve Sensor:

```
MAP STN_1_SLOT SVLV_SEN HI TO DIGITAL_IN 0X3
```

This example command maps STN_1_SLOT as a SVLV_SEN input, to I/O DIGITAL_IN #0 and #1 represented by hex #3. The input is wired to connector MISC I/O EXT_IN0 and EXT_IN1 pins 1 and 2. The io_num (0X3) is determined by the example in the figure below.

Copyright © 2023, Brooks Automation, Inc.

Map I/O

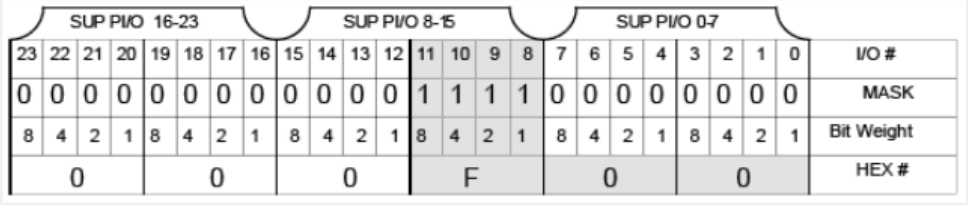


SUP PI/O 16-23								SUP PI/O 8-15								SUP PI/O 0-7								
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	I/O #
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	MASK
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	Bit Weight
0				0				0				0				0				3				HEX #

Mapping a Multiple Input Sensor:

```
MAP PRESSURE NUMERIC_IN HI TO DIGITAL_IN 0XF00
```

This example command maps PRESSURE as a NUMERIC_IN input, to I/O DIGITAL_IN lines #11, #10, #9, and #8 represented by hex # F00. The input is wired to connector MISC I/O EXT_IN8, EXT_IN9, EXT_IN10, and EXT_IN11 pins 9, 10, 11, and 2. The io_num (0XF00) is determined by the example in the figure below.



SUP PI/O 16-23								SUP PI/O 8-15								SUP PI/O 0-7								
23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	I/O #
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	MASK
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	Bit Weight
0				0				0				F				0				0				HEX #

Copyright © 2023, Brooks Automation, Inc.

Remove I/O

This command is used to remove a previously defined I/O name (defined using the MAP command) from the I/O map. Removing a name from the I/O map frees that name for redefinition. Using the ALL option will remove all existing I/O entries. Any factory default entries will be recovered (DIO_IN, DIO_OUT, and HW_EMER_STOP) or set to the default value (RETRACT_SEN, SVLV_SEN, etc.) at the next robot boot-up.

If an I/O was defined for a station sensor or a station option, first remove it at the station, then REMOVE I/O.

If the user tries to remove a non-existent I/O map item, WAVE will report an error (1002). LEAP will remove the I/O item if it exists, but will not report an error if the item does not exist.

Remove I/O Map		
Command Syntax	`REMOVE IO [ALL|name]`	
Response Syntax	There is no response to this command.	
Arguments	name:	Name that is assigned to the specified I/O.
	ALL:	All defined I/O points.
	NOTE: *At least one argument must be specified.*	
Description	Removes a previously defined I/O name (defined using the MAP command) from the I/O map.	
Example	The following example removes the name P_GAUGE_1 from the I/O map: `REMOVE IO P_GAUGE_1`	

Copyright © 2023, Brooks Automation, Inc.

Request I/O Map

This command is used to display the current I/O map. The definition of a specific named I/O may be displayed by referencing that I/O name, or all I/O names defined using the MAP command may be listed along with their definition by using the ALL specifier.

Request I/O Map		
Command Syntax	`RQ IO (MAP (ALL|NAME))|BLOCKS|TYPES [INPUTS|OUTPUTS])`	
Response Syntax	`IO MAP name type active_state TO io_name io_num`	
Arguments	name:	The name that is assigned to the specified I/O Range.
	type:	Type of I/O by its specific function.
	io_name:	Name assigned to physical I/O for device that is mapped or assigned. The I/O devices available to the robot are: DIGITAL_IN DIGITAL_OUT
	io_num:	The I/O channels that are mapped or assigned. This variable is an eight digit (maximum) hex number of the form 0x12345678 representing the specific I/O channel(s). See the examples that follow for assistance in designating the io_num. ***NOTE:*** *That leading zeros may be dropped from this number.*
	ALL:	All defined I/O points.
	NOTE: *At least one argument must be specified.*	
Description	Displays the current I/O map.	

Copyright © 2023, Brooks Automation, Inc.

Request I/O Map	

Examples	The following example requests the current definition for all specified I/O. The robot returns the current map of all named I/O including the definition of each I/O name.
	
	`RQ IO MAP ALL`
	`IO MAP P_GAUGE_1 DISCRETE_IN HI DIGITAL_IN 0x00000004`
	`IO MAP STN_1_WFR WAF_SEN HI DIGITAL_IN 0x00000040`
	`IO MAP STN_1_SLOT SBIT_SVLV_SEN DIGITAL_IN 0x00000003`
	`IO MAP PRESSURE NUMERIC_IN DIGITAL_IN 0x00000F00`
	
	`RQ IO BLOCKS`
	`DIGITAL_IN`
	`DIGITAL_OUT`
	`_RDY`
	
	`RQ IO TYPES INPUTS`
	`NUMERIC_OUT`
	`DISCRETE_OUT`
	`RETRACT_PIN`
	`RETRACT_SEN`
	`EXTEND_SEN`
	`CDM_CTRL`
	`POWER_IND`
	`MOTOR_POWER_IND`
	`STOP_FUNC_IND`
	`LIGHT_TOWER`
	`STN_FACET`
	`SYS_CTRL_ON`
	`HEART_BEAT`
	`NEAR_STN`
	`EXCL_REQ`
	`HANDOFF_OUTPUT`
	`VIBRATION_IND`
	`_RDY`
	
	The following example requests the current definition for a specific I/O point. The robot returns the current map of the I/O point named P_GAUGE_1.
	
	`RQ IO MAP P_GAUGE_1`
	`IO MAP P_GAUGE_1 DISCRETE_IN HI DIGITAL_IN 0x00000004`

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Debounce Time

Set Debounce Time		
Command Syntax		
Response Syntax		
Arguments		
Description		
Examples		

Request Debounce Time		
Command Syntax		
Response Syntax		
Arguments		
Description		
Examples		

Store Debounce Time	
Command Syntax	

Copyright © 2023, Brooks Automation, Inc.

Store Debounce Time		
Response Syntax		
Arguments		
Description		
Examples		

Set/Request I/O State

Sets the current status for the outputs and requests specified I/O state.

The set command is used to set the physical I/O by referencing the I/O names defined using the map command. The values used when issuing this command are defined by the I/O type being referenced, their settings, and a description of the settings.

This command is available for trouble-shooting or testing purposes allowing the user to toggle output bits for verification or to operate devices through the commands interface.

The request command is used to monitor the current status of physical I/O by referencing the I/O names defined using the map command. The response from issuing this command is defined by the I/O command type being referenced, refer to the following tables for a list of Operation Interlock I/O types, their RQ I/O STATE responses, and a description of the responses.

NOTE: No input signals can be set.

Set I/O State	
Command Syntax	`SET IO STATE io_name setting` *This command sets outputs only. No Input type I/O can be set. This command cannot be stored. This command is not supported in VT5 Compatibility.*
Response Syntax	`IO STATE io_name io_state`

Copyright © 2023, Brooks Automation, Inc.

Set I/O State		
Arguments	io_name:	The name assigned to physical I/O using the MAP command. Response Field Size = 20 digits maximum.
	setting:	The output value for the I/O referenced by the io_name. The output set will be defined by the type of I/O being referenced.
	io_state:	The status of the I/O referenced by the io_name. Note that the I/O state returned will be defined by the type of I/O being referenced.
Description	Sets the current status for the outputs and requests specified I/O state.	

Example	The following examples provide an overview of the settings for the different types of I/O. Note that the current arm status of the robot does not apply to this command.
	
	The current status of the I/O defined by `PUMP_CTRL` (set using the `MAP` command), which in this case is a `DISCRETE_OUT`, is set to `ACTIVE`.
	`SET IO STATE PUMP_CTRL ACTIVE`
	
	The current status of the I/O defined by `SLT_1_DRV` (set using the `MAP` command), which in this case is a `SVLV_CTRL`, is set to `CLOSE`.
	`SET IO STATE SLT_1_DRV CLOSE`	

Request I/O State		
Command Syntax	`RQ IO STATE` *`io_name`*	
Response Syntax	`IO STATE` *`io_name io_state`*	
Arguments	io_name:	Requests the name assigned to the physical I/O using the MAP command.
	io_state:	The status of the I/O referenced by the io_name. Note that the I/O state returned will be defined by the type of I/O being referenced.
Description	Requests the current status for the outputs and requests specified I/O state.	

Copyright © 2023, Brooks Automation, Inc.

Request I/O State	

Example	The following example returns the current status of the I/O defined by `P_GAUGE_1` (set using the `MAP` command), which in this case is a pressure sensor, indicating that there is a high pressure condition.
	`RQ IO STATE P_GAUGE_1 IO STATE P_GAUGE_1 ACTIVE`
	
	The following example returns the current status of the I/O defined by `STN_1_WFR` (set using the `MAP` command), which in this case is a wafer sensor, indicating that there is a wafer present.
	`RQ IO STATE STN_1_WFR`
	`IO STATE STN_1_WFR BLOCKED`
	
	The following example returns the current status of the I/O defined by `STN_1_WFR` (set using the `MAP` command), which in this case is a slot valve, indicating that the valve is open.
	`RQ IO STATE STN_1_SLT`
	`IO STATE STN_1_SLT OPEN`
	
	The following example returns the current status of the I/O defined by `PRESSURE` (set using the `MAP` command), which in this case is a pressure sensor, indicating that the pressure is 14.
	`RQ IO STATE PRESSURE`
	`IO STATE PRESSURE 14`
	
	The following example returns the current status of all named I/O.
	`RQ IO STATE ALL`
	`IO STATE P_GAUGE_1 ACTIVE`
	`IO STATE STN_1_WFR BLOCKED`
	`IO STATE STN_1_SLT OPEN`
	`IO STATE PRESSURE 14`

Set/Request/Store Event I/O

Sets, requests, and stores specific events. This command allows enabling or disabling specific events. An event that is enabled will generate an _EVENT message whenever the event occurs while an event that is disabled will not. It can be used to monitor the system for specific events only by enabling them and disabling all other events.

Set Event I/O		
Command Syntax	`SET EVENT IO name eventNo (ENB|DIS)`	
Response Syntax	There is no response to this command.	
Arguments	name:	The name of the IO device.
	eventNo:	The number of the event.
	ENB|DIS:	Assigns the same polarity of the input bit to the output bit.

Copyright © 2023, Brooks Automation, Inc.

Set Event I/O		
Description	Sets specific events.	
Example	The following example enables event 2100 (Input Valid bit is on) for E84 device MISCIO: `SET EVENT IO MISCIO 2100 ENB`	

Request Event I/O		
Command Syntax	`RQ EVENT IO name [eventNo|(ALL)]`	
Response Syntax	`IO name EVENT eventNo status`	
Arguments	name:	The name of the IO device being requested.
	eventNo:	The number of the event being requested.
	ALL	Returns the number of all events.
	ENB|DIS:	Assigns the same polarity of the input bit to the output bit.
Description	Requests specific events.	

Example	The following example requests the status of event 2000 for Test20:
	`RQ EVENT IO TEST20 2000`
	`IO TEST20 EVENT 2000 ENB`
	The following example requests the status of all events for Test20:
	`RQ EVENT IO TEST20 ALL`
	`IO TEST20 EVENTS 2000 ENB 2001 ENB 2002 ENB . . .`	

Store Event I/O		
Command Syntax	`STORE EVENT IO name [eventNo|(ALL)]`	
Response Syntax	`IO name EVENT eventNo status`	
Arguments	name:	The name of the IO device.
	eventNo:	The number of the event being requested.
	ENB|DIS:	Assigns the same polarity of the input bit to the output bit.
Description	Stores specific events.	

Copyright © 2023, Brooks Automation, Inc.

Store Event I/O	
Example	The following example enables event 2100 (Input Valid bit is on) for E84 device MISCIO: `STORE EVENT IO MISCIO 2100 ENB`

Map Passthrough

This command is used to pass information through the robot from valves, etc. The map passthrough command allows the flexibility to send single bit to single bit information through the robot. This command also assigns the specific characteristics of the I/O.

The I/O types are defined as performing specific functions with the settings and responses defined by those functions. Map the IN bit to the same OUT bit.

Map Passthrough		
Command Syntax	`MAP name PASSTHROUGH DIGITAL_IN io_num_in (TO|NOT) DIGITAL_OUT` *`io_num_out`*	
Response Syntax	There is no response to this command.	
Arguments	name:	Specifies the user reference name to be assigned to the specified I/O (20 characters maximum).
	io_num:	Specifies the I/O channels being mapped or assigned. This variable is an eight digit (maximum) hex number of the form 0x12345678 representing the specific I/O channel(s). See the examples in the Map I/O command for assistance in designating the io_num. Note that leading zeros may be dropped from this number.
	TO:	Assigns the same polarity of the input bit to the output bit.
	NOT:	Assigns the opposite polarity of the input bit to the output bit (inverts bits).
Description	Passes information through the robot from valves, etc.	

Copyright © 2023, Brooks Automation, Inc.

Map Passthrough	

Example	Same Polarity:
	
	`MAP DIG_1 PASSTHROUGH DIGITAL_IN 0X40 TO DIGITAL_OUT 0X40`
	
	In this same polarity example, the input that is high is passed through the robot and output as high.
	
	Opposite polarity:
	
	`MAP DIG_1 PASSTHROUGH NOT DIGITAL_IN 0X40 TO DIGITAL_OUT 0X40`
	
	In this opposite polarity example, the input that is high is passed through the robot and output as low.

Copyright © 2023, Brooks Automation, Inc.

Extended I/O Mode

Extended I/O Mode is the second I/O Mode supported by LEAP. It allows more signals to be dedicated as inputs than Standard I/O Mode. This mode is only supported by LEAP robots. Extended I/O has 32 input signals and 10 output signals.

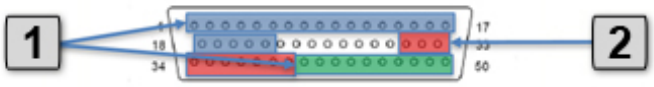


Figure 4-3: Extended I/O Mode

1	32 Inputs (22 + 10 additional)
2	10 Outputs

Extended I/O is very similar to Standard I/O except for the allocation of input signals to output signals. Extended I/O input pins are configured and read in in the exact same manner as Standard Mode.

Extended I/O Mode is configured using specific options selection in the CONFIG IO DRIVE command.

After selecting Extended I/O Mode, the robot must be rebooted in order for this mode to configure the signals appropriately.

Extended I/O Mode Pinout

Pin #	Signal Name	Signal Description	I/O Mask	Notes
1	IN_0	INPUT 1	0x01	
2	IN_1	INPUT 2	0x02	
3	IN_2	INPUT 3	0x04	
4	IN_3	INPUT 4	0x08	
5	IN_4	INPUT 5	0x10	
6	IN_5	INPUT 6	0x20	
7	IN_6	INPUT 7	0x40	
8	IN_7	INPUT 8	0x80	
9	IN_8	INPUT 9	0x100	

Copyright © 2023, Brooks Automation, Inc.

Pin #	Signal Name	Signal Description	I/O Mask	Notes
10	IN_9	INPUT 10	0x200	
11	IN_10	INPUT 11	0x400	
12	IN_11	INPUT 12	0x800	
13	IN_12	INPUT 13	0x1000	
14	IN_13	INPUT 14	0x2000	
15	IN_14	INPUT 15	0x4000	
16	IN_15	INPUT 16	0x8000	
17	IN_16	INPUT 17	0x10000	
18	IN_17	INPUT 18	0x20000	
19	IN_18	INPUT 19	0x40000	
20	IN_19	INPUT 20	0x80000	
21	IN_20	INPUT 21	0x100000	
22	IN_21	INPUT 22	0x200000	
23	INT_LCK 0	INTLCK 0	-	CONNECT TO INTLCK_1 (PIN 24) - See note
24	INT_LCK 1	INTLCK 1	-	CONNECT TO INTLCK_0 (PIN 23) - See note
25	+PWR_ISOL	+PWR_ISOL	-	
26	+PWR_ISOL	+PWR_ISOL	-	CONNECT TO +24 VDC (PIN 30) -See note
27	PWR RET ISOL	PWR RET ISOL	-	CONNECT TO 24V RET (PIN 29) -Note
28	PWR RET ISOL	PWR RET ISOL	-	
29	24V RET	24V RET	-	CONNECT TO PWR RET ISOL (PIN 27) - Note
30	+24 VDC	+24 VDC	-	CONNECT TO +PWR_ISOL (PIN 26) - Note
31	OUT_0	OUTPUT 1	0x01	
32	OUT_1	OUTPUT 2	0x02	
33	OUT_2	OUTPUT 3	0x04	
34	OUT_3	OUTPUT 4	0x08	
35	OUT_4	OUTPUT 5	0x10	
36	OUT_5	OUTPUT 6	0x20	
37	OUT_6	OUTPUT 7	0x40	

Copyright © 2023, Brooks Automation, Inc.

Pin #	Signal Name	Signal Description	I/O Mask	Notes
38	OUT_7	OUTPUT 8	0x80	
39	OUT_8	OUTPUT 9	0x100	
40	OUT_9	OUTPUT 10	0x200	
41	IN_22	INPUT 23	0x400000	
42	IN_23	INPUT 24	0x800000	
43	IN_24	INPUT 25	0x1000000	
44	IN_25	INPUT 26	0x2000000	
45	IN_26	INPUT 27	0x4000000	
46	IN_27	INPUT 28	0x8000000	
47	IN_28	INPUT 29	0x10000000	
48	IN_29	INPUT 30	0x20000000	
49	IN_30	INPUT 31	0x40000000	
50	IN_31	INPUT 32	0x80000000	

NOTE: There are numerous connections possible for these pins - please consult the *MagnaTran LEAP User Manual* for other configurations. Configuration shown is the simplest to support measurable outputs from the robot.

Configure Miscellaneous I/O Port

The miscellaneous I/O port can be configured in four ways:

a. High-side standard
b. Low-side standard
c. High-side extended
d. Low-side extended.

For more information, see the Misc I/O chapter of the *MagnaTran LEAP User Manual*. Note that the options HIGH_SIDE_EXT and LOW_SIDE_EXT are used to select the extended I/O mode.

Copyright © 2023, Brooks Automation, Inc.

Configure Miscellaneous I/O Port		
Command Syntax	`CONFIG IO DRIVE [HIGH_SIDE][LOW_SIDE][HIGH_SIDE_EXT][LOW_SIDE_EXT]`	
Response Syntax	There is no response to this command.	
Arguments	option:	HIGH_SIDE_EXT LOW_SIDE_EXT HIGH_SIDE LOW_SIDE
Description	Configures the inputs and outputs on the MISC I/O connector.	

Example	The following example puts the robot in High Side Extended I/O Mode:
	`CONFIG IO DRIVE HIGH_SIDE_EXT`
	NOTE: *The robot must be reset for this change to take place.*
	
	`RQ IO MAP ALL`
	`IO MAP TEST_IN NUMERIC_IN HI DIGITAL_IN 0x0000001c`
	`IO MAP dio_in NUMERIC_IN HI DIGITAL_IN 0xffffffff`
	`IO MAP dio_out NUMERIC_OUT HI DIGITAL_OUT 0x000003ff`
	`IO MAP hw_emer_stop EMER_STOP LOW DIGITAL_IN 0x100000000`
	NOTE: *The 3ff result shows extended mode.*	

DIO I/O Mode

DIO Mode is the third I/O mode supported by LEAP. It is a mode that sets predetermined outputs to show the state of current robot motion. When this mode is chosen by the user, certain output signals are set according to a specific list.

DIO Mode is like DIO Mode in WAVEII Version 3.5N, but several extensions have been made on LEAP to accommodate robots with two independent arms. The refresh rate of the signals in this mode is like the WAVEII version.

When DIO mode is entered, the definition of the output signals from Standard I/O Mode is changed. The specific output signals are detailed in the next section. Note that the input signals in DIO Mode work the same as in Standard I/O Mode.

If DIO Mode is desired after every power on boot up, the STORE DIO command must be used to store DIO Mode as the permanently stored I/O Mode.

High Side/Low Side selection for this mode is the same as for Standard I/O Mode.

Copyright © 2023, Brooks Automation, Inc.

DIO Signal Definitions

This section provides a description of the output signals when DIO Mode is selected. Note the input signals need to be defined in the same fashion as in Standard I/O Mode.

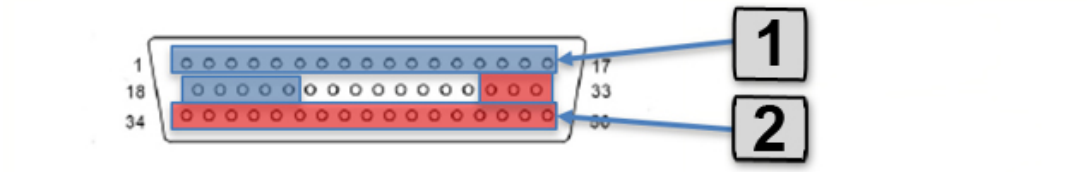


Figure 4-4: Standard I/O Input and Output Signals

1	22 Inputs
2	20 Outputs

18 bits are used to represent different states of the robot. These are as follows:

- bit 18 = SERVO ON
- bit 17 = DISC CNTRL (Always zero since only outputs are being used)
- bit 16 = ERROR NOT
- bit 15-13 = 3 bit error code
- bit 12 =REFERENCED (Always true (absolute encoders))
- bit 11 = NOT MOVING
- bit 10 = ARM IN USE
- bit 08 = AT STATION
- bit 08-04 = 5 Bit station number (Station number -1) Station 1 = 0, 31 = 30
- bit 03-02 R POS (NOT AT STATON/RE/EX)
- bit 01-00 Z POS (NOT AT STATION/DN/UP)

Pin #	Signal Name	Signal Description	I/O Mask	Notes
1	IN_0	INPUT 1	0x01	
2	IN_1	INPUT 2	0x02	
3	IN_2	INPUT 3	0x04	
4	IN_3	INPUT 4	0x08	
5	IN_4	INPUT 5	0x10	
6	IN_5	INPUT 6	0x20	
7	IN_6	INPUT 7	0x40	
8	IN_7	INPUT 8	0x80	

Copyright © 2023, Brooks Automation, Inc.

Pin #	Signal Name	Signal Description	I/O Mask	Notes
9	IN_8	INPUT 9	0x100	
10	IN_9	INPUT 10	0x200	
11	IN_10	INPUT 11	0x400	
12	IN_11	INPUT 12	0x800	
13	IN_12	INPUT 13	0x1000	
14	IN_13	INPUT 14	0x2000	
15	IN_14	INPUT 15	0x4000	
16	IN_15	INPUT 16	0x8000	
17	IN_16	INPUT 17	0x10000	
18	IN_17	INPUT 18	0x20000	
19	IN_18	INPUT 19	0x40000	
20	IN_19	INPUT 20	0x80000	
21	IN_20	INPUT 21	0x10000	
22	IN_21	INPUT 22	0x20000	
23	INT_LCK 0	INTLCK 0	-	CONNECT TO INTLCK_1 (PIN 24) - See note
24	INT_LCK 1	INTLCK 1	-	CONNECT TO INTLCK_0 (PIN 23) - See note
25	+PWR_ISOL	+PWR_ISOL	-	
26	+PWR_ISOL	+PWR_ISOL	-	CONNECT TO +24 VDC (PIN 30) - See note
27	PWR RET ISOL	PWR RET ISOL	-	CONNECT TO 24V RET (PIN 29) - See note
28	PWR RET ISOL	PWR RET ISOL	-	
29	24V RET	24V RET	-	CONNECT TO PWR RET ISOL (PIN 27) - See note
30	+24 VDC	+24 VDC	-	CONNECT TO +PWR_ISOL (PIN 26) - See note
31	Z_POS_0	Z_POS_0	-	00 = ROBOT IS NOT AT SET POSITION
32	Z_POS_1	Z_POS_1	-	01 = ROBOT IS AT LOWER POSITION
-			-	10 = ROBOT IS AT Z UP POSITION
33	R_POS_0	R_POS_0	-	00 = ROBOT IS NOT AT SET POSITION
34	R_POS_1	R_POS_1	-	01 = ROBOT IS AT RETRACT POSITION
-			-	10 = ROBOT IS AT EXTEND POSITION
35	STN_0	STATION_0	-	See Table 4-2
36	STN_1	STATION_1	-	See Table 4-2

Copyright © 2023, Brooks Automation, Inc.

Pin #	Signal Name	Signal Description	I/O Mask	Notes
37	STN_2	STATION_2	-	See Table 4-2
38	STN_3	STATION_3	-	See Table 4-2
39	STN_4	STATION_4	-	See Table 4-2
40	AT_STN	AT STATION	-	0 = NOT AT STATION, 1 = AT STATION
41	ARM_IN_USE	ARM IN USE	-	0 = ARM A IN USE, 1 = ARM B IN USE
42	NOT_MOVING	NOT MOVING	-	0 = MOVING, 1 = NOT MOVING
43	REFERENCED	REFERENCED	-	ALWAYS 1 FOR LEAP ROBOTS
44	ERROR_0	ERROR_0	-	ALWAYS 0
45	ERROR_1	ERROR_1	-	ALWAYS 0
46	ERROR_2	ERROR_2	-	ALWAYS 0
47	ERROR_NOT	ERROR NOT	-	0 = Any error, 1 = No error
48	DIO_CNTRL	DIO CONTROL	-	Always 0 when DIO mode is selected
49	SERVO_PWR	SERVO POWER	-	0 = Servos Off, 1 = Servos On
50	UNUSED	NOT USED	-	

NOTE: There are numerous connections possible for these pins - please consult the *MagnaTran LEAP User Manual* for other configurations. Configuration shown is the simplest to support measurable outputs from the robot.

Table 4-2: DIO Mode Station Values

STN_4	STN_3	STN_2	STN_1	STN_0	SELECTED STATION
0	0	0	0	0	STATION #1
0	0	0	0	1	STATION #2
0	0	0	1	0	STATION #3
0	0	0	1	1	STATION #4
0	0	1	0	0	STATION #5
0	0	1	0	1	STATION # 6
0	0	1	1	0	STATION # 7
0	0	1	1	1	STATION # 8
0	1	0	0	0	STATION # 9

Copyright © 2023, Brooks Automation, Inc.

STN_4	STN_3	STN_2	STN_1	STN_0	SELECTED STATION
0	1	0	0	1	STATION # 10
0	1	0	1	0	STATION # 11
0	1	0	1	1	STATION # 12
0	1	1	0	0	STATION # 13
0	1	1	0	1	STATION # 14
0	1	1	1	0	STATION # 15
0	1	1	1	1	STATION # 16
1	0	0	0	0	STATION # 17
1	0	0	0	1	STATION # 18
1	0	0	1	0	STATION # 19
1	0	0	1	1	STATION # 20
1	0	1	0	0	STATION # 21
1	0	1	0	1	STATION # 22
1	0	1	1	0	STATION # 23
1	0	1	1	1	STATION # 24
1	1	0	0	0	STATION # 25
1	1	0	0	1	STATION # 26
1	1	0	1	0	STATION # 27
1	1	0	1	1	STATION # 28
1	1	1	0	0	STATION # 29
1	1	1	0	1	STATION # 30
1	1	1	1	0	STATION # 31
1	1	1	1	1	STATION # 32

NOTE: Station numbers above 32 are not supported by DIO mode.

Copyright © 2023, Brooks Automation, Inc.

DIO I/O Mode Commands

This DIO Mode commands are listed in this section.

Request DIO Output		
Command Syntax	`RQ DIO OUTPUT`	
Response Syntax	`DIO OUTPUT [YES|NO]`	
Arguments	Y:	DIO Output has been enabled.
	N:	DIO Output has been disabled.
Description	Requests the current mode set for the enabling or disabling of the Discrete I/O Output while the serial I/O is in control of the robot.	

Example	The following example requests the current Discrete I/O Output function mode and the response reports enabled.
	
	`RQ DIO OUTPUT`
	`DIO OUTPUT YES`	

Set DIO Output		
Command Syntax	`SET DIO OUTPUT [Y|N]` `Default state is N`	
Response Syntax	There is no response to this command.	
Arguments	Y:	Enables DIO Output.
	N:	Disables DIO Output.
Description	Turns on the Discrete Output (DIO) output monitoring function while in serial mode. The robot may be controlled and monitored using discrete I/O lines instead of using the serial communications link. In normal operation, the serial control is disabled when the Discrete I/O control is on. This function allows enabling or disabling of the Discrete I/O Output while the serial I/O is in control of the robot.	
Example	The following example turns on the Discrete I/O Output function: `SET DIO OUTPUT Y`	

Copyright © 2023, Brooks Automation, Inc.

Store DIO Output		
Command Syntax	`STORE DIO OUTPUT`	
Response Syntax	There is no response to this command.	
Arguments	None	
Description	Stores the current function that allows enabling or disabling of the Discrete I/O Output while the serial I/O is in control of the robot.	
Example	The following example stores the Discrete I/O Output function: `STORE DIO OUTPUT`	

Start/Stop DIO		
Command Syntax	`DIO [START | STOP]`	
Response Syntax	There is no response to this command.	
Arguments	START:	Generates return message.
	STOP:	Generates return message.
Description	For backwards compatibility for WAVE. Command will do nothing but respond until DIO Inputs are used.	

Example	The following example turns on the Discrete I/O Output function:
	`DIO START`
	`DIO STOP`	

Copyright © 2023, Brooks Automation, Inc.

DIO I/O Mode Errors

All error codes must map to the following error codes represented by 3 bits.

DIO_ERR_NONE	0
DIO_ERR_HOME_SYNC	2
DIO_ERR_MOTION	3
DIO_ERR_SUBSTRATE	5
DIO_ERR_VALVE_INT	6
DIO_ERR_OTHER	7

Any error code that is not found in the mapping will map to the OTHER error code.

DIO Commands

Output Signal Timing

Bit level changes will occur within 50 ms relative to the start of an associate motion command.

DIO OUTPUT Refresh Rate

DIO mode on MagnaTran LEAP was modeled after the WAVE II_3.5N to have a refresh rate of 50 ms.

Existing Inputs Mappings and Functionality

Existing Inputs mappings and functionality must remain as is except for any associated outputs when DIO Outputs mode is active. An example would be the Stop Function Indicator. All associated inputs (e-stops) will be process as is but the associated mappings to the outputs will be blocked.

Copyright © 2023, Brooks Automation, Inc.

5. Robot Setup

Each robot has a configuration ID that is used to set all of the robot's specific tuning parameters. This ID can be found in the robot's bill of materials and is usually set at the factory. Be careful when modifying this ID because an incorrect ID can cause the robot to malfunction.

Request Configuration

Request Configuration	
Command Syntax	`RQ CONFIG`
Response Syntax	`ROBOT CONFIG ID: application`
Arguments	None
Description	Requests the current configuration of the robot.
Example	`RQ CONFIG` `ROBOT CONFIG ID: G005-K004-F001-MS001-W002-L2150`

Copyright © 2023, Brooks Automation, Inc.

Request Configuration File Version

Request Configuration File Version	
Command Syntax	`RQ CONFIG FILE VERSION`

Response Syntax	`Product Name : product name`
	`Part Number : number`
	`Version : version`
	`Build Date : date in mm/dd/yyy`
	`Build Time : time`

Arguments	None
Description	Requests the configuration file name, version, and build date installed on the robot. The file version request command is used to request version information of .BIN configuration files.

Example	`RQ CONFIG FILE VERSION`
	`Product Name : LEAP`
	`Part Number : 002-5089-01`
	`Version : CF2.3`
	`Build Date : 04/08/2002`
	`Build Time : 16:17:27`

Copyright © 2023, Brooks Automation, Inc.

Configure/Request Robot Application

Configures and requests application information from the firmware into the robot for use during normal operations for a specific robot. The config command will automatically store the new parameter.

Brooks Automation assigns an Application Number to each robot type. This number ensures the correct performance of the robot according to its configuration; that is, its arm length, arm weight, and the radial displacement in microns.

An example ID number is: G001-C001-P001-DS001-W001-L3100M.

This command resets, or changes, various robot parameters for a specific arm set. After the firmware has been loaded, a "Reset" must be performed.

NOTE: This command is sometimes used when installing a new version of firmware or replacing specific hardware. All instructions for this command are not included in this manual.

NOTE: This command is NOT used in the normal operation of the robot. Call Brooks Automation Technical Support for instructions on the correct use of this command. For more information on contacting Brooks Technical Support, please see "Contact Brooks Automation Technical Support".

Configure Robot Application		
Command Syntax	`CONFIG ROBOT APPLIC application_number`	
Response Syntax	There is no response to this command.	
Arguments	application_number:	Specific number for the robot.
Description	Configures the application information from the firmware into the robot for use during normal operations for a specific robot.	
Example	`CONFIG ROBOT APPLIC G042-K022-F010-B001-W007-L2050`	

Copyright © 2023, Brooks Automation, Inc.

Request Robot Application		
Command Syntax	`RQ ROBOT APPLIC`	
Response Syntax	`ROBOT APPLIC G###-C###-P###-DS####-W###-L####M`	
Arguments	application_number:	Specific number for the robot.
Description	Requests the application information from the firmware into the robot for use during normal operations for a specific robot.	
Example	`RQ ROBOT APPLIC` `G001-C001-P001-DS001-W001-L3100M`	

Find/Set/Request/Store Phase

The SET SYNC PHASE command is used to enter the average calculated value from the FIND PHASE command.

The FIND PHASE command may be aborted with <CTRL><C> at the user keyboard.

Finds, sets, requests, and stores a Find Phase on individual or all motor phase linkages.

> ***NOTE:*** This command is NOT used in the normal operation of the robot. Call Brooks Automation Technical Support for instructions on the correct use of this command. For more information on contacting Brooks Technical Support, please see "Contact Brooks Automation Technical Support".

Find Phase	
Command Syntax	`FIND PHASE [ALL| (R|T|Z)` *R is in FIND command only. FIND does not separate the theta axes. In VT5 Compatibility, you must enter the options.* ***NOTE:*** *There is a 30 second delay after the command is entered before a result is returned.*
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Find Phase		
Arguments	ALL:	Finds all axis references.
	T:	Finds theta axes values.
	R:	Finds the radial axis value.
	Z:	Finds the Z-axis value.
Description	Finds a Find Phase on individual or all motor phase linkages.	

Example	`FIND PHASE ALL`
	`Calibrate motor T1:0 at port [0] => From 0 to 1.5708`
	`Calibrate motor T2:0 at port [1] => From 0 to 1.5708`
	`Calibrate motor Z:0 at port [3] => From 0 to 1.5708`	

Set Sync Phase		
Command Syntax	`SET SYNC PHASE (T1 value|T2 value|Z value)`	
Response Syntax	There is no response to this command.	
Arguments	T1 value:	Theta axis outer shaft value
	T2 value:	Theta axis inner shaft value
	Z value:	Z-axis value
Description	Sets a Find Phase on individual or all motor phase linkages.	
Example	`SET SYNC PHASE T1 1.570796`	

Request Sync Phase		
Command Syntax	`RQ SYNC PHASE [[ALL]|T1|T2|Z]`	
Response Syntax	`RQ SYNC ALL` `SYNC PHASE t1-value t2-value z-value`	
Arguments	T1 value:	Theta axis outer shaft value.
	T2 value:	Theta axis inner shaft value.
	Z value:	Z-axis value.
Description	Requests a Find Phase on individual or all motor phase linkages.	
Example	`RQ SYNC PHASE ALL` `SYNC PHASE 0.953842 2.309949 0.868019`	

Copyright © 2023, Brooks Automation, Inc.

Find/Set/Request/Store Zero

Finds, sets, requests, and stores the Zero or Home reference for the specifies axis. The Home position may be changed from the factory settings to accommodate the users requirements.

NOTE: This command is NOT used in the normal operation of the robot. Call Brooks Automation Technical Support for instructions on the correct use of this command. For more information on contacting Brooks Technical Support, please see "Contact Brooks Automation Technical Support".

This command normally requires the Arm Shipping Fixture to be installed on the armset. However, this command may be used without the fixture. Enter the command. Move the arms so that the robot's end effector A points in the direction that will be defined as zero. The arms must be moved at least 15° within 20 seconds. Hold the arms in place or against the Homing Fixture until the Ready prompt appears. Store the new values.

Find Zero		
Command Syntax	`FIND ZERO (T|Z)` R is not used in this command. FIND does not separate the theta axes.	

Response Syntax	`SYNC ZERO { motor_zero }`
	or
	`SYNC ZERO {t1_value | t2_value | t3_value | z_value }`	

Arguments	T: Z:	Find theta axis zero values. Find Z-axis zero value.
Description	Finds the Zero or Home reference for the specifies axis.	
Example	`FIND ZERO T`	

Set Sync Zero	
Command Syntax	`SET SYNC ZERO (T1 position|T2 position|Z position)`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Sync Zero		
Arguments	T1 position:	Find theta axis zero values.
	T2 position:	Find Z-axis zero value.
	Z position:	Z-axis value.
Description	Sets the Zero or Home reference for the specifies axis. ***NOTE:*** *This command should be used under the direction of Brooks Service because an incorrect value could cause the arm to hard track.*	
Example	`SET SYNC ZERO T1 - 6.23185`	

Request Sync Zero		
Command Syntax	`SET SYNC ZERO [ALL|(T1|T2|Z`	
Response Syntax	`RQ SYNC [[ALL]|[t1_value|t2_value|t3_value|z_value]`	
Arguments	ALL:	Find theta axis zero values.
	T1 position:	Find Z-axis zero value.
	T2 position:	Z-axis value
	Z:	Requests the Z-axis value.
Description	Requests the Zero or Home reference for the specifies axis.	
Example	`RQ SYNC ZERO ALL` `SYNC ZERO 4.88845 -4.49423 -5462.65`	

Store Sync Zero		
Command Syntax	`STORE SYNC ZERO [ALL| (T1|T2|Z)]` R is not used in this command. FIND does not separate the theta axes.	
Response Syntax	There is no response to this command.	
Arguments	ALL:	Stores all axis references.
	T1 position:	Stores the theta axis outer shaft value.
	T2 position:	Stores the theta axis inner shaft value.
	Z:	Stores the Z-axis value.
Description	Sets the Zero or Home reference for the specifies axis.	
Example	`RQ SYNC ZERO ALL` `SYNC ZERO 4.88845 -4.49423 -5462.65`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request Interlock T/Z Motion

Sets and requests the state of specific interlocking capabilities. In normal operation, all T and Z moves are performed simultaneously.

`ON`: R-axis, then T and W at the same time, then Z.

`OFF`: R-axis, then T, W and Z at the same time.

NOTE: This command cannot be stored.

Set Interlock T/Z Motion		
Command Syntax	`SET (INTLCK) TZ (ON|OFF)`	
Response Syntax	`TZ ------- (ON|OFF)`	
Arguments	TZ:	Sets the T/Z motion mode. ON : The robot will execute T moves before performing Z moves. OFF : The robot will execute T and Z moves simultaneously.
Description	Sets the state of specific interlocking capabilities. In normal operation, all T and Z moves are performed simultaneously.	
Example	`SET INTLCK TZ ON`: To move the T-axis, then the Z-axis. `SET INTLCK TZ OFF`: To move the T-axis and the Z-axis simultaneously.	

Request Interlock T/Z Motion	
Command Syntax	`RQ INTLCK [WAF_SEN] [TZ] [LOAD] [DEFEAT_SAFETY] [ALL]`
Response Syntax	`INTLCK TZ OFF/ON`

Copyright © 2023, Brooks Automation, Inc.

Request Interlock T/Z Motion		
Arguments	TZ:	The TZ option is used to support *PICK* and *PLACE* commands to flipped stations. With this option enabled, when moving to the station, T-axis move is performed before Z-axis move.
	WAF_SEN:	Y|N
	LOAD:	Lists compatibility type.
	DEFEAT_SAFETY	Y|N
	ALGN_WF_PRS	Y|N
	ALL	Lists all options.
Description	Requests robot interlocks for wafer sensing. In normal operation, all T and Z moves are performed simultaneously.	
Example	`RQ INTLCK ALL` `INTLCK WAF_SEN Y TZ OFF LOAD WAVE DEFEAT_SAFETY N ALGN_WFR_PRS N`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Home Position Z-axis

Sets, requests, and stores the Z-axis Home position. The Z-axis HOME position can be changed through a command line entry. Stations are taught based on zero Z. Changing the HOME Z value does not affect station parameters.

> ***NOTE:*** This command is NOT used in the normal operation of the robot. Call Brooks Automation Technical Support for instructions on the correct use of this command. For more information on contacting Brooks Technical Support, please see "Contact Brooks Automation Technical Support".

Set Home Position Z-axis		
Command Syntax	`SET HOME POS Z z_value`	
Response Syntax	There is no response to this command.	
Arguments	z_value:	Absolute position value in microns.
Description	Sets the Z-axis Home position.	
Example	`SET HOME POS Z 17500`	

Request Home Position Z-axis		
Command Syntax	`RQ HOME POS Z`	
Response Syntax	`HOME POS ABS Z z_value`	
Arguments	z_value:	Absolute position value in microns.
Description	Requests the Z-axis Home position.	
Example	`RQ HOME POS Z` `HOME POS ABS Z 17500`	

Store Home Position Z-axis	
Command Syntax	`STORE HOME POS Z`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores the Z-axis Home position.
Example	`STORE HOME POS Z`

Copyright © 2023, Brooks Automation, Inc.

Set Brake

Sets the brake to control the specified axis. This command is used for troubleshooting and maintenance purposes only. With the brake OFF, manual movement of the axis is permitted.

NOTE: This command is NOT used in the normal operation of the robot. Call Brooks Automation Technical Support for instructions on the correct use of this command. For more information on contacting Brooks Technical Support, please see "Contact Brooks Automation Technical Support".

NOTE: Using the ZBRAKE command will cause the Z-axis to fall. Brace something underneath the armset to prevent free-fall.

Set Brake		
Command Syntax	`SET ZBRAKE (ON|OFF)`	
Response Syntax	There is no response to this command.	
Arguments	state:	ON: Activates the brake . OFF: Releases the brake.
Description	Sets the brake to control the specified axis.	
Example	`SET ZBRAKE ON`	

Request/Set/Store Resolution

Requests, sets, and stores the theta resolution. HI will allow theta resolution of two decimal places, i.e. 90000.45. The default resolution is LOW.

Request Resolution	
Command Syntax	`RQ RESOLUTION`
Response Syntax	`RESOLUTION LOW`
Arguments	None
Description	Requests the theta resolution.
Example	`RQ RESOLUTION` `RESOLUTION HIGH`

Copyright © 2023, Brooks Automation, Inc.

Set Resolution		
Command Syntax	`SET RESOLUTION [HIGH|LOW]`	
Arguments	HIGH:	Set high theta resolution
	LOW:	Set low theta resolution
Response Syntax	There is no response to this command.	
Description	Sets the theta resolution.	
Example	`SET RESOLUTION HIGH`	

Store Resolution	
Command Syntax	`STORE RESOLUTION`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores the theta resolution.
Example	`STORE RESOLUTION`

Fan Commands

NOTE: All robot configurations have Fan 1, but only larger configurations have Fan 2.

Request Fan Mode		
Command Syntax	`RQ FAN MODE` *fanID*	
Response Syntax	`FAN MODE (ON|OFF)`	
Arguments	fanID:	FAN1 – The fan connected to FAN1 connector FAN2 – The fan connected to FAN2 connector
Description	Requests fan operational or configuration. ON: The fan is enabled. OFF: The fan is disabled.	
Example	`RQ FAN MODE FAN1` `FAN MODE ON`	

Copyright © 2023, Brooks Automation, Inc.

Request Fan Speed		
Command Syntax	`RQ FAN SPEED` *`fanID`*	
Response Syntax	`FAN SPEED` *`speed`*	
Arguments	fanID:	FAN1 – The fan connected to FAN1 connector FAN2 – The fan connected to FAN2 connector
	speed:	The fan current speed in RPM Values around or under 1000 RPM indicate the fan has stopped or that no fan is connected. A value of 0 is reported when the software is unable to communicate with the fan controller chip.
Description	Reports the speed of the fan while running.	
Example	`RQ FAN SPEED FAN1` `FAN SPEED 960`	

Request Fan Status		
Command Syntax	`RQ FAN STATUS` *`fanID`*	
Response Syntax	`FAN STATUS (UNKNOWN|STOPPED|STALLED|ON)`	
Arguments	fanID:	FAN1 – The fan connected to FAN1 connector FAN2 – The fan connected to FAN2 connector ***NOTE:*** *Many robots do not have FAN2, and will show FAN2 as stopped.*
	UNKNOWN:	Unable to communicate with the fan controller chip.
	STOPPED:	The fan is stopped, disconnected, or running under 1200 RPM.
	STALLED:	The fan is running above 1200 RPM but below the stalled threshold speed.
	ON:	The fan is running above 1200 RPM and above the stalled threshold speed.
Description	Reports the speed-status for a fan.	
Example	`RQ FAN STATUS FAN1` `FAN STATUS STOPPED`	

Request Fan Stalled Threshold	
Command Syntax	`RQ FAN STALLED_THRESHOLD` *`fanID`*

Copyright © 2023, Brooks Automation, Inc.

Request Fan Stalled Threshold		
Response Syntax	`FAN STALLED_THRESHOLD speed`	
Arguments	fanID:	FAN1 – The fan connected to FAN1 connector FAN2 – The fan connected to FAN2 connector
	speed:	The fan stalled speed threshold, in RPM
Description	Reports the stalled speed threshold for a fan.	
Example	`RQ FAN STALLED_THRESHOLD FAN1` `FAN STALLED_THRESHOLD 4500`	

Set Fan Stalled Threshold		
Command Syntax	`SET FAN STALLED_THRESHOLD fanID speed`	
Response Syntax	There is no response to this command.	
Arguments	fanID:	FAN1 – The fan connected to FAN1 connector FAN2 – The fan connected to FAN2 connector
	speed:	The fan stalled speed threshold, in RPM
Description	Sets the stalled speed threshold for a fan.	
Example	`SET FAN STALLED_THRESHOLD FAN1 4400`	

Set Fan Status Event		
Command Syntax	`SET EVENT FAN eventID (ENB|DIS)`	
Response Syntax	There is no response to this command.	
Arguments	eventID:	4001: ON 4002: STALLED 4003: STOPPED 4004: UNKNOWN
Description	Enables or disables a fan status event for a fan.	
Example	`SET EVENT FAN 4003 ENB`	

Request Fan Status Event		
Command Syntax	`RQ EVENT FAN (eventID|ALL)`	
Response Syntax	`FAN EVENT eventID (ENB|DIS)`	
Arguments	eventID:	See Event IDs in "Set Fan Status Event".

Copyright © 2023, Brooks Automation, Inc.

Request Fan Status Event	
Description	Reports the status of the fan events for a fan.

Example	`RQ EVENT FAN ALL`
	`FAN EVENTS 4001 ENB 4002 ENB 4003 ENB 4004 ENB`
	
	`RQ EVENT FAN 4001`
	`FAN EVENT 4001 ENB`

Store Fan Settings	
Command Syntax	`STORE FAN ALL`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores the fan settings.
Example	`STORE FAN ALL`

Copyright © 2023, Brooks Automation, Inc.

6. Events System

Events Handling

The Events System is an optional subsystem that provides a way to get the status of certain robot conditions without the user having to specifically request the status. Some examples of events include: MOTION_PAUSED, IO_ACTIVE, FAN STALLED, and ROBOT_PICK_AWC_OFFSET.

The Events System can be turned on or off globally so that all events can be turned on or off by the user. The default choice is off. To employ events, a user must first enable them globally.

Each specific event can also be turned on or off individually.

When an event is triggered, a message specifically related to that event will occur from the robot to the console. The event message will be sent based on the event itself, and not related to any command or response message. An example of the format of the event message is:
`_EVENT IO TEST_IN 03001`

Request/Set/Store Event System

The commands in this section affect all LEAP events. These commands supersede any of the individual events commands later in this chapter.

Request Event System	
Command Syntax	`RQ EVENT SYSTEM`
Response Syntax	`EVENT SYSTEM: ON|OFF`
Arguments	None
Description	Provides the user with the current global state of the Events System.
Example	`RQ EVENT SYSTEM` `EVENT SYSTEM: OFF`

Copyright © 2023, Brooks Automation, Inc.

Set Event System	
Command Syntax	`SET EVENT SYSTEM ON|OFF`
Response Syntax	There is no response to this command.
Arguments	`ON|OFF`
Description	Enables or disables the current global state of the Events System.
Example	`SET EVENT SYSTEM`

Store Event System	
Command Syntax	`STORE EVENT SYSTEM ON|OFF`
Response Syntax	There is no response to this command.
Arguments	`ON|OFF`
Description	Persistently stores the state of the current global Events System.
Example	`STORE EVENT SYSTEM`

Individual Events Commands

Individual event commands can be broken up into subcategories. For LEAP robots, events can be divided into the following categories:

- Motion
- Robot
- IO events
- Fan

Individual events are subordinate to the General Events commands of the previous section. This section lists all of the individual events supported by LEAP and provides a few examples. The details of each specific event are covered in each event's designated chapter, i.e. motion or I/O.

Individual events currently supported by LEAP include:

Event Type/Group	Event ID(s)	Comments
MOTION_PAUSED	2600	
MOTION_RESUMED	2601	
ROBOT_POSITION	2604	

Copyright © 2023, Brooks Automation, Inc.

Event Type/Group	Event ID(s)	Comments
ROBOT_RTR	2605	
ROBOT_PICK_AWC_OFFSET	2610	
ROBOT_PLACE_AWC_OFFSET	2611	
IO_INACTIVE	3000	
IO_ACTIVE	3001	
IO_BLOCKED	3002	
IO_NOT_BLOCKED	3003	
IO_ENABLED	3009	
IO_DISABLED	3010	
IO_IN	3015	
IO_OUT	3016	
IO_ON	3017	
IO_OFF	3018	
IO_RETRACTED	3019	
IO_NOT_RETRACTED	3020	
IO_NEAR	3025	
IO_NOT_NEAR	3026	

Fan Events

Event Type/Group	Event ID(s)	Comments
FAN ON	4001	
FAN STALLED	4002	
FAN STOPPED	4003	
FAN UNKNOWN ERROR	4004	

Copyright © 2023, Brooks Automation, Inc.

Request/Set/Store Robot/Motion Event

This section covers individual events generated from either the robot or motion.

NOTE: These commands are subordinate to the global event commands previously shown.

Request Event	
Command Syntax	`RQ EVENT (ROBOT|MOTION|FAN) (EVENT_CODE|ALL)`
Response Syntax	`ROBOT EVENTS 2602 DIS 2603 DIS 2604 DIS 2605 ENB 2610 ENB 2611 ENB 2612 DIS 2613 DIS 2614 DIS 2616 DIS 2617 DIS`
Arguments	`(ROBOT|MOTION|FAN) (EVENT_CODE|ALL)`
Description	Requests event settings.

Example	`RQ EVENT ROBOT ALL`
	
	`ROBOT EVENTS 2602 DIS 2603 DIS 2604 DIS 2605 ENB 2610 ENB 2611 ENB 2612 DIS 2613 DIS 2614 DIS 2616 DIS 2617 DIS`
	
	`RQ EVENT ROBOT 2602`
	`ROBOT EVENT 2602 DIS`
	
	`RQ EVENT MOTION ALL`
	`MOTION EVENTS 2600 DIS 2601 DIS`
	
	`RQ EVENT FAN ALL`
	`FAN EVENTS 4001 DIS 4002 DIS 4003 DIS 4004 DIS`

Set Event	
Command Syntax	`SET EVENT`
Response Syntax	There is no response to this command.
Arguments	`(ROBOT|MOTION) EVENT_CODE (ENB|DIS)`
Description	Enables or disables the event.
Example	`SET EVENT ROBOT 2602 ENB`

Copyright © 2023, Brooks Automation, Inc.

Store Event	
Command Syntax	`STORE EVENT`
Response Syntax	There is no response to this command.
Arguments	`(ROBOT|MOTION) (EVENT_CODE|ALL)`
Description	Stores the event setting.
Example	`STORE EVENT ROBOT 2602`

Request/Set/Store IO Event

This section covers individual events generated from various IO conditions.

NOTE: These are similar to the previous section but contain the additional text “IO." These commands are subordinate to the global event commands previously shown.

Request IO Event	
Command Syntax	`RQ EVENT IO`
Response Syntax	There is no response to this command.
Arguments	`(ROBOT|MOTION) (EVENT_CODE|ALL)`
Description	Requests event settings for the mapped IO.
Example	`RQ EVENT IO IO_NAME EVENT_CODE`

Set IO Event	
Command Syntax	`SET EVENT IO`
Response Syntax	There is no response to this command.
Arguments	`(IO_NAME) (EVENT_CODE) (ENB|DIS)`
Description	Enables or disables event for the mapped IO.

Example	`MAP TEST_IN DISCRETE_IN HI TO DIGITAL_IN 0x1`
	`SET EVENT SYSTEM ON`
	`SET EVENT IO TEST_IN 3001 ENB`
	`ECHO change input signal on 0x1 to active to cause event`
	`_EVENT IO TEST_IN 03001`

Copyright © 2023, Brooks Automation, Inc.

Store IO Event	
Command Syntax	`STORE EVENT IO`
Response Syntax	There is no response to this command.
Arguments	`(ROBOT|MOTION) (EVENT_CODE|ALL)`
Description	Stores the event settings for the mapped IO.
Example	Stores the event settings for the mapped IO: `STORE EVENT IO TEST_IN 3001 ENB`

Copyright © 2023, Brooks Automation, Inc.

7. Homing And Basic Motion

Coordinate System

The Polar Coordinate System is used with all robots. In this system, the center of the robot drive is the center of the system.

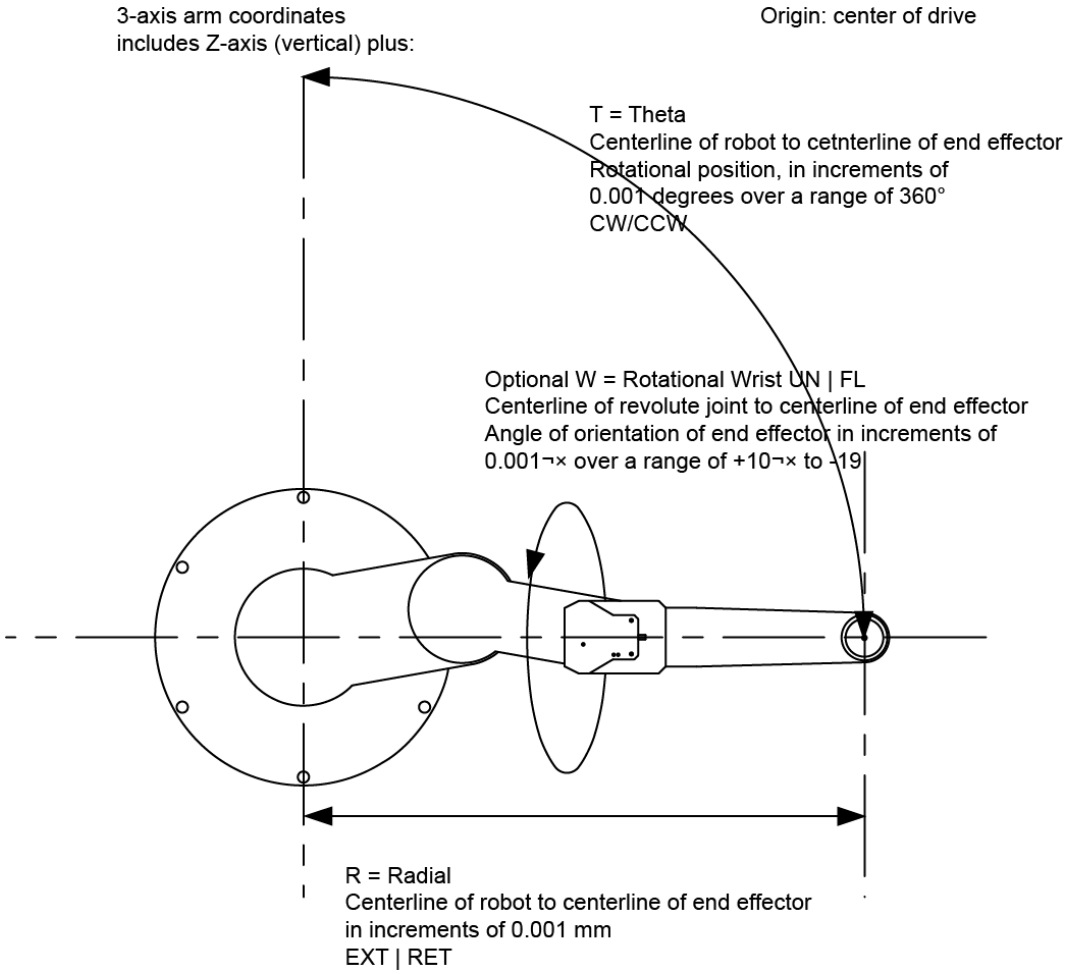


Figure 7-1: Polar Coordinates

Copyright © 2023, Brooks Automation, Inc.

NOTE: R and T determines the destination location of the center of the end effector. W defines the destination orientation of the end effector.

Reference and Request Reference

References the specified axis and reports the current status of the axis. The robot reference function references the robot at the current position and then holds the arm at that position. If a reference is requested that is not present, and error is reported.

NOTE: The Z-axis cannot be referenced.

Reference		
Command Syntax	`REF [R|T]`	
Response Syntax	There is no response to this command.	
Arguments	R:	References the radial axis.
	T:	References the theta axis.
Description	References the specified axis and reports the current status of the axis.	
Example	`REF R`	

Request Reference		
Command Syntax	`RQ REF [[ALL] | [(R|T)]`	
Response Syntax	`REF (Y|N)(Y|N)` ***NOTE:*** *Response will be in the following order: R and then T.*	
Arguments	ALL:	Request reference for all axes.
	R:	References the radial axis.
	T:	References the theta axis.
Description	References the specified axis and reports the current status of the axis.	
Example	`RQ REF R` `REF R Y`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request Servos

Sets the servos and requests the servo activation state. Turning the servos off releases the servos and allow hand locating the arm for teaching. To engage the servos, enter HOME R or HOME ALL. The Request command will report the activation state of the servo. If an axis is not present, an error is reported for that axis.

> ***NOTE:*** This command is NOT used in the normal operation of the robot. Call Brooks Automation Technical Support for instructions on the correct use of this command. For more information on contacting Brooks Technical Support, please see "Contact Brooks Automation Technical Support".

Set Servos		
Command Syntax	`SET SERVOS (ON | OFF)`	
Response Syntax	There is no response to this command.	
Arguments	ON:	Turns Servos on.
	OFF:	Turns Servos off.
Description	Sets the servos and requests the servo activation state.	
Example	`SET SERVOS OFF`	

Request Servos		
Command Syntax	`RQ SERVO [ALL | (R|T|Z)]`	
Response Syntax	`RQ SERVO ALL` `SERVO (Y|N)(Y|N)(Y|N)`	
Arguments	ALL:	Requests value of all axes.
	RL:	Requests value of R-axis.
	T:	Requests value of T-axis.
	Z:	Requests value of Z-axis.
Description	Requests the servos and requests the servo activation state.	
Example	`RQ SERVO ALL` `SERVO Y Y Y`	

Copyright © 2023, Brooks Automation, Inc.

Release

Releases servo control of the robot. The release command performs a controlled stop of any motion in progress, turns the servos off, and remains referenced.

A background ready response will be returned.

Release	
Command Syntax	`RELEASE`
Response Syntax	There is no response to this command.
Arguments	None
Description	Releases servo control of the robot.
Example	`RELEASE`

Halt

NOTICE

While no damage to the equipment occurs due to a HALT command, the HALT command may result in abrupt motions, and may cause the misalignment of a wafer that is on the end effector, load port, or chuck.

The halt command immediately halts all motion operations. This command is available only when the robot is in Background Mode, Multi Mode, or Multi-Device Mode.

A controlled stop to the robot is applied to halt all motion while minimizing wafer movement on the end effector and the Z-axis brake applied.

In MAG6 Compatibility: Servos are maintained after a HALT; all axes that were servoed before the HALT are still servoed after the HALT. The Z-brake disengages after receiving the next motion command.

In VT5 Compatibility: After the robot comes to a controlled stop, the servos for the arm disengage.

To command a Halt in Sequential Mode, enter `<Ctrl-C>`.

The HALT command can also be entered from the keyboard as `<Ctrl-C>`.

Copyright © 2023, Brooks Automation, Inc.

Halt	
Command Syntax	`HALT`
Response Syntax	There is no response to this command.
Arguments	None
Description	Immediately halts all motion operations.

Example	The following example stops all current movement of the robot arms and the traverser.
	
	Multi Mode:
	`02 HALT`
	`_ACK 02`
	`_DONE 02`
	`_ERR 01 603 (command halted)`
	(must add identifier)
	
	NOTE: *`Ctrl-C` is a non-printable character. `Ctrl-C` must be physically pressed on the keyboard.*
	
	`<Ctrl-C>`
	`_ACK 00`
	`_DONE 020`
	`_ERR 01 603 (command halted)`
	(automatically adds identifier)
	
	NOTE: *`Ctrl-C` is a non-printable character. `Ctrl-C` must be physically pressed on the keyboard.*
	
	SEQ Mode:
	`<Ctrl-C>`
	`_ERR 01 603 (command halted)`

Copyright © 2023, Brooks Automation, Inc.

Home

NOTICE

The inter-axis interlocks used during HOME ALL are not active during individual axis home operations. The robot responds to a home Z or home T command even if the arm is extended. The user should verify that the arm is retracted before attempting to home on T or Z.

The purpose of the Home command is to move the robot into a known safe-position.

Sequence for multi-axis Homes:

a. R-axis (homes toward retract position)
b. T-axis (homes clockwise or counterclockwise, whichever is the shortest distance)
c. Z-axis (homes downward)

If a home command is entered and the robot is already at the home position, no motion occurs. The robot is not allowed to home Z or home T if it is not referenced and the R-axis is not completely retracted.

Home		
Command Syntax	`HOME ( ALL | R | T | Z )`	
Response Syntax	There is no response to this command.	
Arguments	ALL:	Moves all axes to their home position.
	R:	Specifies homing in the R-axis.
	T:	Specifies homing in the T-axis.
	Z:	Specifies homing in the Z-axis.
Description	Moves the robot into a known safe-position.	

Example	The following example homes the arm's R-axis by moving the arm to the "home" reference position.
	`HOME R`
	In the following example arm 'A' is currently extended and in the down position. The robot will retract the arm and home the R-axis. If the Z-axis option is present the robot will lower the arms and home Z. The robot will then rotate the arms counterclockwise and home the T-axis.
	`HOME ALL`	

Copyright © 2023, Brooks Automation, Inc.

Request Position Absolute

Requests, for the specified axis, the actual position of the pan in absolute coordinates for the specified arm. The numbers given represent the actual physical location of the end effector in absolute coordinates. This position may differ slightly from the position to which the robot was commanded.

If multiple axes are requested, they are presented in the following order: R, T, Z. In MAG6 Compatibility, no matter what order the axes are requested, they are returned in the order shown above; In VT5 Compatibility, the values are returned in the order they were requested.

Request Position Absolute		
Command Syntax	`RQ POS ABS [[ARM] (A|B)] (ALL | { R | T | Z })`	
Response Syntax	`POS ABS (A|B) rVal tVal zVal`	
Arguments	ALL:	Specifies R, T, and Z in the order presented in the command format.
	ARM:	Arm descriptor
	R:	`rVal`: The current R-axis location of the robot arm in microns (m) (.001 mm).
	T:	`tVal`: The current T-axis location of the robot arm in 0.001 degrees.
	Z:	`zVal`: The current Z-axis location of the arm in microns (m) (.001 mm).
Description	Requests, for the specified axis, the actual position of the pan in absolute coordinates for the specified arm.	

Example	In the following example, the robot returns the current position of Arm A in absolute coordinates:
	`RQ POS ABS ARM A R`
	`POS ABS A R 175000`	

Copyright © 2023, Brooks Automation, Inc.

Request Position Destination

Requests the destination position for the current motion segment. The numbers given represent the physical destination location of the pan for the current particular motion. The destination location will depend on where the arm is in the move cycle. The actual position the robot moves to may differ slightly from the position to which the robot was commanded.

If the request is made after an abort or the robot is standing still, the last move destination will be displayed.

Request Position Destination		
Command Syntax	`RQ POS DST [[ARM] (A|B)] (ALL|{R|T|Z})`	
Response Syntax	`RQ POS DST ALL` `POS DST (A|B) rVal tVal zVal`	
Arguments	ALL:	Specifies axis in the order presented in the response format.
	ARM:	Arm descriptor
	R:	`rVal`: The current R-axis destination of the robot arm in microns (m) (.001 mm).
	T:	`tVal`: The current T-axis destination of the robot arm in 0.001 degrees.
	Z:	`zVal`: The current Z-axis destination of the arm in microns (m) (.001 mm).
Description	Requests the destination position for the current motion segment.	

Copyright © 2023, Brooks Automation, Inc.

Request Position Destination	
Example	Station #1 and station #2 are defined as different theta locations. Arm is currently at station #1 with R retracted. Communication Flow is not Sequential Mode. The following command is entered: `GOTO N 2 Z DN R EX`. The robot will experience three stages of movement. Stage 1: Retract the arm to station #1 retracted location Stage 2: Perform a theta rotation to station #2 Stage 3: Extend at station #2 If the RQ POS DST ALL command is made during stage 1, the destination location will return: R = station #1 retracted position T = station #1 theta Z = current position If the RQ POS DST ALL command is made during stage 2, the destination location will return: R = station #2 retracted position T = station #2 the= station #2 lower position Z = station #2 lower position If the RQ POS DST ALL command is made during stage 3, the destination location will return: R = station #2 extended position T = station #2 theta Z = current position

Copyright © 2023, Brooks Automation, Inc.

Request Position Target

Requests, for the specified axis, the position in absolute coordinates to which the robot arm has been commanded. The numbers given represent the physical final destination of the end effector in absolute coordinates to which the robot was commanded. The position to which the robot actually moves may vary slightly from the position to which the robot was commanded.

Request Position Target		
Command Syntax	`RQ POS TRG [[ARM](A|B)] [ALL | {R|T|Z}]` *ARM is not supported in VT5 Compatibility.*	
Response Syntax	`POS TRG (A|B) rVal tVal zVal`	
Arguments	ALL:	Specifies all axes in the order presented in the response format.
	ARM:	Arm descriptor
	R:	`rVal`: Radial value. The target R-axis location of the robot arm in microns (m) or "------" if unreferenced.
	T:	`tVal`: Theta value. The target T-axis location of the robot arm in 0.001 degrees or "---- --" if unreferenced.
	Z:	`zVal`: Z-axis value. The target Z-axis location of the arm in microns (m) or "------" if unreferenced.
Description	Requests, for the specified axis, the position in absolute coordinates to which the robot arm has been commanded.	
Example	`POS TRG 0306220 0000000 0000000`	

Copyright © 2023, Brooks Automation, Inc.

Request Minimum/Maximum Axis Values

Requests the minimum and maximum values for each axis. Reports the values set for the all available axes of the robot.

The software applies the following limit checks:

- Theta station position between the minimum and maximum allowed (0° and 360°)
- Z position between the minimum and maximum allowed, based on the arm geometry
- R position between the minimum and maximum allowed, based on the arm geometry

Request Maximum Axis Values		
Command Syntax	`RQ MAX [[ARM] (A|B)] (ALL | R | T | Z)`	
Response Syntax	`POS MAX (A|B) rLimit tLimit zLimit`	
Arguments	MAX:	Maximum value axis limits
	ARM:	Arm descriptor
	ALL:	Requests the maximum value of all axes.
	R:	Requests the maximum value of the R-axis.
	T:	Requests the maximum value of the T-axis.
	Z:	Requests the maximum value of the Z-axis.
Description	Requests the maximum values for each axis.	

Example	`RQ MAX A R`
	`POS MAX A 0604192`
	
	`RQ MAX ALL`
	`POS MAX 0610472 0359999 0150000`	

Request Minimum Axis Values		
Command Syntax	`RQ MIN [[ARM] (A|B)] (ALL|R|T|Z)`	
Response Syntax	`POS MIN (A|B) rLimit tLimit zLimit`	
Arguments	MIN:	Minimum value axis limits
	ARM:	Arm descriptor
	ALL:	Requests the minimum value of all axes.
	R:	Requests the minimum value of the R-axis.
	T:	Requests the minimum value of the T-axis.
	Z:	Requests the minimum value of the Z-axis.

Copyright © 2023, Brooks Automation, Inc.

Request Minimum Axis Values	
Description	Requests the minimum values for each axis.
Example	`RQ MIN ALL` `POS MIN -130471 0000000 0000000`

Set/Request/Store Load Mode

This command sets the mode for reporting the load status of the arm. The load status is used to determine the speed of all motion commands. The two state mode will report load status as ON or OFF. The command WAF_SEN is not required after homing. The default at power up is ON. The three state mode will report load status as ON, OFF or ? (unknown). The CHECK LOAD command or the SET LOAD command must be executed after homing.

Set Load Mode		
Command Syntax	`SET LOAD MODE (BI|TRI)(ON|OFF|?)`	
Response Syntax	There is no response to this command.	
Arguments	BI:	Set two state mode (ON|OFF).
	TRI:	Set three state mode (? |ON|OFF).
	ON:	Set mode on.
	OFF:	Set mode off.
	?:	Set mode to unknown.
Description	Sets the load mode.	
Example	`SET MODE TRI ON`	

Request Mode Load		
Command Syntax	`RQ LOAD MODE`	
Response Syntax	`LOAD MODE (BI|TRI)(ON|OFF|?)`	
Arguments	BI:	Two state mode
	TRI:	Three state mode
	ON:	Mode is on.
	OFF:	Mode is off.
	?:	Mode is unknown.

Copyright © 2023, Brooks Automation, Inc.

Request Mode Load	
Description	Requests the current load mode.
Example	`RQ LOAD MODE` `LOAD MODE TRI ON`

Store Load Mode	
Command Syntax	`STORE LOAD MODE`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores the current load mode.
Example	`STORE LOAD MODE`

Set/Request Load

Sets and requests the load status for the specified arm and pan. This command sets the load status for the arm. The load status is used to determine the speed of all motion commands.

Setting the load to “(?)” will cause the robot to move at slow speed until determination of true status of the end effector is known. The unknown status is only available if the load mode has been set to TRI. When the active status is unknown, the robot will move at “with wafer” slow speed. When load mode is set to TRI, at power-up the robot arm loads are set to unknown. The robot will continue to assume the load is unknown until either a pick, place, check load, or set load on|off command is executed.

Power-up: The robot’s arm(s) is assumed to be unknown. The robot will continue to assume that the arm(s) is/are unknown until either a place or a set load off command is executed. Once the arm(s) is/are defined as empty, the robot will continue to assume it is empty until a pick or a set load on command is executed.

The Request command is used to determine the current load status of the robot’s arm(s). Note that at power-up, the robot’s arm(s) is assumed to be loaded. The robot will continue to assume the arm (s) is loaded until either a place or a set load off command is executed.

NOTE: This command cannot be stored.

Copyright © 2023, Brooks Automation, Inc.

Set Load		
Command Syntax	`SET LOAD [[ARM](A|B)] [PAN (L|R)] [?|ON|OFF]`	

Response Syntax	BI state mode/MAG6 Compatibility:
	`LOAD A|B PAN L|R (ON|OFF)`
	
	TRI state mode/MAG6 Compatibility:
	`LOAD A|B PAN L|R (?|ON|OFF)`
	
	This command is not supported in VT5 Compatibility Mode.	

Arguments	ARM: LOAD: PAN:	Arm descriptor `ON`: Arm has a load on the specified end effector (after successful check for material). `OFF`: Arm does not have a load on the specified end effector; all moves will be at high speed (after a PLACE or if state set to OFF). `?`: (UNKNOWN) Unsure of load status (at power-up, or other times when status is unknown). `L`: Left Pan `R`: Right Pan
Description	Sets the load status for the specified arm and pan	

Example	The following command will set Load ON for Arm A. The robot software will assume that arm 'A' is carrying a wafer during all subsequent motion command until either the load is turned OFF or a PLACE is performed.
	
	`SET LOAD ARM A ON`
	
	The following command will set Load OFF for Arm A. The robot software will assume that arm 'A' is not carrying a wafer during all subsequent motion command until either the load is turned ON or a PICK is performed.
	
	`SET LOAD ARM A OFF`	

Request Load	
Command Syntax	`RQ LOAD [[ARM](A|B)] [PAN (L|R)]`
Response Syntax	`LOAD (A|B) [PAN (L|R)] (?|ON|OFF)`

Copyright © 2023, Brooks Automation, Inc.

Request Load		
Arguments	ARM:	Arm descriptor
	LOAD:	`ON`: Arm has a load on the specified end effector (after successful check for material). `OFF`: Arm does not have a load on the specified end effector; all moves will be at high speed (after a PLACE or if state set to OFF). `?`: (UNKNOWN) Unsure of load status (at power-up, or other times when status is unknown).
	PAN:	`L`: Left Pan `R`: Right Pan
Description	Requests the load status for the specified arm and pan	

Example	The following example requests the Load status for Arm A. The robot responds that it currently assumes that a wafer is on arm 'A.'
	
	`RQ LOAD A`
	`LOAD A ON`
	
	`RQ LOAD ARM A`
	`LOAD A ON`
	
	The following example requests the load status for Arm A, Left Pan. The robot responds that it currently assumes that a wafer is on Arm A, Left Pan.
	
	`RQ LOAD ARM A PAN L`
	`LOAD A PAN L ON`	

Move

CAUTION

Move is not Interlocked

Move is not interlocked. Simultaneous multi-axis moves are allowed, which could result in physical contact between the robot's arms and external structures.

The Move command does not use any station information and it does not prevent the robot from extending into any external structure.



Moves one or more axes to a specified location in "physical" coordinates. If more than one axis is specified, the move is one axis at a time.

Move is a "primitive" motion command: regardless of the axes being moved, no interlocking is in force. This means that the arm moves both rotational and vertical axes while in the extended position. Take great care when using move to avoid impact against obstructions.

Copyright © 2023, Brooks Automation, Inc.

The REL and ABS move-types require that the axis be homed. This is the normal mode of operation and is recommended for safe, reliable motion.

The set load command may be used to define the load status of the arm before executing the move command. All motions follow the speed and acceleration profile appropriate for the currently defined load.

> ***NOTE:*** Due to the nature of the move command, only single axis moves should be performed.

Move		
Command Syntax	`MOVE {axis (ABS|REL) value} [[ARM] (A|B)]`	
Response Syntax	There is no response to this command.	
Arguments	ARM:	Arm descriptor
	R|T|Z:	Specifies the axis to be moved.
	ABS|REL:	`ABS`: use absolute location. `REL`: use relative distance from current position.
	value:	Specifies the distance or location to which to move in integer value. For the `REL` move-type, the amount is a relative distance. For `ABS` it is an absolute position. The range for R motion depends on the geometry of the arms.
	UP|DN:	Move the Z axis UP or Down when unreferenced by spinning the Z motor a specified amount of time.
	val:	Specifies the theta position to move to in millidegrees.
	CW|CCW:	`CW`: Moves the arm clockwise. `CCW`: Moves the arm counter-clockwise.
Compatibility Notes	In MAG6 Compatibility, entering the word ARM is optional; in VT5 Compatibility, ARM is required and Wafer Speed is used regardless of the load status.	
Description	Moves one or more axes to a specified location in “physical” coordinates.	

Copyright © 2023, Brooks Automation, Inc.

Move	

Example	In the following example, the robot extends arm A by 100 microns:
	`MOVE R REL 100`
	In the following example, the robot moves the ‘A’ arm to the R axis coordinate 13.500 mm (.531 in):
	`MOVE R ABS 13500`
	In the following example, the robot moves to 180° in the absolute position, rotating in the counter clockwise direction:
	`MOVE T ABS 180000 CCW`

Copyright © 2023, Brooks Automation, Inc.

Set Speed

Sets the speed to low, medium, or high.

The set low speed command allows the operator to ensure that the arm will move at low speed throughout an entire command such as pick or place. To ensure the safest operation, slower forced speed commands will overwrite faster forced speed options, but faster forced speed commands will not overwrite slower forced speed commands. This option is only active for the next single command. Low speed is always set to normal at power-up.

The set medium speed command is only available on some robots. Selecting medium speed on a robot that does not support it, will have no effect.

The set high speed command forces the next move motion to happen at high speed. This speed will apply to one move only. When the wafer is held in position on the end effector only by friction, high speed motion is likely to cause misalignment of the wafer relative to the end effector. To prevent this, the only allowed high speed motions are as follows:
The “without wafer” velocity and acceleration will be used only during the initial motions in a pick and the final motions in a place, and they are selected by the robot based on the pick/place history of both arms.

The optimal speed of the wafer transfer is calculated by the presence of a wafer on the end effector. The preferred method of controlling the robot’s speed is to use the set load command.

NOTE: This command cannot be stored. It toggles back to N after the subsequent action command. HISPD is always set to N at power-up.

Set Speed		
Command Syntax	`SET (LOSPD|HISPD|MESPD) (Y|N)`	
Response Syntax	There is no response to this command.	
Arguments	LOSPD:	Sets the speed to low.
	HISPD:	Sets the speed to high.
	MESPD:	Sets the speed to medium.
Description	Sets the speed to WAF/PAN speed for next move sequence.	
Example	`SET LOSPD Y`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request CDM Speeds

Sets and requests the teaching speeds of the robot. The Set CDM Speeds or the Set Teach command will allow the robot to operate at CDM or jog speeds through the Serial interface. Setting the speed to On will run all commands at the slow speeds until the Off command is set or the robot is Reset.

Set Speeds		
Command Syntax	`SET TEACH (ON|OFF)` `SET CDM SPEEDS (ON|OFF)`	
Response Syntax	`CDM SPEEDS (ON|OFF)`	
Arguments	ON: OFF:	Forces motion to slow speed. Normal speeds (default)
Description	Sets the teaching speeds of the robot.	
Example	`SET CDM SPEEDS OFF`	

Request Speeds	
Command Syntax	`RQ CDM SPEEDS`
Response Syntax	`CDM SPEED (ON|OFF)`
Arguments	None
Description	Requests the teaching speeds of the robot.
Example	`RQ CDM SPEEDS` `CDM SPEED OFF`

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Retract 2 Value

Sets, requests, and stores the second retract value (R2) for the pick and place with an offset command.

Set Retract 2 Value		
Command Syntax	`SET RTRCT2 (retractSec|CLEAR)`	
Response Syntax	There is no response to this command.	
Arguments	retractSec: CLEAR:	Sets the second retract location value in microns. Clears R2 value.
Description	Sets the value of the second retract location when using the pick with an offset and place with an offset command. The default Retract 2 value is the factory set value for the retract position.	
Example	`SET RTCRT2 625000`	

Request Retract 2 Value	
Command Syntax	`RQ RTRCT2`
Response Syntax	`RTRCT2 retract_value`
Arguments	None
Description	Requests the value of the second retract location when using the pick with an offset and place with an offset command.
Example	`RQ RTRCT2` `RTRCT2 625000`

Store Retract 2 Value	
Command Syntax	`STORE RTRCT2`
Response Syntax	There is no response to this command.
Arguments	None

Copyright © 2023, Brooks Automation, Inc.

Store Retract 2 Value	
Description	Stores the value of the second retract location when using the pick with an offset and place with an offset command.
Example	`STORE RTRCT2`

Copyright © 2023, Brooks Automation, Inc.

Request Arm Retract Value

The retract circle is when R is retracted (R home) while T varies from 0 to 359°.

Behavior of robot when VIA is defined within retract circle:

- WAVE: VIA is ignored and the station is treated as a radial station.
- LEAP: VIA is defined within the retract circle.
  - When attempting to access the station, an error is reported.
  - When homing, the station is ignored.

Request Arm Retract Value	
Command Syntax	`RQ ARM RTRCT`
Response Syntax	`ARM 0089800`
Arguments	None
Description	Requests the retract circle for the R-axis.
Example	`RQ ARM RTRCT` `ARM 0089800`

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Interlock

Sets, requests, and stores interlocks to avoid wafer collisions when using an aligner.

`ALIGNER WFR_PRS`: This Wafer Present Interlock is used to check if a wafer is on the aligner before attempting to place a wafer at the aligner station.

`ALIGNER CHUCK`: If the Chuck Interlock is set to N, the robot will place a wafer to the aligner regardless of the actual situation.

`ALGN 300_WFR_PRS`: Prevents wafer collision by inhibiting robot motion when the aligner has a 300mm wafer. If this interlock is set to Y, the robot will check for 300mm wafer presence on the aligner. Only action commands will be refused. This interlock does not affect the HOME command.

`HOME WFR_PRS`: Sets the mode for checking wafer presence on end effector and aligner. If a wafer is present on any aligner, all action commands will not start and the robot will report an error, with the following exception. The pick command is allowed if it is issued to the aligner and the robot theta position is pointing +20° to the aligner station theta position. If this interlock is set to Y, the robot will check for wafer presence on the end effector and on the aligner before homing in the Theta axis. If a wafer is present on either the robot or the aligner, the robot will report error number 226 “Wafer is present, must do home Z command now”. This forces a home Z action before home T. This interlock assumes the Z home position (Z=0) is a safe position to rotate the robot with a wafer on the end effector OR with a wafer on the aligner.

Set Interlock		

Command Syntax	`SET INTLCK [ALIGNER WFR_PRS (Y|N)]`
	`[ALIGNER CHUCK (Y|N)]`
	`[ALGN 300_WFR_PRS (Y|N)]`
	`[INTLCK HOME WFR_PRS (Y|N)]`	

Response Syntax	There is no response to this command.	
Arguments	ALIGNER WFR_ PRS:	Y: Robot will check for wafer presence on the aligner. N: Robot will not check for wafer presence.
	ALIGNER CHUCK:	Y: Checks chuck position relative to pins. N: Ignores chuck position.
	ALGN 300_WFR_ PRS:	This command is used with all aligners using 300mm wafers. 300mm Wafer Presence check interlock.
	HOME WFR_PRS:	Y: Robot will check for wafer presence on the aligner and end effector. N: Robot will not check for wafer presence.
Description	Sets interlocks to avoid wafer collisions when using an aligner.	

Copyright © 2023, Brooks Automation, Inc.

Set Interlock	

Example	`SET INTLCK ALGN 300_WFR_PRS Y`
	
	`SET INTLCK HOME WFR_PRS Y`

Request Interlock		
Command Syntax	`RQ INTLCK [ALIGNER WFR_PRS] [CHUCK] [ALGN 300_WFR_PRS] [INTLCK HOME WFR_PRS]`	

Response Syntax	`INTLCK`
	`ALGN_WFR_PRS (Y|N)`
	`ALGN_CHUCK (Y|N)`
	`ALGN 300_WFR_PRS (Y|N)`
	`HOME WFR_PRS (Y|N)`	

Arguments	ALIGNER WFR_ PRS:	Y: Robot checks for wafer presence on the aligner. N: Robot does not check for wafer presence.
	ALIGNER CHUCK:	Y: Checks chuck position relative to pins. N: Ignores chuck position.
	ALGN 300_WFR_ PRS:	This command is used with all aligners using 300mm wafers. 300mm Wafer Presence check interlock.
	HOME WFR_PRS:	Y: Robot checks for wafer presence on the aligner and end effector. N: Robot does not check for wafer presence.
Description	Requests, and stores interlocks to avoid wafer collisions when using an aligner.	
Example	`RQ INTLCK ALGN 300_WFR_PRS` `300_WFR_PRS Y`	

Store Interlock		
Command Syntax	`STORE INTLCK [ALIGNER WFR_PRS] [CHUCK] [ALGN 300_WFR_PRS] [INTLCK HOME WFR_PRS]`	
Response Syntax	There is no response to this command.	
Arguments	ALIGNER WFR_PRS:	Stores the specific interlock setting.
	ALIGNER CHUCK:	Stores the specific interlock setting.
	ALGN 300_WFR_PRS:	Stores the specific interlock setting.
	HOME WFR_PRS:	Stores the specific interlock setting.
Description	Stores interlocks to avoid wafer collisions when using an aligner.	

Copyright © 2023, Brooks Automation, Inc.

Store Interlock	
Example	`STORE INTLCK ALIGNER WFR_PRS`

Copyright © 2023, Brooks Automation, Inc.

Find/Set/Request/Store Mount

The mount command causes the robot to move to the “mount new arm” position, which is defined as an R-axis shaft angle of 90°, and a Z position as established by the SET MOUNT Z command. The mount executed sequence is: Home R-axis, Home Z-axis, Home T-axis.

The set command assigns the position to which the arm moves in response to the mount command. During the mount motion, the robots moves in slow speed along Z to the vertical mount position as established by the SET MOUNT command. The default setting for SET MOUNT Z-axis is 10.000 mm above the Home position.

NOTE: This command is NOT used in the normal operation of the robot. Call Brooks Automation Technical Support for instructions on the correct use of this command. For more information on contacting Brooks Technical Support, please see "Contact Brooks Automation Technical Support".

Mount	
Command Syntax	`MOUNT`
Response Syntax	There is no response to this command.
Arguments	None
Description	Causes the robot to move to the “mount new arm” position, which is defined as an R-axis shaft angle of 90°, and a Z position as established by the SET MOUNT Z command.
Example	`MOUNT`

Find Mount	
Command Syntax	`FIND MOUNT`
Response Syntax	There is no response to this command.
Arguments	None
Description	Causes the robot to move to the “mount new arm” position, which is defined as an R-axis shaft angle of 90°, and a Z position as established by the SET MOUNT Z command.
Example	`FIND MOUNT`

Set Mount	
Command Syntax	`SET MOUNT [(ALL|R|T|Z|S) value]`

Copyright © 2023, Brooks Automation, Inc.

Set Mount		
Response Syntax	There is no response to this command.	
Arguments	value:	The distance to which the arm moves, relative to the Home position, when it receives the MOUNT command prior to mounting or dis-mounting the arm.
	r-value:	Radial value
	br-value:	Arm B radial value (if present)
	t-value:	Theta value
	bt-value:	Arm B theta value (if present)
	z-value:	Z-axis value
Description	Assigns the position to which the arm moves in response to the mount command.	
Example	`SET MOUNT B R 249527`	

Request Mount		
Command Syntax	`RQ MOUNT`	
Response Syntax	`MOUNT r-value br-value t-value bt-value z-value s-value wa-value wb-value w-value`	
Arguments	r-value:	Radial value
	br-value:	Arm B radial value (if present)
	t-value:	Theta value
	bt-value:	Arm B theta value (if present)
	z-value:	Z-axis value
Description	Requests the position to which the arm moves in response to the mount command.	
Example	`RQ MOUNT` `Mount Position 00200000 00358273 00010001`	

Store Mount	
Command Syntax	`STORE MOUNT`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Mount	
Arguments	None
Description	Stores the position to which the arm moves in response to the mount command.
Example	`STORE MOUNT`

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Event Motion

Set Event Motion		
Command Syntax	`SET EVENT MOTION event_code [ENB|DIS]`	
Response Syntax	There is no response to this command.	
Arguments	ENB|DIS:	`ENB`: Enables motion events. `DIS`: Disables motion events.
	event_code:	`2600`: Pauses robot motion. `2601`: Resumes robot motion.
Description	Sets the events to control motion, robot, and fan.	
Example	`SET EVENT MOTION 2600 ENB` `SET EVENT MOTION 2601 ENB`	

Request Event Motion		
Command Syntax	`RQ EVENT MOTION (event_code|ALL)`	
Response Syntax	`MOTION EVENTS event_code (ENB|DIS)`	
Arguments	ENB|DIS:	`ENB`: Enables Motion Events. `DIS`: Disables Motion Events.
	event_code:	`2600`: Pauses robot motion. `2601`: Resumes robot motion.
Description	Requests the events to control motion, robot, and fan.	
Example	`RQ EVENT MOTION ALL` `MOTION EVENTS 2600 ENB 2601 DIS`	

Store Event Motion	
Command Syntax	`STORE EVENT MOTION (ROBOT|MOTION|FAN) (EVENT_CODE|ALL)`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores the events to control motion, robot, and fan.
Example	`STORE EVENT MOTION`

Copyright © 2023, Brooks Automation, Inc.

Request Total Travel

Requesting the Total Travel reports the distance each specific axis has traveled since the birth of the robot.

Request Total Travel		
Command Syntax	`RQ TOTAL TRAVEL (ALL|R|T|Z)`	
Response Syntax	`TOTAL TRAVEL: R_axis T_axis Z_axis`	
Arguments	ALL:	Specifies travel for all axes.
	R:	Specifies travel in the R-axis.
	T:	Specifies travel in the T-axis.
	Z:	Specifies travel in the Z-axis.
Description	Requests the actual amount of motion accumulated for each axis.	

Example	`RQ TOTAL TRAVEL ALL`
	`TOTAL TRAVEL: 779547.88 20235854.00 13369607.00`
	
	`RQ TOTAL TRAVEL T`
	`TOTAL TRAVEL: 20235854.00`	

Copyright © 2023, Brooks Automation, Inc.

8. Station Setup

Station Setup Overview

Stations are assigned numbers to represent their location coordinates inside a system. Instead of telling a robot the radial, theta, and z units each time we want it to go to a station, we simply tell a robot to go to Station 1. All station numbers are dependent upon the specific system configuration. Stations are typically numbered in a clockwise direction around the robot. Assigning a station number for each module connected to the system allows a convenient shorthand for directing all robot motion and ensures that all wafer movement to and from the stations remains consistent. All arms must be separately taught each station. If only Arm A has been taught Station 1, Arm B is incapable of going to Station 1.

Each station represents a module connected to the system. A single module may have more than one station assigned to it. The stations will have the same Theta and Radial coordinates, but different Z coordinates. Any modules with more than one station are numbered on a second clockwise circuit around the robot. In Figure 8-1, if the Process Modules all had two shelves, the upper shelves would be on the second clockwise circuit. Process Module A would have Station 4 and Station 7, Process Module B would have Station 5 and Station 8, and Process Module C would have Station 6, and Station 9.

Copyright © 2023, Brooks Automation, Inc.

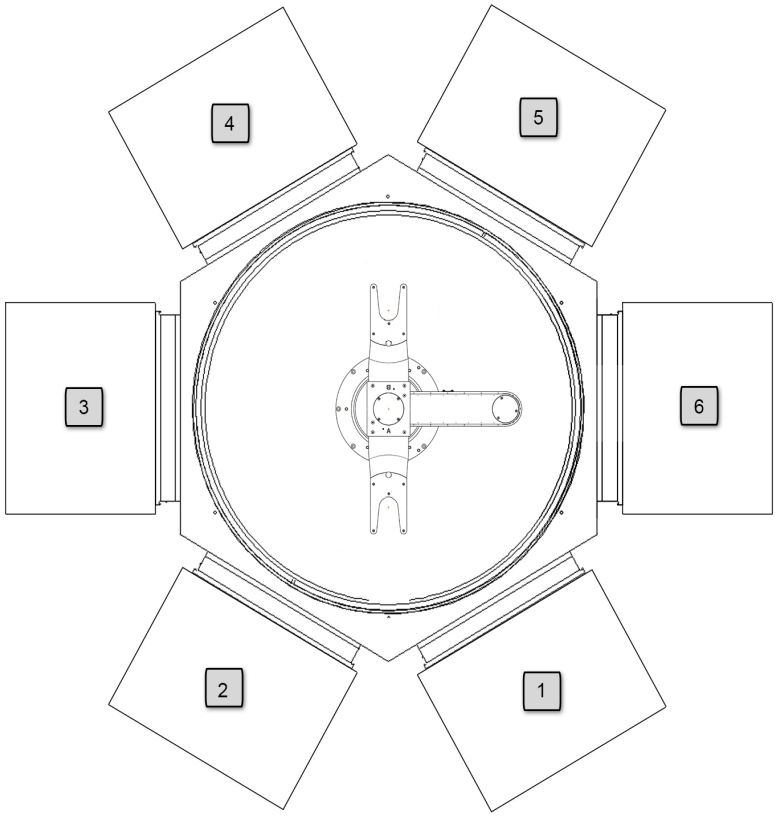


Figure 8-1: Example of Station Numbering and the Corresponding Modules

Table 8-1: Example of Station Numbering and the Corresponding Modules - Explanation

Number	Explanation
1	Station 1
2	Station 2
3	Station 3
4	Station 4
5	Station 5
6	Station 6

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store/Remove Station

Sets, requests, stores and removes the absolute coordinate values of the various station-related parameters for the specified station.

The robot software provides a station coordinate system for identifying stations and specific locations. Each station is identified by its Radial position (radial distance from Home), its Theta position (rotational position from zero), and its Z position (vertical distance from zero). By identifying the stations in this manner, it is only necessary to provide the robot with the station number instead of the complete coordinate set each time a command is issued to the robot. Multiple station numbers may be given to a single station.

All stations being used must be set for both Arm A and Arm B. For a station at a given physical location, the T parameter for Arm B may be different from the T parameter for Arm A.

The specified values for Base Transfer Offset (BTO), Pitch, and Number of Slots are checked according to the following formula: BTO + (NSLOTS - 1) PITCH < maximum allowed vertical travel.

This command requires the station number and one or more data fields. Setting one value initializes the station number. All other values will automatically default to zero.

The Remove command is used to delete a previously defined station and its values. All station values will be set to zero. Removing the station number frees that number for redefinition.

> ***NOTE:*** If station coordinates are set using the `SET STN` or `STORE STN` commands, they should be verified before performing any wafer transfers to ensure accurate station definition.

Set Station	

Command Syntax	`SET STN station <<(ARM)>[(A|B)]>`
	`[R rVal] [T tVal] [Z zVal] [LOWER lowDist][Z_WAF zWafVal] [NSLOTS numSlots] [PITCH pitch] [S sVal]`
	`[W wVal] [WA waVal] [WB wbVal] [W UN wUnFlpVal| W FL wFlpVal)]`

Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Station		
Arguments	STN:	Station number Range 1 - 99
	ARM:	Arm descriptor `A`: Arm A `B`: Arm B Default: A The arm descriptor must be specified to transfer wafers with Arm B. The “ARM” identifier is optional.
	R:	`rVal`: The station’s radial extend location in microns. Response = 7 digits
	T:	`tVal`: The station’s rotational axis location in units of millidegrees. Response = 6 digits
	Z:	`zVal`: The BTO Z-axis of the wafer Transfer Plane relative to Home in microns. Response = 6 digits
	LOWER:	`lowDist`: The distance in microns below the wafer Transfer Plane that becomes the down position location. Response = 6 digits
	NSLOTS:	`numSlots`: The number of slots at the specified station. A value of 0 or 1 indicates a non-cassette type station. Range= 0 - 100 Default = 1 Response = 6 digits ***NOTE:*** *0 and 1 both mean that the station has a single slot.*
	PITCH:	`pitch`: The pitch between slots in microns at the specified station. Default = 0 Response = 6 digits
	W UN:	*For rotating wrists only.* `wUNFlpVal`: Specifies unflip position of the end effector Rotated position axis location in units of millidegrees Response = 6 digits
	W FL:	*For rotating wrists only*. `wFlpVal`: Specifies flip position of the end effector Rotated position axis location in units of millidegrees Response = 6 digits
	S:	*This command is used with the Traverser option.* `sVal`: The station’s horizontal axis location in units of microns. Response = 6 digits
	W:	*This command is used with the 4-axis Armset.* `wVal`: Specifies the stations wrist location in millidegrees Response = 6 digits

Copyright © 2023, Brooks Automation, Inc.

Set Station		
	WA:	*This command is used with the 5-axis Armset.* `waVal`: Specifies the stations wrist location in millidegrees for the upper end effector. Response = 6 digits
	WB:	*This command is used with the 5-axis Armset.* `wbVal`: Specifies the stations wrist location in millidegrees for the lower end effector. Response = 6 digits
	Z_WAF:	*This command is used with the Z-axis option.* `zWafVal`: The value, in microns, set for the third WF position, wafer position, at the station. The value must be an absolute value between the UP position and the DN position. For example, if the BTO=50000 and LWR=1000, then a valid Z_WAF value is between 49001 and 49999. The 3_Z Station Option for Z position must be set to Yes. Response = 6 digits
Description	Sets the absolute coordinate values of the various station-related parameters for the specified station.	

Example	`SET STN 1 R 780000`
	`SET STN 1 T 180000`
	`SET STN 1 Z 38000`
	`SET STN 1 LOWER 5000`
	`SET STN 1 NSLOTS 25`
	`SET STN 1 PITCH 10000`
	
	`SET STN 1 R 0780000 T 180000 Z 38000 LOWER 5000 NSLOTS 25 PITCH 010000`	

Request Station	
Command Syntax	`RQ STN` *`station`* `[[ARM]|[(A|B)]] [ALL|{R|T|Z|LOWER|Z_WAF|PITCH|NSLOTS}]`
Response Syntax	`STN` *`station`* `[ARM (A|B)] R` *`R_Value`* `T` *`T_Value`* `Z` *`Z_Value`* `LOWER` *`Lower_Distance`* `NSLOTS` *`Number_Slots`* `PITCH` *`Pitch_Value`*

Copyright © 2023, Brooks Automation, Inc.

Request Station		
Arguments	STN:	Station number Range 1 - 99
	ARM:	Arm descriptor `A`: Arm A `B`: Arm B Default: A The arm descriptor must be specified to transfer wafers with Arm B. The “ARM” identifier is optional.
	R:	Requests the station’s radial extend location in microns. Response = 7 digits
	T:	Requests the station’s rotational axis location in units of millidegrees. Response = 6 digits
	Z:	Requests the BTO Z-axis of the wafer Transfer Plane relative to Home in microns. Response = 6 digits
	LOWER:	Requests the distance in microns below the wafer Transfer Plane that becomes the down position location. Response = 6 digits
	NSLOTS:	Requests the number of slots at the specified station. A value of 0 or 1 indicates a non-cassette type station. Range= 0 - 100 Default = 1 Response = 6 digits
	PITCH:	Requests the pitch between slots in microns at the specified station. Default = 0 Response = 6 digits
	W UN:	*For rotating wrists only.* `wUNFlpVal`: Specifies unflip position of the end effector Rotated position axis location in units of millidegrees Response = 6 digits
	W FL:	*For rotating wrists only*. `wFlpVal`: Specifies flip position of the end effector Rotated position axis location in units of millidegrees Response = 6 digits
	S:	*This command is used with the Traverser option.* `sVal`: The station’s horizontal axis location in units of microns. Response = 6 digits
	W:	*This command is used with the 4-axis Armset.* `wVal`: Specifies the stations wrist location in millidegrees Response = 6 digits

Copyright © 2023, Brooks Automation, Inc.

Request Station		
	WA:	*This command is used with the 5-axis Armset.* `waVal`: Specifies the stations wrist location in millidegrees for the upper end effector. Response = 6 digits
	WB:	*This command is used with the 5-axis Armset.* `wbVal`: Specifies the stations wrist location in millidegrees for the lower end effector. Response = 6 digits
	Z_WAF:	*This command is used with the Z-axis option.* `zWafVal`: The value, in microns, set for the third WF position, wafer position, at the station. The value must be an absolute value between the UP position and the DN position. For example, if the BTO=50000 and LWR=1000, then a valid Z_WAF value is between 49001 and 49999. The 3_Z Station Option for Z position must be set to Yes. Response = 6 digits
Description	Requests the absolute coordinate values of the various station-related parameters for the specified station.	

Example	`RQ STN 1 ALL`
	`STN 0001  0780000  180000  38000  206500  5000  25  010000`
	
	`RQ STN 2 ARM B`
	`STN 2 ARM B R 0780000 T 090000 Z 38000 W 206500 LOWER 5000 NSLOTS 25 PITCH 010000`	

Remove Station		
Command Syntax	`REMOVE STN <<(ARM)>[(A|B)]> [`*`station`*`|ALL]`	
Response Syntax	There is no response to this command.	
Arguments	STN:	Station number Range 1 - 99
	ARM:	Arm descriptor `A`: Arm A `B`: Arm B
	ALL:	Removes all values for all axes and other parameters.

Copyright © 2023, Brooks Automation, Inc.

Remove Station	
Description	Removes the absolute coordinate values of the various station-related parameters for the specified station. ***NOTE:*** *The Remove Station All command will remove all 100 stations for both arms, all station options, and any VIAs.*
Example	`REMOVE STN 1 A`

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Station Option Curve Move

Sets, requests, and stores the Curve Move Mode options for using the VIA point in move commands. The CURVE_MOVE option allows operation of the VIA point moves. This command is used with additional station settings to perform compound moves rather than traditional pure radial moves by introducing a VIA point. Setting this value to Y without setting VIA points will produce an error when motion is commanded.

In WAVE, if VIA positions (R, T, DIST) are configured for a station, but CURVE_MOVE is not enabled, an error will be reported when the user attempts to access the station (GOTO, PICK, PLACE).

In LEAP, if VIA positions (R, T, DIST) are configured for a station, but CURVE_MOVE is not enabled, the robot will follow the radial path into the station.

This distinction allows users to more easily enable/disable the VIA path if desired, without having to CLEAR the VIA to access the station with the radial path and restore the VIA settings (R, T, DIST).

Set Station Option Curve Move		
Command Syntax	`SET STN station <<(ARM)>[(A|B)]> OPTION CURVE_MOVE (Y|N)`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	Mode:	VIA Point Curve Move mode `Y`: Sets the VIA option ON. `N`: Sets the VIA option OFF. VIA point moves are allowed only if a non-zero value for the point is entered and the CURVE_MOVE option is set to Y.
Description	Sets the Curve Move Mode options for using the VIA point in move commands.	
Example	`SET STN 1 OPTION CURVE_MOVE Y`	

Request Station Option Curve Move	
Command Syntax	`RQ STN station <<(ARM)>[(A|B)]> OPTION CURVE_MOVE`
Response Syntax	`STN station(Y|N)`

Copyright © 2023, Brooks Automation, Inc.

Request Station Option Curve Move		
Arguments	station:	Station number
	ARM:	Arm descriptor
	Mode:	VIA Point Curve Move mode `Y`: The VIA option is set to ON. `N`: The VIA option is set to OFF
Description	Requests the Curve Move Mode options for using the VIA point in move commands.	
Example	`RQ STN 1 OPTION CURVE_MOVE` `STN 1 OPTION CURVE_MOVE Y`	

Store Station Option Curve Move		
Command Syntax	`STORE STN` *`station`* `<<(ARM)>[(A|B)]> OPTION CURVE_MOVE`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	Mode:	VIA Point Curve Move mode `Y`: Stores the VIA option ON. `N`: Stores the VIA option OFF.
Description	Stores the Curve Move Mode options for using the VIA point in move commands.	
Example	`STORE STN 1 OPTION CURVE_MOVE Y`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Station Option VIA

Sets, requests, and stores the VIA point for Off-Center functions for the specified station.

The Off-Center pick, place, and transfer feature allows the robot to execute compound move trajectories (Curve Move).

- Simultaneous rotation and extension.
- Combination of straight lines and curves.
- Allows pick and place to cassettes and pods off the radial axis without interference.
- Minimizes unnecessary acceleration and deceleration experienced with sequential moves.
- Computational efficient implementation using pre-defined optimal trajectory segments where a single constraint is active.

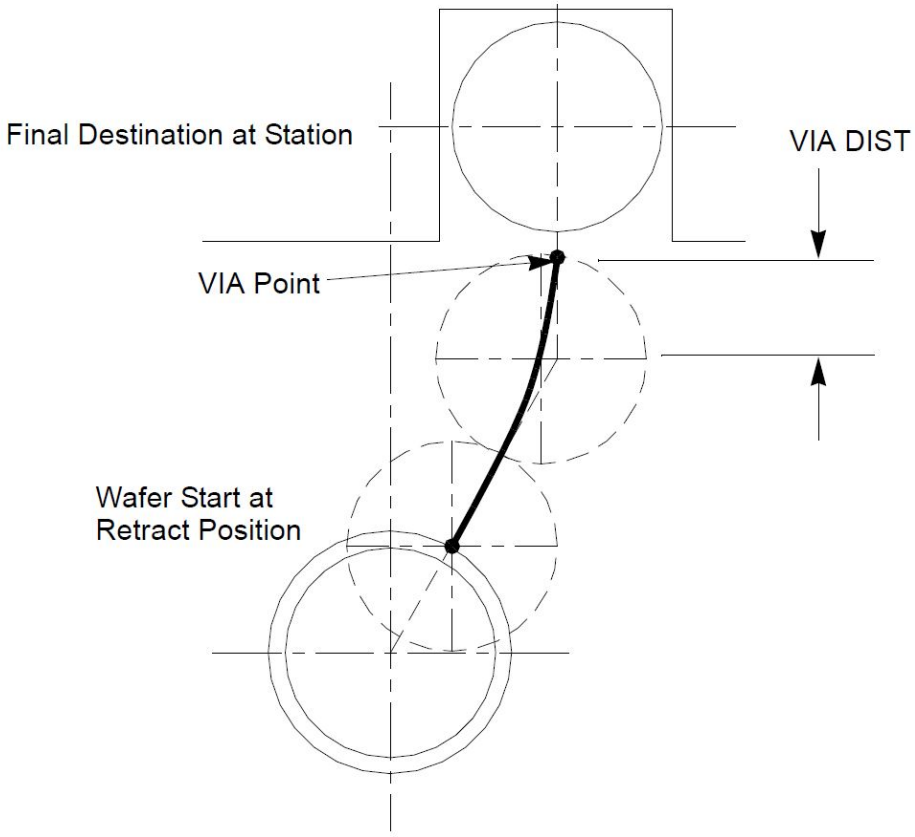


Figure 8-2: Off=Center PICK and PLACE

The VIA is defined as the point that the end effector moves through to perform a compound or curve move. The VIA point determines the point from which the station is approached and is specified in terms of R, T, and W coordinates.

Copyright © 2023, Brooks Automation, Inc.

Figure 8-2 shows that the end effector first moves through an intermediate (VIA) point on the way to its final destination.

To set-up the OCP feature, two positions must be defined:

- The station values
- The VIA value: a location radially outward from the station to which the robot travels in a straight path in order to remove a wafer without interference.

The travel path of the arm is adjusted for optimal travel in relation to the origin and destination of these two taught positions. This travel path is followed for both pick and place functions.

Using the W coordinate with the VIA specifications allows for adjustment of the end effector orientation while approaching the destination. This feature is beneficial when space or reach constraints do not allow the arm to pass through the VIA point with the end effector pointing in accordance with the destination orientation. Changing the W default values is restricted for stations without a VIA point.

To perform straight, non-compound moves to a specified station, set the value of POSR to zero for that station. The value for POST is remembered by the robot, but is not used unless the POSR value is set to non-zero again.

If a VIA point is set, modifications to station values are not allowed if R is changed more than 10mm or T is changed more than 3°. To change these stations values, the VIA point must be cleared first. To clear the VIA point, set the station option for the VIA point to zero or enter SET STN station OPTION VIA CLEAR.

NOTE: If station coordinates are set using the SET STN or SET STN OPTION VIA commands, they should be verified before performing any wafer transfers to ensure accurate station definition.

Set Station Option VIA	
Command Syntax	`SET STN station <<(ARM)>[(A|B)]> OPTION VIA [DIST viaDist][POSR viaPosR|POST viaPosT|POSW viaPosW|POSWA viaPosWa|POSWB viaPosWb] [CLEAR]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Station Option VIA		
Arguments	station:	Station number
	ARM:	Arm descriptor
	DIST:	`viaDist`: The distance value, in microns, for the specified VIA point.
	POSR:	`viaPosR`: The radial coordinate of the VIA point in microns. A non-zero value for POSR causes Compound Moves to occur for the designated station. Setting this value to CLEAR causes only straight moves to occur for the designated station.
	POST:	`viaPosT`: The theta coordinate of the VIA point in millidegrees.
	POSW:	This command is used with the 4-axis armsets only. `viaPosW`: The rotating wrist coordinate of the VIA point in millidegrees. Response = 6 digits
	POSWA:	This command is used with the 5-axis armsets only. `viaPosWa`: The Articulated Wrist coordinate for Arm A VIA point in millidegrees. Response = 6 digits
	POSWB:	This command is used with the 5-axis armsets only. `viaPosWb`: The Articulated Wrist coordinate for Arm B VIA point in millidegrees. Response = 6 digits
	CLEAR:	Removes the VIA point. .
Description	Sets the VIA point for Off-Center functions for the specified station.	

Example	`SET STN 1 OPTION VIA DIST 10000`
	
	`SET STN 1 OPTION VIA POSR 621000 POST 2234000 POSW 234000`	

Request Station Option VIA	
Command Syntax	`RQ STN` *`station`* `<<(ARM)>[(A|B)]> OPTION VIA [(DIST|POSR|POST|POSW|POSWA|POSWB`
Response Syntax	`STN` *`station viaDist viaPosR viaPosT viaPosW viaPosWa viaPosWb`* `(Y|N)`

Copyright © 2023, Brooks Automation, Inc.

Request Station Option VIA		
Arguments	station:	Station number
	ARM:	Arm descriptor
	DIST:	`viaDist`: The distance value, in microns, for the specified VIA point. Response = 6 digits
	POSR:	`viaPosR`: The radial coordinate of the VIA point in microns. A non-zero value for POSR causes Compound Moves to occur for the designated station. Setting this value to CLEAR causes only straight moves to occur for the designated station.
	POST:	`viaPosT`: The theta coordinate of the VIA point in millidegrees.
	POSW:	This command is used with the 4-axis armsets only. `viaPosW`: The rotating wrist coordinate of the VIA point in millidegrees. Response = 6 digits
	POSWA:	This command is used with the 5-axis armsets only. `viaPosWa`: The Articulated Wrist coordinate for Arm A VIA point in millidegrees. Response = 6 digits
	POSWB:	This command is used with the 5-axis armsets only. `viaPosWb`: The Articulated Wrist coordinate for Arm B VIA point in millidegrees. Response = 6 digits
	PATH_OVER_CENTER:	This command is used with the 4-axis and 5-axis armsets only. Allows the wrist to pass over or go around Singular Point. **`Y`**: Move through the center point. **`N`**: Move around the center point.
	CLEAR:	Sets the VIA Point values to **CLEAR**.
Description	Requests the VIA point for Off-Center functions for the specified station.	
Example	`RQ STN 2 OPTION VIA DIST VIA POSR VIA POST VIA POSW` `STN 0002 VIA DIST 000000 VIA POSR 344702 VIA POST 069988 VIA POSW 069988`	

Store Station Option VIA	
Command Syntax	`STORE STN` *`station`* `<<(ARM)>[(A|B)]> OPTION VIA [(DIST|POSR|POST|POSW|POSWA|POSWB|PATH_OVER_CENTER)]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Station Option VIA		
Arguments	station:	Station number
	ARM:	Arm descriptor
	DIST:	Stores the distance value for the specified VIA point in uM.
	POSR:	Stores the radial coordinate of the VIA point.
	POST:	Stores the theta coordinate of the VIA point in uM.
	POSW:	This command is used with the 4-axis armsets only. `viaPosW`: The rotating wrist coordinate of the VIA point in millidegrees. Response = 6 digits
	POSWA:	This command is used with the 5-axis armsets only. `viaPosWa`: The Articulated Wrist coordinate for Arm A VIA point in millidegrees. Response = 6 digits
	POSWB:	This command is used with the 5-axis armsets only. `viaPosWb`: The Articulated Wrist coordinate for Arm B VIA point in millidegrees. Response = 6 digits
	PATH_OVER_CENTER:	Stores the path over center status.
	CLEAR:	Clears the VIA point values.
Description	Stores the VIA point for Off-Center functions for the specified station.	
Example	`STORE STN 1 A OPTION VIA` `DIST 10000 POSR 610000 POST 43000`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Station Option Z

Sets, requests, and stores the optional station-related parameters for the Z-axis.

The Set Station Option command requires the station number and a variable to assign the related operation to a specific station. The Z_UP_SLOW and the Z_DOWN_SLOW parameter allows the place of a wafer at the specified station to be performed in low speed.

3_ZPOS: The 3-position station operation allows the WF feature used in the GOTO command to be enabled (the station has three positions: UP, WF, DN) or disabled (the station has two positions: UP, DN). The WF or third position is an absolute Z coordinate value between the UP and the DN positions. The value for the WF position is set using the Z_WAF parameter.

NOTE: In WAVE, once one of these settings (ONE_Z_MOVE and 3_ZPOS_STN) is assigned to 'Y' for a station, the other cannot be assigned at all (the software will report an error).

In LEAP, once one of these settings (ONE_Z_MOVE and 3_ZPOS_STN) is assigned to 'Y' for a station, the other cannot be assigned to 'Y' (the software will report an error if this is attempted) but it can be assigned to 'N' (no error is reported).

The 3-position station is also affected when using the Active Grip in a PICK or PLACE command. If the Z option is set to Y (SET STATION station OPTION 3_ZPOS_STN Y) and the PICK command is issued using the GRIP option, the grip feature will automatically check the 3-position option and activate the grip at the WF position. If the Z option is set to N and the PICK command is issued using the GRIP option, the grip will activate at the Z UP position.

Set Station Option Z	
Command Syntax	`SET STN station <<(ARM)>[(A|B)]> OPTION [Z_DOWN_SLOW (Y|N)] [Z_UP_SLOW (Y|N)] [3_ZPOS_STN (Y|N)]]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Station Option Z		
Arguments	station:	Station number
	ARM:	Arm descriptor
	Z_DOWN_SLOW :	Mode of Z down speed during PLACE. **Y:** Z down moves at slow speed. **N:** Z down moves at normal speed.
	Z_UP_SLOW :	Mode of Z up speed during PICK. **Y:**Z up movesat slow speed. **N:** Z up moves at normal spped.
	3_ZPOS_STN :	Mode of 3_ZPOS_STN operation. **Y:** Turns the WF feature ON. **N:** Turns the WF feature OFF.
	ONE_Z_MOVE	Use the averaged offset of both pans to correct the wafer offset with a single z down. **Y:** Turns the ONE_Z_MOVE feature on. **N:** Turns the ONE_Z_MOVE feature off.
Description	Sets the optional station-related parameters for the Z-axis.	
Example	`SET STN 1 OPTION Z_UP_SLOW Y` `SET STN 1 OPTION Z_DOWN_SLOW Y`	

Request Station Option Z		
Command Syntax	`RQ STN station <<(ARM)>[(A|B)]> OPTION [Z_DOWN_SLOW|Z_UP_SLOW|3_ZPOS_STN]`	
Response Syntax	`STN` *`station`*`(Y|N)(Y|N)(Y|N)`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	Z_DOWN_SLOW :	Requests status of Z_DOWN_SLOW: **Y:** Z down moves slow. **N:** Z down moves normal.
	Z_UP_SLOW :	Requests status of Z_UP_SLOW: **Y:**Z up moves slow (HOME speed). **N:** Z up moves normal.
	3_ZPOS_STN :	Requests status of the 3_ZPOS_STN: **Y:** WF feature ON. **N:** WF feature OFF.
Description	Requests the optional station-related parameters for the Z-axis.	

Copyright © 2023, Brooks Automation, Inc.

Request Station Option Z	
Example	`RQ STN 2 ARM B OPTION 3_ZPOS_STN` `STN 2 ARM B 3_ZPOS_STN Y`

Store Station Option Z	
Command Syntax	`STORE STN station <<(ARM)>[(A|B)]> OPTION [Z_DOWN_SLOW|Z_UP_SLOW|3_ZPOS_STN]`
Response Syntax	There is no response to this command.
Arguments	See the arguments listed in "Request Station Option Z".
Description	Stores the optional station-related parameters for the Z-axis.
Example	`STORE STN 1 ARM A OPTION 3_ZPOS_STN`

Set/Request/Store Station Option I/O

Sets, requests, and stores the various optional station-related parameters for sensors.

The Set Station Option command requires the station number and a variable to assign the related operation to a specific station. This command is used to define the location and operation of all sensors linked to the robot and used in the system where the robot is installed.

The Store Station Option command requires the station number and all variables. This command is used to store the location and operation of all wafer sensors used in the system the robot is installed in. This command can also be used to store the amount that the wafer may be moved on the end effector during a pick or place operation at the specified station.

NOTE: The preferred method for setting EX/RE sensors and R_MT sensors is with the single-step command . The SET STN OPTION command on this page requires two steps.

Set Station Option IO	

Command Syntax	`SET STN station [<ARM> (A|B)] OPTION`
	`[SVLV_SEN name |RETRACT_SEN name|EX_ENABLE name|VLV_SEN name|NONE name]`
	`[WAF_SEN (EX name|RE name|R_MT name)]`

Copyright © 2023, Brooks Automation, Inc.

Set Station Option IO		
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	SVLV_SEN name:	Specifies the name of the slot valve sensor at the specified station using a single bit.
	RETRACT_SEN name:	Specifies the name of the retract sensor at the specified station.
	EX_ENABLE name:	Specifies the name of the extend sensor at the specified station. This command is designed for the Extend type sensor only.
	VLV_SEN name:	Specifies the name of the valve closed sensor at the specified station.
	NONE name:	Eliminates an already defined sensor.
	WAF_SEN:	Specifies the location and name of the wafer sensor at the specified station. **EX name**: Sensor at the extended position (in the Process Module) **RE name**: Sensor at the retracted position (in the Transport Chamber) **R_MT name**: Sensor at the radial motion position (in the Transport Chamber)
	name:	In all cases, *name* specifies the existing name of the sensor assigned previously using the command Map/Request/Remove Map I/O.
Description	Sets the various optional station-related parameters for sensors.	

Example	Both commands below are identical and set the station options at station #4 for arm 'A' for sensor number 17 in the extended position to active high.
	`SET STN 4 ARM A OPTION WAF_SEN EX STN_4_WAFR_SEN`
	or
	`SET STN 4 A OPTION WAF_SEN EX STN_4_WAFR_SEN`	

Request Station Option IO	
Command Syntax	`RQ STN station [<ARM> (A|B)] OPTION [SBIT_SVLV_SEN|RETRACT_SEN |EX_ENABLE|VLV_SEN] |WAF_SEN (EX|RE|R_MT|LEFT|RIGHT)`
Response Syntax	`STN station [<ARM> (A|B)] [SBIT_SVLV_SEN|RETRACT_SEN |EX_ENABLE|VLV_SEN]|WAF_SEN (EX|RE|R_MT|LEFT|RIGHT)`

Copyright © 2023, Brooks Automation, Inc.

Request Station Option IO		
Arguments	station:	Station number
	ARM:	Arm descriptor
	SVLV_SEN:	Requests the name of the slot valve sensor at the specified station using a single bit.
	RETRACT_SEN:	Requests the name of the retract sensor at the specified station.
	EX_ENABLE:	Requests the name of the extend sensor at the specified station. This command is designed for the Extend type sensor only.
	VLV_SEN:	Requests the name of the valve closed sensor at the specified station.
	WAF_SEN:	Requests the the location and name of the wafer sensor at the specified station. EX name: Sensor at the extended position (in the Process Module) RE name: Sensor at the retracted position (in the Transport Chamber) R_MT name: Sensor at the radial motion position (in the Transport Chamber)
	name:	In all cases, *name* specifies the existing name of the sensor assigned previously using the command Map I/O.
Description	Requests the various optional station-related parameters for sensors.	

Example	In the following example, the Request command is used to verify the sensor name for the R_MT sensor set for the Left Pan of the QuadraFly using Set/Request/Store/Remove Station.
	
	`RQ STN 1 OPTION WAF SEN LEFT R_MT`
	`STN 0001 A STN01ASENSOR_LEFT`	

Store Station Option IO	
Command Syntax	`STORE STN` *`station`* `[[ARM] (A|B)] OPTION {Z_DOWN_SLOW|Z_UP_SLOW|3_ZPOS_STN|CURVE_MOVE|4_AXIS_STN|RETRACT_SEN|EXTEND_SEN|EX_ENABLE|VLV_SEN|SBIT_SVLV_SEN|INTLCK TIMEOUT|NEAR_STN}`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Station Option IO		
Arguments	station:	Station number
	ARM:	Arm descriptor
	SVLV_SEN:	Stores the slot valve sensor at the specified station using a single bit.
	RETRACT_SEN:	Stores the retract sensor at the specified station.
	EX_ENABLE:	The name of the extend sensor at the specified station. This command is designed for the Extend type sensor only.
	VLV_SEN:	The name of the valve closed sensor at the specified station.
	WAF_SEN:	Specifies the location and name of the wafer sensor at the specified station. EX name: Sensor at the extended position (in the Process Module) RE name: Sensor at the retracted position (in the Transport Chamber) R_MT name: Sensor at the radial motion position (in the Transport Chamber)
Description	Stores the station's option settings.	
Example	`STORE STN 1 OPTION RETRACT_SEN`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Station Facet

A facet is the area on a Transport Module where Process Modules, or other types of modules, can be connected for access by the central wafer handler. Typically, a Transport or Process Module can have two to eight facets. See Figure 1-1 for an example of a machine with six facets.

The facet value provides the user with an indication of the station to which the robot is pointing.

The user can arbitrarily define a 3-digit number for each station. The robot at Home will have a different value for any of the facets.

A typical configuration would return a facet value of 1 for station 1, 2 for station 2, etc. Home usually returns 0.

Set Station Facet		
Command Syntax	`SET STN station [<ARM> (A|B)] FACET value`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	FACET:	Value = number of facet (1 to 16)
Description	The facet value provides the user with an indication of the station to which the robot is pointing.	

Example	To set up a facet value of 1 when the robot is placing at the station at physical facet #1:
	`MAP FACET_OUT STN_FACET HI TO DIGITAL_OUT 0x0000001c`
	
	`SET STN 1 ARM A FACET 01`
	
	`SET STN 1 ARM B FACET 01`	

Request Station Facet		
Command Syntax	`RQ STN station [<ARM> (A|B)] FACET`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
Description	Provides the facet value associated with the requested station.	

Copyright © 2023, Brooks Automation, Inc.

Request Station Facet	
Example	`RQ STN 1 ARM A FACET` `STN 0001 ARM A FACET 1`

Store Station Facet		
Command Syntax	`STORE STN` *`station`* `[<ARM> (A|B)] FACET`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
Description	Stores a unique value for the facet associated with the requested station.	
Example	`STORE STN 1 ARM A FACET`	

Set/Request/Store Facet Delay

When using facets, the facet delay is a user-specified delay that occurs before the facet number is set on the outputs. The facet delay is generally used with the safety hub.

Set Facet Delay	
Command Syntax	`SET FACET DELAY`
Response Syntax	There is no response to this command.
Arguments	`DELAY`
Description	Sets a user-specified delay before the facet number appears on the outputs (microseconds).
Example	`SET FACET DELAY 20000`

Request Facet Delay	
Command Syntax	`STORE FACET DELAY`
Response Syntax	`FACET DELAY 20000` Number back (in microseconds).
Arguments	None

Copyright © 2023, Brooks Automation, Inc.

Request Facet Delay	
Description	Provides the current facet delay in the robot (microseconds).
Example	`REQUEST FACET DELAY` `FACET DELAY 20000`

Store Facet Delay	
Command Syntax	`STORE FACET DELAY`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores a user-specified delay before the facet number appears on the outputs (micro-seconds).
Example	`STORE FACET DELAY`

Copy Station

The Copy Station command copies a defined robot station (source) to another robot station (destination). This can be useful in setting up a B station's initial parameters from the associated A station, and then the B station parameters can be adjusted slightly as needed. This command copies all the source's physical coordinates and any station options that are selected for that station.

The source station must have at least one of the settings for R, T, or Z non-zero. If this condition is not met, an error response will result.

> ***NOTE:*** Using COPY STN in WAVE reports an error if the destination station has AWC, VIA, or R_MT options set. LEAP simply overwrites all destination station options with those of the source station.

Copy Station		
Command Syntax	`COPY STN STATION [[ARM] (A|B)] TO STATION [[ARM] (A|B)]`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
Description	Copies a defined robot station (source) to another robot station (destination).	

Copyright © 2023, Brooks Automation, Inc.

Copy Station	

Example	Copy station to a different station:
	`COPY STN 1 A TO 2 A`
	
	Copy ARM A station to ARM B station:
	`COPY STN 1 A TO 1 B`
	
	Copy station to a different station (abbreviated):
	`COPY STN 1 TO 41`

Set/Request/Store Active Handoff Option

Sets, requests, and stores the optional station-related parameters for the Active Handoff feature.

Handoff can be used during PICK, PLACE and XFER commands. The end effector will extend to PICK or PLACE position and set the IO output to indicate ready for handoff. When the IO input is set by an external mechanism, the robot will return (in its original path) to the retracted position.

The SET STN OPTION command requires the station number and a variable to assign the related operation to a specific station. With the HANDOFF feature Enabled, the robot employs the Active Handoff feature during PICK and PLACE commands defined as follows.

First Segment of PICK

Step	Action
1.	Enter the following command: `PICK station`
2.	Retract, if required.
3.	Rotate to the station's Theta position and simultaneously move to the Z Down position for the station.
4.	Extend to the station's R position.
5.	Move to the station's Z Up position.
6.	Robot asserts Handoff output.

Copyright © 2023, Brooks Automation, Inc.

Second segment of PICK

Step	Action
7.	Robot receives Handoff input signal from external mechanism.
8.	Activate the grip (if PLACE, release grip).
9.	Move to the station's Z Down position (with Handoff Enable).
10.	Retract the arm.

If the Handoff is disabled, the PICK command may still be converted into a two segment Handoff by using the STRT and ENRT options of the PICK command. The vacuum grip may also be controlled in the Active Handoff mode without setting the DIO (handoff_input = NONE).

```
PICK 1 SLOT 5 ENRT NR GRIP ON
```

If a PICK is performed, the vacuum grip will be turned ON after the robot receives the HANDOFF_INPUT in the active state.

If a PLACE is performed, the vacuum grip will be turned OFF prior to setting the HANDOFF_OUTPUT bit to the active state.

Set Active Handoff Option		
Command Syntax	`SET STN station [<ARM> (A|B)] OPTION`	
Response Syntax	`[HANDOFF_OUTPUT IO_name|HANDOFF_INPUT IO_name]` `[HANDOFF_ENABLE (ENB|DIS)]`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	HAND_OFF ENABLE:	Mode of Active Handoff feature ENB: Enabled DIS: Disabled
	HAND_OFF INPUT:	Robot receives Handoff input signal. IO_name: User name assigned to bit.
	HAND_OFF OUTPUT:	Robot asserts Handoff output signal. IO_name: User name assigned to bit.
Description	Sets the optional station-related parameters for the Active Handoff feature.	

Copyright © 2023, Brooks Automation, Inc.

Set Active Handoff Option	

Example	`SET STN 99 OPTION`
	`HANDOFF_OUTPUT`
	`HANDOFF_OUT_1`
	`HANDOFF_ENABLE ENB`

Request Active Handoff Option		
Command Syntax	`RQ STN` *station* `[<ARM> (A|B)] OPTION (HANDOFF_OUTPUT|HANDOFF_INPUT|HANDOFF_ENABLE)`	
Response Syntax	`STN station io_name`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	HAND_OFF ENABLE:	Mode of Active Handoff feature ENB: Enabled DIS: Disabled
	HAND_OFF INPUT:	Robot receives Handoff input signal. IO_name: User name assigned to bit.
	HAND_OFF OUTPUT:	Robot asserts Handoff output signal. IO_name: User name assigned to bit.
Description	Requests the optional station-related parameters for the Active Handoff feature.	
Example	`RQ STN 99 OPTION HANDOFF_OUTPUT` `STN 0099 HANDOFF_OUT_1`	

Store Active Handoff Option		
Command Syntax	`STORE STN` *station* `[<ARM> (A|B)] OPTION (HANDOFF_OUTPUT|HANDOFF_INPUT) (HANDOFF_ENABLE)`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	HAND_OFF ENABLE:	Mode of Active Handoff feature ENB: Enabled DIS: Disabled
	HAND_OFF INPUT:	Robot receives Handoff input signal. IO_name: User name assigned to bit.
	HAND_OFF OUTPUT:	Robot asserts Handoff output signal. IO_name: User name assigned to bit.

Copyright © 2023, Brooks Automation, Inc.

Store Active Handoff Option	
Description	Stores the optional station-related parameters for the Active Handoff feature.
Example	`STORE STN 1 ARM A HANDOFF_OUTPUT`

Exclusion Zones

An exclusion zone is a shared space between two robots. Each robot can access the shared space, but not at the same time. The exclusion zone I/O signals are used to ensure that only one robot accesses the exclusion zone at a time.

The exclusion zone I/Os are designed to work together to control access to a station, for example between two robots in a single chamber. When a station is set with EXCL_IO_CTRL, a command to access a station (GOTO/PICK/PLACE) will first set the assigned EXCL_IO_OUPUT to request access to the exclusion zone station. The command will then wait for EXCL_IO_TIMEOUT period for a response on the EXCL_IO_INPUT IO. If the input is set, the command will proceed. If the input does not get set in the allotted time an error 235 will be thrown. EXCL_IO_OUTPUT will remain set until the robot has retracted from the station.

Map Active Handoff Option		
Command Syntax	`MAP EXCL_OUT EXCL_REQ HI TO DIGITAL_OUT 0x2`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	HAND_OFF ENABLE:	Mode of Active Handoff feature ENB: Enabled DIS: Disabled
	HAND_OFF INPUT:	Robot receives Handoff input signal. IO_name: User name assigned to bit.
	HAND_OFF OUTPUT:	Robot asserts Handoff output signal. IO_name: User name assigned to bit.
Description	Defines and maps an I/O bit used by two robots to coordinate access to a shared space.	

Example	`MAP USER_INPUT`
	`EXCL_CTRL LOW TO`
	`DIGITAL_IN 0x1`	

Copyright © 2023, Brooks Automation, Inc.

Set Active Handoff Option		
Command Syntax	`SET STN station [<ARM> (A|B)] OPTION`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	HAND_OFF ENABLE:	Mode of Active Handoff feature ENB: Enabled DIS: Disabled
	HAND_OFF INPUT:	Robot receives Handoff input signal. IO_name: User name assigned to bit.
	HAND_OFF OUTPUT:	Robot asserts Handoff output signal. IO_name: User name assigned to bit.
Description	Allows the user to turn the handoff I/O bit on or off.	
Example	`SET STN 99 ARM A OPTION EXCL_IO_CTRL Y`	

Request Active Handoff Option		
Command Syntax	`RQ STN station [<ARM> (A|B)] OPTION (HANDOFF_OUTPUT|HANDOFF_INPUT|HANDOFF_ENABLE)`	
Response Syntax	`STN number Y|N`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	HAND_OFF ENABLE:	Mode of Active Handoff feature ENB: Enabled DIS: Disabled
Description	Determines if a particular station has the handoff option enabled.	
Example	`RQ STN 99 ARM A OPTION EXCL_IO_CTRL`	

Store Active Handoff Option	
Command Syntax	`STORE STN station [<ARM> (A|B)] OPTION (HANDOFF_OUTPUT|HANDOFF_INPUT) (HANDOFF_ENABLE)`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Active Handoff Option		
Arguments	station:	Station number
	ARM:	Arm descriptor
	HAND_OFF ENABLE:	Mode of Active Handoff feature ENB: Enabled DIS: Disabled
	HAND_OFF INPUT:	Robot receives Handoff input signal. IO_name: User name assigned to bit.
	HAND_OFF OUTPUT:	Robot asserts Handoff output signal. IO_name: User name assigned to bit.
Description	Allows the user to store the handoff option.	
Example	`STORE STN 99 ARM A OPTION HAND_OFF ENABLE`	

Copyright © 2023, Brooks Automation, Inc.

9. GoTo and Transfer Commands

Go To

Moves to a specified station-referenced location.

> ***NOTE:*** This command performs all interlocking necessary to maintain safe wafer handling.

Any or all of the data fields may be specified on a single command line. If the arm is not already at a station, N must be specified as part of the command. Otherwise, a “not at station” error will occur. All motions will follow the speed and acceleration profile appropriate for the currently defined load.

> ***NOTE:*** The SET LOAD (ON|OFF) command may be used to define the load status of the robot’s arms before or after executing the GOTO command. The GOTO command will not update the LOAD state.

For multi-axis moves, the following sequence of checks and motions occur in the order given for the currently defined load:

- If “N” is specified (station #), or if “SLOT” is specified, or if “R RE” (retract) is specified, the arm will retract if it is not already retracted.
- The rotation axis (variable N) and the Z axis (variables SLOT and Z) will move to their target locations simultaneously. If N is specified and SLOT is not, the slot is assumed to be #1. If N is specified and Z is not, the position is assumed to be Down.
- The arm will extend if so commanded. If no arm is specified, ARM A will extend. This means that unless the arm is explicitly commanded to extend as part of a GOTO command that specifies a Station or Slot number, it will remain in the retracted position. This is true even if the arm is already at the specified Station or Slot number.

Copyright © 2023, Brooks Automation, Inc.

Go To	
Command Syntax	GOTO[N station] <<(ARM)>[(A|B)]> [[MAP(EX|RE)](MAP)])(VIA)) {((R)[(RE|EX|R2|PUSH|SAFETY)] <(NO_VIA)>) [Z(WF|UP|DN)] [ALTWAFSPD (Y|N)] ((N goTo.stnNum)) [SLOT goTo.slotNum] [(MAT)(ON|OFF)] <<ARM> (A|B|AB)>) [W (UN|FL)] [NWR2 goTo.noWristR2] [(PATH)<(POINT) intpointNum >)}] *In MAG6 Compatibility, entering the word ARM is optional; in VT5 Compatibility, ARM is required and Wafer Speed is used regardless of the load status.*
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Go To		
Arguments	station:	Station number
	ARM:	Arm descriptor *Arm AB is used with Radius armsets only.*
	slot:	*For use with Z-axis drive.* Indicates the specific slot to which the arm should move. See "Set/Request/Store/Remove Station" on page 145.
	MAP N:	*For use with Wafer Present Sensors.* Moves radial position of arm at MAP position. `EX`: Move to station's MAP position and Extend. `RE`: Move to station's MAP position. ***NOTE:*** *GOTO MAP is the same as GOTO MAP RE.*
	VIA:	Moves to the VIA point associated with the station. See " Set/Request/Store Station Option VIA" on page 153.
	R (EX|RE|R2) <NO VIA>	Specifies radial position of arm. `EX`: Extended. `EX NO_VIA`: Extend without VIA point. `RE`: Retracted (default). `RE NO_VIA`: Retract without VIA point. `R2`: Secondary retract position (Retract 2).
	Z:	*For use with Z-axis drive.* Specifies vertical deployment of arm. `UP`: Up. `DN`: Down (default). `WF`: Optional third position; wafer position.
	MAT:	*For use with R_MT sensors only.* Checks for the presence of Material on the end effector using the Radial Motion sensors. If the material state is ON and the command GOTO N station MAT OFF was sent, an error will be reported. `ON`: Enter ON if Material is suspected on the end effector. `OFF`: Enter OFF if end effector is suspected empty.
	ALTWAFSPD:	Allows the user to select an alternate speed setting for vacuum robots that have a dual Wafer Speed configuration. `Y` = Use alternate speed. `N` = Do not use alternate speed. Default = N.
Description	Moves to a specified station-referenced location.	

Copyright © 2023, Brooks Automation, Inc.

Go To	

Example	Robot has a Z-axis drive and a Rotational Wrist.
	Starting position: Station 1, retracted, Z down, W unflipped
	
	To go to Station 2, which is a flipped station, and the command was GOTO N 2:
	a. Goes to the R2 position if Load state is ON
	b. Move in T and Z to Station 2 at the Z UP position (still at the Retract Position)
	c. Rotate W and T and move to Z-UP Flipped Position
	To go to Station 2, which is a flipped station, and the command was entered GOTO N 2 NWR2
	a. Move in T , W and Z to Station 2 at the Z UP position (without going to Retract2 Position), if Load state is ON
	If Station 1 has a unflipped value of 0 and a flipped value 180000 and the command was entered GOTO W FL:
	a. Retract to R2
	b. Rotate W to the flipped position
	c. Extend back to the retract position
	
	In the following example arm 'A' is currently extended in station #5, slot #2 and in the down position.
	
	`GOTO Z UP`
	The Arm A will move to the up position without retracting
	
	In the following example arm 'A' is currently retracted.
	
	`GOTO N 5 ARM A`
	The robot will rotate arm 'A' to station 5
	
	`GOTO N 2 R EX Z UP SLOT 2 ARM B MAT ON`

Copyright © 2023, Brooks Automation, Inc.

Go To Offsets

The Offset value represents absolute coordinate offsets to/from the destination station position. Offset values can be positive or negative. Offsets are defined by Coordinate System (see "Coordinate System" on page 113). Offsets can be applied individually or in any combination, but the combination must be in the same Coordinate System.

The offset limits are checked according to the limits for each axis. These limits are hard coded and cannot be set or requested.

Offsets can be applied ONLY for motion to or into the extended position.

- Robot could be in any position except extended and robot will be in extended position after GOTO execution.
- Robot could be in extended position and it will stay in extended position after GOTO execution. It means that GOTO command must contain R EX; otherwise, the error “Command not supported” will be reported.
- Z offset will effect the robot position if GOTO command contains not only R EX, but also Z UP; otherwise, robot wouldn’t go to (BTO+Z offset) position.
- Offset(s) cannot be used with any other GOTO command: GOTO R PUSH, GOTO R SAFETY, GOTO R RE, GOTO R R2, GOTO VIA, GOTO MAP EX or GOTO MAP RE.

Compound offsets (that is, RO and TO) from VIA to EX position could cause the error: “Error, unable to calculate trajectory”, but the motion from RETRACTED position to the same EXTENDED position with the same offsets shouldn't cause errors.

The axis offset can be reset by doing one of the following:

- Use another offset value (including 0).
- If the station number is not specified, issue the next GOTO R EX command with another offset value. Issuing the command GOTO R EX without an offset will not reset axis/axes offset(s).
- Specify the station number in the next GOTO command.

  The robot stays at Z position with offset if other axes positions (except Z position) have been changed. For example, if the robot is extended and is in Z UP position with Z offset, using the GOTO R RE command will not reset the offset.

NOTE: This command performs all interlocking necessary to maintain safe wafer handling.

Copyright © 2023, Brooks Automation, Inc.

GoTo Offset		

Command Syntax	If the robot is not at the station:
	`GOTO[N station] <<(ARM)>[(A|B)]> R EX [Z(UP|DN)]`
	`[[RO roffset][TO toffset][ZO zoffset]`
	`[WO woffset][WAO waoffset][WBO wboffset]]`
	`[(XO xoffset)(YO yoffset)]`
	
	If the robot is at the station:
	`GOTO R EX [Z(UP|DN)]`
	`[[RO roffset][TO toffset][ZO zoffset]`
	`[WO woffset][WAO waoffset][WBO wboffset]]`
	`[(XO xoffset)(YO yoffset)]`	

Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	R EX:	Specifies radial position of arm. `EX`: extended
	Z:	*For use with Z-axis drive*. Specifies vertical deployment of arm. `UP`: up `DN`: down (default)
	RO *roffset*:	Radial offset value Limits = + 10000 microns
	TO *toffset*:	Theta offset value Limits = + 2000 millidegrees
	ZO *Zoffset*:	Z offset value Limits = + 6000 microns Offset value is added to Robot BTO position and Z Offset value.
Description	Allows the user to add an arbitrary offset.	

Copyright © 2023, Brooks Automation, Inc.

GoTo Offset	

Example	***NOTE:*** *If the Z-axis offset is negative, it will be subtracted from the sum; if it is positive, it will be added to the sum.*
	
	Robot is at HOME position or at other station
	`GOTO N station R EX RO roffset TO toffset ZO zoffset`
	
	Z offset will be ignored.
	Robot rotates to the station T with T offset position.
	Robot extends to the station R with R offsets position.
	
	`GOTO N station R EX Z UP RO roffset TO toffset ZO zoffset`
	
	Robot moves to Z position = station Z position + Z offset.
	Robot rotates to the station T with T offset position.
	Robot extends to the station R with R offsets position.
	
	Robot is at retracted position or VIA position of the same station.
	`GOTO Z UP XO xoffset YO yoffset ZO zoffset`
	
	Robot moves to Z position = station Z position + Z offset.
	Robot calculates new R,T and W position based on the station R,T and W positions and X and Y offsets.
	Robot rotates to new T position.
	Robot extends to new R position Robot extends to new W position.
	
	`GOTO Z DN RO roffset TO toffset ZO zoffset WO woffset`
	
	Z offset will be ignored.
	Robot rotates to the station T with T offset position.
	Robot rotates to the station W with W offset position.
	Robot extends to the station R with R offsets position.

Copyright © 2023, Brooks Automation, Inc.

Pick

Picks a wafer from a specified station and slot number. The speed and acceleration at which the robot moves during a pick operation is dependent on the status of the pans: with or without wafers. The robot always moves at Wafer Speed (slow) speed and acceleration for all axes when there is a wafer on the active arm. For some robots, if the active arm is empty and the inactive arm contains a wafer, the robot moves at medium speed for the R axis, but slow speed for T and Z. If both arms are empty, the robot uses high speed for all axes.

During a pick operation, the robot executes the following sequence of moves.

a. Retracts the arm using a speed and acceleration profile appropriate for the currently defined load.
b. Simultaneously moves Z-axis down and rotates to the Down position. The robot chooses the shortest path at the Station and Slot number specified using a speed and acceleration profile appropriate for the currently defined load.
c. Extends the arm using a speed and acceleration profile appropriate for the currently defined load to the R position for the station.
d. Moves to the Up position using a speed and acceleration profile appropriate for the currently defined load and picks up a wafer.
e. Defines the arm executing the pick as being “loaded.” See "Set/Request Load" on page 125.
f. Retracts the arm using a speed and acceleration profile appropriate for the currently defined load. If ENRT NR, ENRT R2, or ENRT VIA is used, this may not happen.

NOTE: The operator can force a uniform high speed throughout the pick operation by first invoking the set high speed command. The set speed remains in effect only until the completion of the action command following the set speed command.

The operator can force the Z-axis speed during the PICK to slow speed using the Z_UP_SLOW command option.

If the 3_Z_POS_STN option is set to Y, for NR status, the robot stops at Z_WAF and activates the grip. If set to N, the robot stops at Z_UP.

For the dual pan robot using the pick command, the load status is very important. If the active pan is in proper state (off) and the passive pan load status is on, the pick command responds with error “Possible material on arm.” See "Set/Request Load" on page 125.

Pick	
Command Syntax	`PICK station [[ARM] (A|B|AB|BA)] [PAN (L|R)] [SLOT slot] [STRT (NR|R2|VIA)] [ENRT (NR|R2|VIA)] [EX NO_VIA] [RE NO_VIA] [RO r_offset] [TO t_offset] [ALTWAFSPD (Y|N)] [ALGN [PALGNID] [SLPS POSTNUM]] [PALGN [PALGNID]] [ADJ_STN] [VTV]`

Copyright © 2023, Brooks Automation, Inc.

Pick	
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Pick		
Arguments	station:	Station number
	slot:	Slot number
	ARM:	Arm descriptor
	PAN:	*This option is used with dual pan armsets only.* End effector `L`: Left Pan `R`: Right Pan Default = Both pans
	offset:	Offset Position `RO r_offset`: Specifies the positive or negative offset from the extend/retract location for that station. Maximum allowable R offset: ±4000 microns. `TO t_offset`: Specifies the positive or negative offset from the theta location for that station. Maximum allowable T offset: ±2000 millidegrees.
	EX NO_VIA|RE NO_VIA:	EX NO_VIA= During the PICK command, extend without using the VIA point. RE NO_VIA= During the PICK command, retract without using the VIA point.
	ALTWAFSPD:	Allows the user to select an alternate speed setting for vacuum robots that have a dual Wafer Speed configuration. `Y` = Use alternate speed. `N` = Do not use alternate speed. Default = N
	STRT:	Start retract location. This is only used if the station specified is the current station and the robot is at that position due to the previous use of the ENRT option or a GOTO command. `NR`: No retract `R1`: Normal retract `R2`: Second retract location `VIA`: Performs the PICK starting from the VIA location. Default = R1 The R2 value is set using SET RTRCT2. See "Set/Request/Store Retract 2 Value" on page 132. The VIA value is set using SET STN ARM OPTION VIA.

Copyright © 2023, Brooks Automation, Inc.

Pick		
	ENRT:	End retract location. `NR`: No retract `R1`: Normal retract `R2`: Second retract location `VIA`: Ends retract at the VIA location. Default = R1 The R2 value is set using SET RTRCT2. See "Set/Request/Store Retract 2 Value" on page 132. The VIA value is set using SET STN ARM OPTION VIA. See " Set/Request/Store Station Option VIA" on page 153.
	CW|CWW:	This former Wave option has been replaced with automatic selection. The robot will always choose the shortest path.
	ALGN:	After picking the wafer from the station, align the wafer and place the wafer at the station.
	SLPS:	Post Position index.
	ADJ_STN:	Wafer alignment is used to adjust the stored position information of a station
	VTV:	Starts at the current station's via point instead of going home, goes to the pick station's via point, and then goes to the pick station.
Description	Causes the robot to lift the wafer off of a station.	

Copyright © 2023, Brooks Automation, Inc.

Pick	

Example	In the following example, arm 'A' is currently extended in station #5, slot #2, and in the down position. The robot retracts the arm, rotates to station #2, extends the arm, raises the arm (picking up the wafer), and retracts the arm.
	
	NOTE: *Since the slot and arm are not specified, the robot defaults to slot #1 and arm 'A'.*
	`:PICK 1 (in DREP AUT mode)`
	`RE 01 0001 DN`
	`EX 01 0001 DN`
	`EX 01 0001 UP`
	`RE 01 0001 UP`
	
	`:PICK 1 ENRT NR`
	`RE 01 0001 DN UN`
	`EX 01 0001 DN UN`
	`EX 01 0001 UP UN`
	`EX 01 0001 UP UN`
	
	`:PICK 1 STRT NR`
	`EX 01 0001 UP UN`
	`EX 01 0001 DN UN`
	`EX 01 0001 UP UN`
	`RE 01 0001 UP UN`
	
	In the following example with the dual pan arm, PAN L is not involved:
	
	`PICK 2 B PAN R`

Copyright © 2023, Brooks Automation, Inc.

Place

Places a wafer at a specified station and (optionally) slot number.

The Place command is meant to be used with robots that have the Z-axis option installed. If using a 2-axis robot, this command may also be used for pick and place commands but no Z-axis motion will occur. No error will be issued.

The speed and acceleration at which the robot moves during a PLACE operation is dependent on the status of the pans: with or without wafers. The robot always moves at “with wafer” (slow) speed and acceleration for all axes when there is a wafer on the active arm. For some robots, if the active arm is empty and the inactive arm contains a wafer, the robot moves at medium speed for the R axis, but slow speed for T and Z. If both arms are empty, the robot uses high speed for all axes.

During a place operation, the robot executes the following sequence of moves:

- Retracts the arm using a speed and acceleration profile appropriate for the currently defined load.
- Simultaneously moves Z-axis up and rotates in the shortest direction to the Up position at the Station and Slot number specified using a speed and acceleration profile appropriate for the currently defined load.
- Extends the arm using a speed and acceleration profile appropriate for the currently defined load to the R position for the station minus the Safety distance. If the Safety distance = zero, the robot extends to the Station.
- Moves to the Down position using a speed and acceleration profile appropriate for the currently defined load depositing the wafer.
- Extends the arm using “with wafer” speed and acceleration profile to the R position for the station plus the “Push” distance.
- Defines the arm executing the PLACE as being “unloaded."
- Retracts the arm using a speed and acceleration profile appropriate for the currently defined load.
- If the 3_ZPOS_STN Option is set to Y and the PLACE *station* ENRT NR is issued:
  - The robot goes to the station BTO position,
  - Extends in the Z_UP position,
  - Moves to the Z_WAF position,
  - Activates the grip (GRIP OFF)
  - Stops.

  If the 3_ZPOS_STN Option is set to N and the PLACE *station* ENRT NR is issued:
  - The robot goes to the station BTO position,
  - Extends in the Z_UP position,
  - Activates the grip (GRIP OFF),
  - moves to the LOWER position (Z_DOWN),
  - Stops.
- See " Set/Request/Store Station Option Z" on page 158.

Copyright © 2023, Brooks Automation, Inc.

NOTE: The operator can force a uniform high speed throughout the PLACE operation by first invoking the SET HISPD command. The set speed remains in effect only until the completion of the action command following the set speed command.

The operator can force the Z-axis speed during the PLACE to slow by invoking the Z_DOWN_SLOW command option. See " Set/Request/Store Station Option Z" on page 158.

For the dual pan robot using the place command, the load status is very important. If the active pan is in proper state (On) and the passive pan load status is On, the place command will respond with error “Possible material on arm.”

Place	
Command Syntax	`PICK station [[ARM] (A|B|AB|BA)] [PAN (L|R)] [SLOT slot] [STRT (NR|R2|VIA)] [ENRT (NR|R2|VIA)] [EX NO_VIA] [RE NO_VIA] [RO r_offset] [TO t_offset] [ALTWAFSPD (Y|N)] [ALGN [PALGNID] [SLPS POSTNUM]] [PALGN [PALGNID]] [ADJ_STN] [VTV]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Place		
Arguments	station:	Station number
	slot:	Slot number
	ARM:	Arm descriptor
	PAN:	This option is used with dual pan armsets only. End effector `L`: Left Pan `R`: Right Pan Default = Both pans
	offset:	Retract 2 Position *r_offset*: Specifies the positive or negative offset from the extend/retract location for that station. Maximum allowable R offset: ±4000 microns. *t_offset*: Specifies the positive (clockwise) or negative (counterclockwise) offset from the theta location for that station. Maximum allowable T offset: ±2000 millidegrees. See "Set/Request/Store Retract 2 Value" on page 132.
	ALTWAFSPD:	Allows the user to select an alternate speed setting for vacuum robots that have a dual Wafer Speed configuration. `Y` = Use alternate speed. `N` = Do not use alternate speed. Default = N
	EX NO_VIA|RE NO_VIA:	EX NO_VIA: during the PLACE command, extend without using the VIA point. RE NO_VIA: during the PLACE command, retract without using the VIA point. If PLACE fails, the robot will retract through the VIA Point.
	STRT:	Start retract location. This is only used if the station specified is the current station and the robot is at that position due to the previous use of the ENRT option or a `GOTO` command. `NR`: No retract `R1`: Normal retract `R2`: Second retract location `VIA`: Performs the PICK starting from the VIA location. Default = R1 The R2 value is set using SET RTRCT2. See "Set/Request/Store Retract 2 Value" on page 132.
	ENRT:	End retract location `NR`: No retract `R1`: Normal retract `R2`: Second retract location `VIA`: Ends the retract at the VIA location. Default = R1 The R2 value is set using SET RTRCT2. See "Set/Request/Store Retract 2 Value" on page 132.

Copyright © 2023, Brooks Automation, Inc.

Place		
	CW|CCW:	This former Wave option has been replaced with automatic selection. The robot will always choose the shortest path.
Description	Places a wafer at a specified station and (optionally) slot number.	

Example	In the following example, arm 'A' is currently extended in station #2, slot #1 and in the up position. The robot will retract the arm, rotate to station #5, extend the arm, lower the arm (placing the wafer), push the wafer into registration, and retract the arm.
	
	NOTE: *Since the slot and arm are not specified the robot will default to slot #1 and arm 'A'.*
	
	`:PLACE 1 STRT NR`
	`EX 01 0001 UP UN`
	`EX 01 0001 UP UN`
	`EX 01 0001 EN UN`
	`RE 01 0001 DN UN`
	
	In the following example with the dual pan arm, PAN L is not involved:
	
	`PLACE 2 B PAN R`
	
	Place wafer at station 3:
	
	`PLACE 3`	

Copyright © 2023, Brooks Automation, Inc.

Request Position Station

Requests the current position of the pan in station coordinates. If the robot has been commanded by a station-oriented command such as pick, place, or GoTo, the position for the specified axis for the specified arm is reported.

To position the arm in Station Coordinates, use the GoTo command with the station number specified.

Request Position Station		
Command Syntax	`RQ POS STN <<(ARM)>[(A|B)]> [ALL| {(R|T|Z|W|WA|WB|SLOT)}]` *The station number is not entered in the request command. The current station number is reported. ARM is not supported in VT5 Compatibility.*	
Response Syntax	`POS STN arm rStr station slot zStr wStr tStr pathStr pathPntStr`	
Arguments	***NOTE:** If a value cannot be determined by the robot, it will be reported as "_ _ _ _ " in the response.*	
	ALL:	Specifies all positions in the order presented in the Response format.
	arm:	Arm descriptor
	R:	`rStr`: Reports radial position as EXTENDED or RETRACTED. If the arm is at the Home position, the response = HM.
	station:	Station Number response. Response field size: 2 Reports the current station number being addressed. The return value will be 0 if no station is addressed.
	SLOT:	`slot`: Response field size: 4 The current slot number of the arm. The return value will be 1 for a station with no slots specified, 0 if the robot is not at a station.
	Z:	`ZStr`: Reports the Z-axis position as UP or DOWN. If the arm is at the Home position, the response = HM.
	T:	Response field size: 6 `TStr`: Reports the Theta axis position. If the arm is at the Home position, the response = HM.
	PATH:	*This command is used with the 4-Axis and 5-Axis Armsets.* Response field size: ___ `pathStr:`
	PATH POINT:	*This command is used with the 4-Axis and 5-Axis Armsets.* Response field size: ___ `pathPntStr:`

Copyright © 2023, Brooks Automation, Inc.

Request Position Station	
Description	Requests the current position of the pan in station coordinates.

Example	`RQ POS STN R T Z NOW`
	`POS STN EX 01 DN`
	
	`RQ POS STN ALL`
	`POS STN EX 01 0001 DN`
	
	Request position station in station coordinates:
	`RQ POS STN 3 ARM A ALL`

Set/Request/Store Behavior

Sets, requests, and stores the behavior of the robot during GoTo, pick, and place commands.

The following section defines the behavior of the GOTO command using MAG 7 and WAVE II behavior.

Table 9-1: GOTO N 1 R EX Z UP

MAG 7 BEHAVIOR	WAVE II BEHAVIOR
If robot is not at station 1: • Robot retracts. • Robot moves to T position and Z UP position. • Robot extends. If robot is at station 1 and retracted in Z DN: • Robot moves to Z UP position. • Robot extends. If robot is at station 1 and retracted in Z UP: • Robot extends. If robot is at station 1 and extended in Z DN: • Robot moves to Z UP position. If robot is at station 1 and extended in Z UP: • No action occurs.	• Robot retracts. • Robot rotates to the specified station (if necessary) and moves into Z UP position. • Robot extends.

Copyright © 2023, Brooks Automation, Inc.

Table 9-2: GOTO N 1 R EX Z DN

MAG 7 BEHAVIOR	WAVE II BEHAVIOR
If robot is not at station 1: • Robot retracts. • Robot moves to T position and Z DN position. • Robot extends. If robot is at station 1 and retracted in Z DN: • Robot extends. If robot is at station 1 and retracted in Z UP: • Robot moves to Z DN position. • Robot extends. If robot is at station 1 and extended in Z UP: • Robot moves to Z DN position. If robot is at station 1 and extended in Z DN: • No action occurs.	• Robot retracts. • Robot rotates to the specified station (if necessary) and moves into Z DN position. • Robot extends to the station R with R offsets position.

Table 9-3: Robot Behavior Response for MAG7 and MAG7 with an Offset

	MAG 7 BEHAVIOR Mode	(MAG7_OFFSET_AT STN Behavior Mode
GOTO N 1 Z UP R EX RQ POS STN ALL	Z-UP Extend POS STN EX 01 0001 UP	Z-UP Extend POS STN EX 01 0001 UP TS
GOTO N 1 Z UP R EX TO 10 RQ POS STN ALL	Move Theta offset POS STN……	Move Theta offset POS STN EX 01 0001 UP TO
GOTO N 1 Z DN R EX TO 10 RQ POS STN ALL	Retract Z-Dn Move Theta offset Extend POS STN -- -- -- --	Z-Dn POS STN EX 01 0001 DN TO

Copyright © 2023, Brooks Automation, Inc.

The following section defines the behavior of the PICK and PLACE commands when the system contains a Radial Motion Sensor (R_MT). The following settings are assumed:

```
INTLCK Wafer Sensor ---------- Y
STN 2 OPTION WAF_SEN R_MT ----- STN02ASENSOR

STN02ASENSOR
TYPE ------------------ R_MT
ACT ------------------- LO
SEN ------------------- 2
STATE ----------------- ON
```

Table 9-4: IF Load is ON and the Command is PICK 2

MAG 7 BEHAVIOR	WAVE II BEHAVIOR
Robot picks from station 2.	_ERR = Possible material on arm.

Table 9-5: IF Load is OFF and the Command is PLACE 2

MAG 7 BEHAVIOR	WAVE II BEHAVIOR
Robot places to station 2.	_ERR = Error prior PLACE: No wafer sensed.

Set Behavior		
Command Syntax	`SET BEHAVIOR (MAG7|MAG7_OFFSET_AT_STN|WAVEII)`	
Response Syntax	There is no response to this command.	
Arguments	BEHAVIOR:	Behavior mode `MAG 7`: Robot behaves as a MagnaTran 7. `MAG7_OFFSET_AT_STN`: Customized behavior used when the robot is extended with an offset position. Refer to "Robot Behavior Response for MAG7 and MAG7 with an Offset". `WAVE II`: Robot behaves as a WAVE II. `LEAP`: Robot behaves as a LEAP robot. Default = LEAP
Description	Sets the behavior of the robot during GoTo, pick, and place commands.	
Example	`SET BEHAVIOR WAVE II`	

Copyright © 2023, Brooks Automation, Inc.

Request Behavior		
Command Syntax	`RQ BEHAVIOR`	
Response Syntax	`BEHAVIOR (MAG 7|MAG7_OFFSET_AT_STN|WAVE II|LEAP)`	
Arguments	BEHAVIOR:	Behavior mode `MAG 7`: Robot behaves as a MagnaTran 7. `MAG7_OFFSET_AT_STN`: Customized behavior used when the robot is extended with an offset position. Refer to "Robot Behavior Response for MAG7 and MAG7 with an Offset". `WAVE II`: Robot behaves as a WAVE II. `LEAP`: Robot behaves as LEAP robot. Default = LEAP
Description	Requests the behavior of the robot during GoTo, pick, and place commands.	
Example	`RQ BEHAVIOR` `BEHAVIOR WAVE II`	

	Store Behavior
Command Syntax	`STORE BEHAVIOR`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores the behavior of the robot during GoTo, pick, and place commands.
Example	`STORE BEHAVIOR`

Copyright © 2023, Brooks Automation, Inc.

Swap

Picks from designated arm and places using other arm.

The swap command causes the following action:
Performs pick operation with the specified or default arm and slot at the specified station. After the wafer has been picked, performs place operation with the other arm at the same slot and station.

There is not a check for wafer presence on the place arm. If this command is executed and a wafer is not present on the place arm, the place action occurs, but no wafer is placed after the pick. The robot reports high level pick and place errors.

Swap		
Command Syntax	`SWAP station <<(ARM)>[(A|B)] <PAN L|R> [SLOT slot] [ALTWAFSPD (Y|N)]`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	slot:	Slot number The slot number is optional. Default = 1
	ARM:	Arm descriptor The Arm used to perform the PICK. Default = Arm A *Using "ARM" is optional.*
	PAN:	*This option is used with dual pan armsets only.* End effector `L`: Left Pan `R`: Right Pan Default = both pans
	ALTWAFSPD	Allows the user to select an alternate speed setting for vacuum robots that have a dual Wafer Speed configuration. `Y` = Use alternate speed `N` = Do not use alternate speed Default = `N`
Supported Devices	This command is used with all dual arm robots.	
Description	Picks from designated arm and places using other arm.	
Example	The following command picks the wafer at Station 1 from Slot 1 using ARM A and places the wafer currently on the end effector of ARM B Station 1 at Slot 1: `SWAP 1`	

Copyright © 2023, Brooks Automation, Inc.

Transfer (XFER)

Transfers a wafer from one specified station to another specified station.

For a normal transfer, this function picks the wafer from one station, retracts, rotates, and places it to another station. With the offset feature, this function picks the wafer from one station and places it to another station with offset values for R and T.

Transfer	
Command Syntax	XFER <<(ARM)>[(A|B|AB)]> <PAN L|R> pickStnNum [SLOT pkSlotNum] <CW|CCW> placeStnNum [SLOT plSlotNum] <CW|CCW> [ALTWAFSPD (Y|N)] [PKRO r_*offset*] [PKTO t_*offset*] [PLRO r_*offset*] [PLTO t_*offset*] [EX NO_VIA|RE NO_VIA] [STRT (R1|N2|R2|VIA)] [ENRT (R1|NR|R2|VIA)] *In VT5 Compatibility, the word ARM is required.*
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Transfer		
Arguments	ARM:	Arm descriptor
	PAN:	*This option is used with dual pan armsets only*. End effector `L`: Left Pan `R`: Right Pan Default = both pans
	pickStnNum:	Station number for pick operation
	placeStnNum:	Station number for place operation
	pkSlotNum:	Slot number for pick operation
	plSlotNum:	Slot number for place operation
	ALTWAFSPD:	Allows the user to select an alternate speed setting for vacuum robots that have a dual Wafer Speed configuration. `Y` = Use alternate speed. `N` = Do not use alternate speed. Default = N
	PKRO:	`r_offset`: Pick radial offset specifies the positive or negative offset from the extend location for that station.
	PLRO:	`r_offset`: Place radial offset specifies the positive or negative offset from the extend location for that station.
	PKTO:	`t_offset`: Pick theta offset specifies the positive or negative offset from the theta location for that station.
	PLTO:	`t_offset`: Place theta offset specifies the positive or negative offset from the theta location for that station.
	EX|RE NO_VIA:	`EX NO_VIA`: During the place command, extend without using the VIA point. `RE NO_VIA`: During the place command, retract without using the VIA point.
	STRT:	Start retract location. This is used only if the station specified is the current station and the robot is at that position. This would be due to a previous use of an ENRT option or a GOTO command. `NR`: No retract `R1`: Normal retract (default) `R2`: Second retract location `VIA`: Performs the PICK starting from the VIA location. The R2 value is set using SET RTRCT2. See "Retract 2 Value (Set/Request/Store)".
Description	Transfers a wafer from one specified station to another specified station.	

Copyright © 2023, Brooks Automation, Inc.

Transfer	

Example	The robot retracts the arm, rotates to station #1, extends the arm, raises the arm (picking up the wafer), retracts the arm, rotates to station #6, extends the arm, lowers the arm (placing the wafer), and retracts the arm.
	
	`XFER 1 6`
	
	In the following example, the pick station and the place station are the same. When the `ENRT NR` option is specified, the last retract is removed.
	`XFER 1 1 ENRT NR PLTO xxx PLRO xxx`
	
	In the following example, the dual pan robot Picks from the left slot of station 4 and places to the left slot of station 3 using Arm B.
	
	`XFER B PAN L 4 3`
	
	In the following example, the robot picks material from station 1, rotating in the counter clock-wise direction, and delivers the material to station 2 by rotating in the clockwise direction.
	
	`XFER 1 CCW 2 CW`

VIA-to-VIA Moves (VTV)

Starts at the current station's via point instead of going home, goes to the pick station's via point, and then goes to the pick station.

Request/Store/Set Speed Scale

The speed scale features allows the user to slow down the robot motion by a factor of between 20-100%, and is expressed by a 0.2-1.0. This is often used to address wafer slippage.

Request Speed		
Command Syntax	`RQ SPEED (ALL|MODE|SCALE WAF` *`axis`*`)`	
Response Syntax	There is no response to this command.	
Arguments	ALL:	Requests speed for all axes.
	MODE:	Requests speed for a specific mode. (ON|OFF)
	SCALE WAF *axis*:	Requests scaling for a specific axis. (R|T|Z)
Description	Allows the user to read the value that has been set.	

Copyright © 2023, Brooks Automation, Inc.

Request Speed	
Example	`SPEED MODE OFF SCALE WAF R 100 WAF T 100 WAF Z 100`

Set Speed Scale		
Command Syntax	`SET SPEED SCALE (WAF AXIS VALUE|CLEAR)`	
Response Syntax	There is no response to this command.	
Arguments	WAF AXIS VALUE:	Scale factor Values are 0.2-1.0.
	CLEAR:	Scale factor removed
Description	Allows the user to arbitrarily slow down robot motion to address wafer slippage.	
Example	`SET SPEED SCALE 0.3`	

Store Speed Scale	
Command Syntax	`STORE SPEED SCALE`
Response Syntax	There is no response to this command.
Arguments	None
Description	Makes the slowdown factor persistent over a power cycle.
Example	`STORE SPEED SCALE`

Copyright © 2023, Brooks Automation, Inc.

10. Wafer Presence Sensors

Check Load

Checks for wafer presence.

For Radial Motion Sensors, the Check Load command first tries to find stations for arm A, or the specified arm, that have wafer sensors mapped. If a station was specified, it will only check that station. The robot will move the arm to the nearest Radial Motion Sensor and updates the load status. The INTLCK option is used to either turn off interlock checking completely (ALL) or allows you to turn off checking of the interlocks individually. If the robot cannot find any stations with a wafer sensor mapped, the command will fail and report an error stating that no station with R_MT sensors could be found. Otherwise, if no R_MT wafer sensor equipped stations succeeded due to sensor errors, the reason the last station was rejected is reported.

Use the Request Load command to request the results of the query .

Check Load	
Command Syntax	`CHECK LOAD [station] [(A|B)] <PAN L|R> [[INTLCK ALL| (DIS|ENB)]| [EX_ENABLE (DIS|ENB)] |[SBIT_SLVL_SEN (DIS|ENB)] |[VLV_SEN (DIS|ENB)]`
Response Syntax	There is no direct response to the Check Load command. Use the request load command to request the results of the query.

Copyright © 2023, Brooks Automation, Inc.

Check Load		
Arguments	station:	Station number
	PAN:	This option is used with dual pan armsets only. End effector `L`: Left Pan `R`: Right Pan Default = both pans
	INTLCK ALL:	All Interlocks `ENB`: Turns all interlocks on. `DIS`: Ignores all interlocks.
	EX_ENABLE:	Process Module sensor `ENB`: Turns the interlock on. `DIS`: Ignores the interlock.
	SBIT_SLVL_SEN:	Slot Valve sensor `ENB`: Turns the interlock on. `DIS`: Ignores the interlock.
	VLV_SEN:	InCooler sensor `ENB`: Turns the interlock on. `DIS`: Ignores the interlock.
Description	Checks for wafer presence.	

Example	The following command checks the load at station 1 and ignores interlocks:
	
	`CHECK LOAD 1 INTLCK ALL DIS`
	
	The following command checks the load at the first valid station and ignores the extend enable interlocks:
	
	`CHECK LOAD INTLCK EX_ENABLE DIS`
	
	The following command checks the load on arm A only at the first valid station and all interlocks are observed:
	
	`CHECK LOAD A`
	
	When the following command is sent to the dual pan, the robot checks the load for the Left Pan on Arm A at station 1. All interlocks are observed. Station 1 R_MT sensor must be set:
	
	`CHECK LOAD 1 A PAN L`
	
	When the following command is sent to the dual pan, the robot checks the load for the both the Left and Right Pans on Arm A at station 1. All interlocks are observed. Station 1 R_MT sensor must be set:
	
	`CHECK LOAD 1 A`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request Interlock Wafer Sensor

Sets and requests the state of specific interlocking capabilities.

Wafer Sensing: The normal operating mode is wafer sensing enabled. Disabling the wafer sensor interlocking will allow the robot to ignore any type of wafer sensors (EXtend, REtract, R_MT, etc.) during action commands. This sensing interlock should be disabled for testing purposes only.

Set Interlock Wafer Sensor		
Command Syntax	`SET (INTER |INTLCK) WAF_SEN (Y|N)`	
Response Syntax	There is no response to this command.	
Arguments	Y:	Enables wafer sensing.
	N:	Disables Wafer Sensing.
Description	Sets the state of specific interlocking capabilities.	
Example	To ignore the wafer sensors during testing: `SET INTLCK WAF_SEN N`	

Request Interlock Wafer Sensor		
Command Syntax	`RQ [(INTER|INTLCK) WAF_SEN`	
Response Syntax	`WAF_SEN (Y|N)`	
Arguments	Y:	Wafer sensing is enabled.
	N:	Wafer sensing is disabled.
Description	Requests the state of specific interlocking capabilities.	
Example	`RQ INTLCK WAF_SEN` `WAF_SEN N`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request Capture

Sets and requests the high speed polling of servo position data triggered by the specified digital input.

The set capture command will enable or disable capturing of servo position data for the specified input such as a sensor. When a SET CPTR ON command is issued, the Servo Position Table is cleared to accept new data. A maximum of ten entries may be made into the Servo Position Table. Attempting to store additional data will cause that data to be lost.

All servo positions are recorded in microns or millidegrees. The state the sensor transitions to is recorded after the servo positions.

> ***NOTE:*** Issuing multiple set capture commands will result in only the last input being active for capture. However, if you need to have all capture to be active, use the command `SET CPTR 0 ON`

The request capture command will list all valid entries, up to the maximum of ten, in the Servo Position Table if there are any entries. If there are no entries, the request capture command will indicate that the capture function has not been triggered.

All servo positions are reported in microns or millidegrees. The state that the sensor transitions to is reported after the servo positions.

The capture trigger may be enabled if any servo is unreferenced; however, if any servos become unreferenced before or during a capture operation, the capture trigger will be disabled.

> ***NOTE:*** This command cannot be stored.

Set Capture	
Command Syntax	`SET CPTR [(OFF|ON)] sensorNum [VAC (OFF|ON)] [MAP (OFF|ON)]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Capture		
Arguments	sensor:	The identifier assigned to the generic digital input for the sensor. Range = 1-22
	CPTR:	ON: Enables multi-input capture. OFF: Uses single input capture.
	sensorNum:	The identifier of the sensor's generic digital inputs
	VAC:	Uses Vacuum Sensor to capture data. ON: Capture is enabled using the Vacuum Sensor. OFF: Capture is disabled.
	MAP:	Uses Mapping Sensor to capture data. ON: Capture is enabled using the Mapping Sensor. OFF: Capture is disabled.
	r_value:	Position of extension
	t_value:	Position of theta axis
	z_value:	Position of up/down axis
	w_value:	For use with Rotational and Articulated Wrists. Orientation of the end effector For WA and WB, displays data for the current arm
	s_value:	Position of the robot on the traverser
	transition:	H= transition from low to high L=transition from high to low
Description	Sets the high speed polling of servo position data triggered by the specified digital input.	
Example:	`SET CPTR 4 ON` `SET CPTR 7 OFF`	

Request Capture	

Command Syntax	`RQ CPTR` *`sensorNum`* `[(VAC|MAP|ALL)]`
	
	`RQ CPTR ALL`

Response Syntax	In MAG6 Compatibility: `CPTR R` *`r_value`* `T` *`t_value`* `Z` *`z_value`* `W` *`w_value`* `S` *`s_value`* `(transition)` In VT5 Compatibility: `CPTR R` *`r_value`* `T` *`t_value`* `Z` *`z_value`* `W` *`w_value`* `S` *`s_value`* `(transition)`

Copyright © 2023, Brooks Automation, Inc.

Request Capture		
Arguments	sensor:	The identifier assigned to the generic digital input for the sensor. Range = 1-22
	CPTR:	`ON`: Enables multi-input capture. `OFF`: Uses single input capture.
	sensorNum:	The identifier of the sensor's generic digital inputs
	r_value:	Position of extension
	t_value:	Position of theta axis
	z_value:	Position of up/down axis
	w_value:	For use with Rotational and Articulated Wrists. Orientation of the end effector. For WA and WB, displays data for the current arm.
	s_value:	Position of the robot on the traverser.
	transition:	H= transition from low to high L=transition from high to low
Description	Requests the high speed polling of servo position data triggered by the specified digital input.	

Example:	`RQ CPTR 1`
	`CPTR R 0036379 T 0090232 Z 1234 H`
	`CPTR R 0036458 T 0090325 Z 1256 L`
	`CPTR R 0036567 T 0090343 Z 1245 H`	

Radial Motion Sensors

Radial Motion Sensors detect wafer presence on the specified arm or pan while the arm is traveling in radial (R-axis) motion. The R_MT sensor verifies the success of any wafer transfer operation. The R_ MT sensors may be used in any application where the sensors are placed outside of the normal robot retract position.

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Radial Motion Sense

Sets, requests, and stores the size of the sensing window for Radial Motion sensors. The Sense Limits, along with the position where the R_MT type sensor is located in the chamber, determine the sensing window. The sensing window must be at least 20mm. Therefore the outer value must be greater than 20mm.

The store command saves the current settings for the Radial Motion detection sensing limits. These values along with the position where the R_MT type sensor is located in the chamber determine the sensing window.

Set Radial Motion Sense		
Command Syntax	`SET R_MT SENSE {LIMITS {OUTER` *`outer`* `|INNER` *`inner`*`} | WAFER SIZE` *`wafer_size`*`}`	
Response Syntax	There is no response to this command.	
Arguments	LIMITS OUTER:	Length of the wafer sensing window in microns.
	LIMITS INNER:	Length from the edge of the wafer to the start of the wafer sensing window in microns.
	wafer_size:	Wafer size in microns: 200000 for 200mm wafers 300000 for 300mm wafers
Description	Sets the size of the sensing window for Radial Motion sensors. The sensing window edge close to robot center will be: Rinner = Rpos + inner_limit. The sensing window edge close to station center will be: Router = Rpos + inner_limit + outer_ limit. The R_MT sensor will only be monitored between the Rinner and Router radial range.	
Example	`SET R_MT SENSE LIMITS INNER 10000 OUTER 20000 WAFER SIZE 300000`	

Request Radial Motion Sense	
Command Syntax	`RQ R_MT SENSE (ALL | {LIMITS {INNER | OUTER} | WAFER SIZE})`
Response Syntax	`R_MT SENSE LIMITS INNER` *`Inner`* `LIMITS OUTER` *`Outer`* `WAFER SIZE` *`wafer_size`*

Copyright © 2023, Brooks Automation, Inc.

Request Radial Motion Sense		
Arguments	Outer:	Length of the wafer sensing window in microns.
	Inner:	Length from the edge of the wafer to the start of the wafer sensing window in microns.
	wafer_size:	Wafer size in microns: 200000 for 200mm wafers 300000 for 300mm wafers
Description	Requests the size of the sensing window for Radial Motion sensors.	
Example	`RQ R_MT SENSE ALL` `R_MT SENSE INNER 10000 OUTER 20000 WAFER SIZE 300000`	

Store Radial Motion Sense	
Command Syntax	`STORE R_MT SENSE (ALL | {LIMITS {INNER | OUTER} | WAFER SIZE})`
Response Syntax	There is no response tothis command.
Description	Stores the size of the sensing window for Radial Motion sensors.
Example	`STORE R_MT SENSE ALL`

Copyright © 2023, Brooks Automation, Inc.

R_MT Sensor Placement

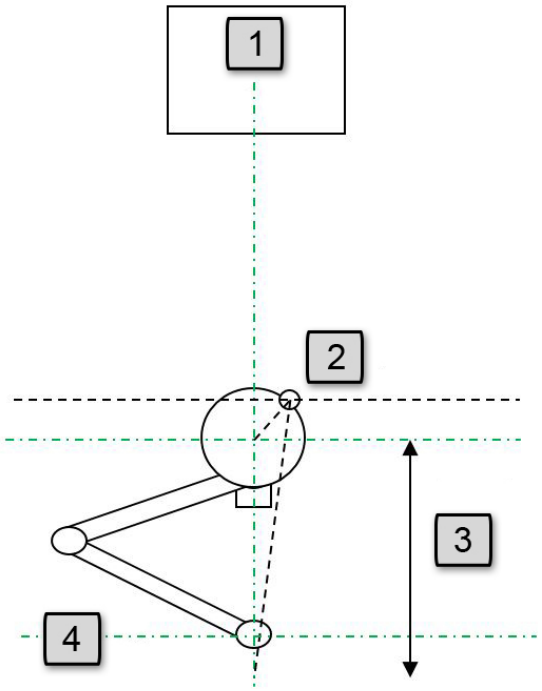


Figure 10-1: R_MT Sensor Placement

Number	Description
1.	Station (R, T)
2.	R_MT Sensor (Rs, Ts)
3.	R, centered wafer hit sensor position
4.	Robot Center

R_MT will typically use one of the sensors being used by AWC.

The maximum possible wafer offset, installation stack-up/tolerances, and robot tracking errors all need to be considered to give some margin between the nominal position of a wafer on a retracted arm and the sensor.. With this criteria, the robot can check the sensor to determine if an operation has failed in the beginning of the motion so the robot motion can stop as quickly as possible.

Copyright © 2023, Brooks Automation, Inc.

R_MT with the Dual Pan Arm

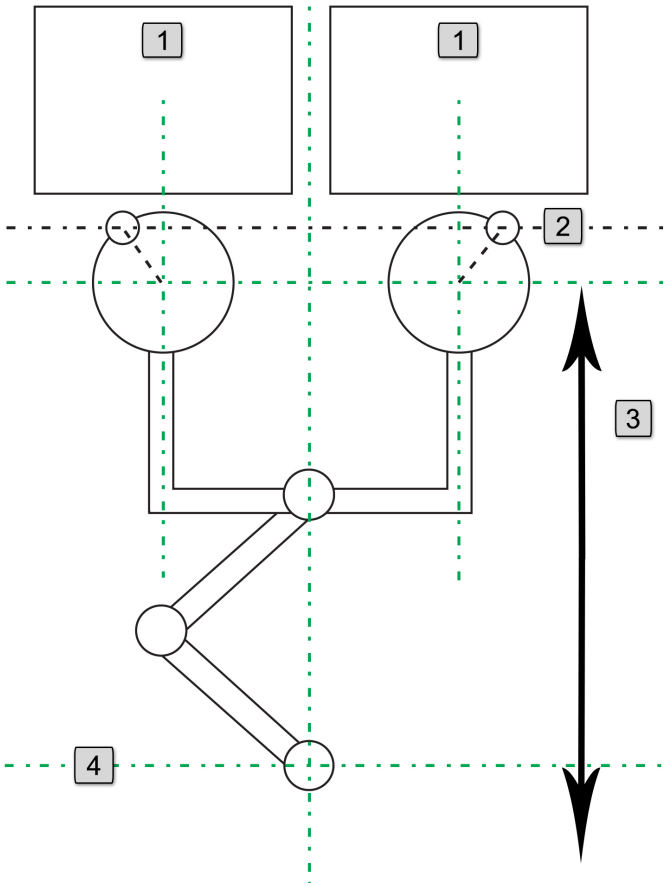


Figure 10-2: Dual Pan Sensor Placement

Number	Description
1.	Station (R, T)
2.	R_MT Sensor (Rs, Ts)
3.	R, centered wafer hit sensor position
4.	Robot Center

The dual pan robot can carry two wafers on each arm, one on each pan (left and right). For this robot, R_MT sensors can be configured for each of the pans.

Copyright © 2023, Brooks Automation, Inc.

Radial Motion Setup Procedure

Before starting the Radial Motion Setup Procedure, ensure you teach each station using serial commands, for serial command.

Step	Action
1.	Physically connect the wafer sensors for each station to the MISC I/O connection. See the section on *Interlocks* in the *Safety* chapter of the *MagnaTran LEAP User Manual*.
2.	Assign a Radial Motion Station Sensor for each arm to each station using the system specific CAD drawings for the R_MT positions. For more information on R_MT Theta, please see the *MagnaTran LEAP User Manual*. ***NOTE:*** *For R_MT Theta, you need to know the location of 0 and then define Theta clockwise.* See "Set/Request/Store Radial Motion Sense" on page 205 for the serial command. Store the R_MT sensor parameters.
3.	**Optional** - Set the inner and outer limits of the Radial Motion Sensor (R_MT) sensing parameters: See "Set/Request/Store Radial Motion Sense" on page 205. Default Inner limit: 10000 Default Outer limit: 20000 Set the size of the wafer. See "Set/Request/Store Radial Motion Sense" on page 205. ***NOTE:*** *This may need to be changed when there is end effector interference, when the installation tolerances aren't optimal, etc.*
4.	Once the R_MT sensor has been defined, verify that it is functioning correctly. This can be done by blocking and unblocking the sensor and requesting its state. `RQ STNSENSOR` *station* `ARM` *arm* `STATE`.
5.	Set the speed to CDM. Perform a CHECK LOAD to initialize the arm load status. The CHECKLOAD uses the sensors to inform the user if there is a load (wafer) present on the end effector or if there is no load present. The CHECKLOAD also checks to see if the positions are setup correctly. See "Check Load" on page 199. Restore the speeds back to normal speeds once the CHECK LOAD has finished.

Copyright © 2023, Brooks Automation, Inc.

Step	Action
6.	To test R_MT: Test the normal, successful cases. Test the error cases. Normal Cases: • Perform a PLACE with a wafer on the arm. • Perform a PICK with a wafer in the station. Error Cases (errors expected): • Perform a PICK with a wafer on the arm (and no wafer in the station). • Remove the wafer from the arm, perform a PICK with no wafer in the station. • Perform a PLACE with no wafer on the arm (or in the station). • Perform a PLACE with a wafer on the arm, that doesn't get dropped off at the station (can assign LOWER to 0 to do this). "Pick" on page 180 "Place" on page 185

R_MT Wafer Sensing Commands

The R_MT type wafer presence sensors work with the PICK, PLACE, and XFER commands, and the GOTO command with the MAT option. During these operations, all motions follow the speed and acceleration profiles according to the arm load status.

NOTE: Wafer sensors are not used during MOVE commands.

Copyright © 2023, Brooks Automation, Inc.

R_MT with PICK

During a PICK operation, the R_MT wafer sensor is active during the extend and the retract at the specified station.

- Robot arm rotates and moves in Z to the station's retract-down position.
- Robot arm extends into the station.
- While the robot arm is extending, the R_MT sensors are used to validate that no wafer is present.
- If a wafer is detected while extending, the robot will halt motion and report an error.
- After completing the extend into the station, the robot will move down up in Z to pick up the wafer.
- After completing the Z motion, the robot arm will retract.
- While the robot arm is retracting, the R_MT sensors are used to validate that a wafer is now present.
- If a wafer is not detected while retracting, the robot will complete the retract motion and then report an error.

Table 10-1: PICK

Motion Segment	Expected State	If a wafer is detected,	If a wafer is NOT detected,
Arm extending	Material is not present	Halt robot and report error, Set LOAD ON	OK
Arm retracting	Material is present	OK	Report error after motion completes

Copyright © 2023, Brooks Automation, Inc.

R_MT with PLACE

During a PLACE operation, the R_MT wafer sensor is active during the extend and the retract at the specified station.

- The robot arm first retracts in the R direction if required.
- The robot arm moves in the T and Z directions to the specified station.
- The robot monitors the R_MT wafer sensor to determine if a wafer is on the end effector.
- If the robot detects a wafer during the extend motion, the PLACE proceeds.
- After the robot extends into the station, performs the PLACE operation, and the arm retracts, the PLACE operation completes.
- During the retract motion, the robot monitors the R_MT sensor to determine if a wafer is not present and the PLACE operation was successful.
- .If the robot detects a wafer on the retract motion, an error is generated and the PLACE fails.

Table 10-2: PLACE

Motion Segment	Expected State	If a wafer is detected,	If a wafer is NOT detected,
Arm extending	Material is present	OK	Halt robot and report error
Arm retracting	Material is not present	Report error after motion completes	OK

Copyright © 2023, Brooks Automation, Inc.

R_MT with GOTO

R_MT wafer sensing is available with the GOTO commands only when using the MAT option.The MAT option is required when the R_MT sensors are setup and an error is reported if the MAT option is not supplied. When the MAT option is specified during an Extend or Retract motion, the robot monitors the R_MT wafer sensor to determine if the load matches that of the MAT option. If the load status and the expected load status do not match, an error will be generated. See Table 10-3 for the possible scenarios of the GOTO command with MAT option.

Table 10-3: GOTO with MAT Option Scenarios

Command	Expected State	If a wafer is detected,	If a wafer is NOT detected,
GOTO R EX MAT OFF	MAT OFF	Halt robot and report error	OK
GOTO R EX MAT ON	MAT ON	OK	Halt robot and report error
GOTO R RE MAT OFF	MAT OFF	Report error after completed	OK
GOTO R RE MAT ON	MAT ON	OK	Report error after completed

Copyright © 2023, Brooks Automation, Inc.

11. Active Wafer Centering

Active Wafer Centering (AWC) is a feature that corrects for minor slippage of the wafer on the robot arm. AWC attempts to correct the slippage to the original placement within user-defined limits.

Global Settings

Global settings are general settings for the AWC feature.

Set/Request/Store Mode

Sets, requests, and stores the AWC mode for the system to Right Left Sensor mode. The AWC Mode command sets the mode for the system. AWC mode setting should be the first command issued setting up the AWC feature and must be done before any other AWC configuration. Mode changes without reconfiguration and recalibration of the AWC are not recommended and may lead to unpredictable results. Each sensor will be set up individually.

Set Mode	
Command Syntax	`SET WAF_CEN MODE RL_SENS`
Response Syntax	There is no response to this command.
Arguments	`RL_SENS`: right and left sensor
Description	Sets the AWC mode for the system to Right Left Sensor mode.
Example	`SET WAF_CEN MODE RL_SENS`

Request Mode	
Command Syntax	`RQ WAF_CEN MODE`

Copyright © 2023, Brooks Automation, Inc.

Request Mode	
Response Syntax	`WAF_CEN RL_SENS`
Arguments	`RL_SENS`: right and left sensors
Description	Requests the AWC mode for the system to Right Left Sensor mode.
Example	`RQ WAF_CEN MODE` `WAF_CEN RL_SENS`

Store Mode	
Command Syntax	`STORE WAF_CEN MODE`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores the AWC mode for the system to Right Left Sensor mode.
Example	`STORE WAF_CEN MODE`

AWC Individual Settings

AWC Individual Settings can be set up on any station selected by the user.

Request/Store All Sensor Settings

Requests and stores all wafer centering parameters for all sensors in the specified station. This command requests and stores all AWC sensor parameters for the specified station for each of the sensors specified by the MODE command. Format is the same for SINGLE_SEN as well as for RL_SENS.

Commands stored are:

```
LEFT_SENS (io)
LEFT_SENS R T
LEFT_SENS DEL_EX LD TRL DEL_RE LD TRL
RIGHT_SENS (io)
RIGHT_SENS R T
RIGHT_SENS DEL_EX LD TRL DEL_RE LD TRL
```

See also STORE STN WAF_CEN RIGHT SENS ALL, STORE STN WAF_CEN LEFT SENS ALL, and STORE STN WAF_CEN SETTING ALL.

Copyright © 2023, Brooks Automation, Inc.

Request All Sensor Parameters for All Sensors in Station		
Command Syntax	`RQ STN station [[ARM] (A|B) [PAN (L|R)] WAF_CEN SENS ALL`	
Response Syntax	`STN station arm WAF_CEN [PAN (L|R)] LEFT_SENS pos_r pos_t del_ex_ld del_ex_trl del_re_ld del_rl_trl RIGHT_SENS pos_r pos_t del_ex_ld del_ ex_trl del_re_ld del_rl_trl`	
Arguments	station:	Station number for the command
	ARM:	Arm descriptor
	PAN:	Left or right pan, if applicable
	pos_r:	Radial position
	pos_t:	Theta position
	del_ex_ld:	Leading edge delay extend value
	del_ex_trl:	Trailing edge delay extend value
	del_re_ld:	Leading edge delay retract value
	del_re_trl:	Trailing edge delay retract value
Description	Requests all wafer centering parameters for all sensors in the specified station.	

Example	Request the AWC settings for station 1, arm B:
	
	`RQ STN 1 ARM B WAF_CEN SENS ALL`
	`STN 1 B WAF_CEN LEFT_SENS 02 497120 001823 000356 000784 000517 001011 RIGHT_SENS 05 505634 002118 000409 000809 000498 000961`	

Store All Sensor Parameters for All Sensors in Station		
Command Syntax	`STORE STN station [[ARM] (A|B)] [PAN (L|R)] WAF_CEN SENS ALL`	
Response Syntax	There is no response to this command.	
Arguments	station:	Specifies the station number for the command.
	ARM:	Arm descriptor
Description	Stores all wafer centering parameters for all sensors in the specified station.	
Example	Store all station 1, arm B sensor parameters: `STORE STN 1 ARM B WAF_CEN SENS ALL`	

Copyright © 2023, Brooks Automation, Inc.

Set Clear Station Wafer Centering Sensor Parameters

Sets all parameters of all sensors for the specified station to default values. This command clears position and delay values related to all sensors for the specified station and the specified arm. Wafer centering sensor parameters must be retaught after any station location changes. For that reason, checks were added to verify station or Via point changes. If changes are made and the sensors are used, the system will report an error. To resume operation, use this command to clear the sensor parameters and reteach.

NOTE: This command sets ENABLE to N and sets all wafer centering sensor parameters to their default values, but keeps the sensor IO number and other wafer centering settings.

NOTE: The CLEAR command cannot be requested or stored.

Set Clear Station Wafer Centering Sensor Parameters		
Command Syntax	`SET STN station [[ARM (A|B)] WAF_CEN CLEAR`	
Response Syntax	There is no response to this command.	
Arguments	station:	The station number for the station involved with the command
	ARM:	Arm descriptor
Description	Sets all parameters of all sensors for the specified station to default values.	
Example	Reset all station sensor parameters: `SET STN 1 WAF_CEN CLEAR`	

Copyright © 2023, Brooks Automation, Inc.

Station Settings

Set/Request/Store Enable

Sets, requests, and stores the Enable AWC command at individual stations.

Set Enable		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN ENABLE (Y|N)`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	ENABLE:	**Y**= enables AWC (turn the AWC feature on) **N**= disables AWC (turn the AWC feature off)
Description	Sets the Enable AWC command at individual stations.	
Example	Disable AWC calculation for station 1 even if all other parameters are set and sensors are calibrated: `SET STN 1 WAF_CEN ENABLE N`	

Request Enable		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN ENABLE`	

Response Syntax	In Mag 6 Packet Mode:
	
	`STN station [A|B] WAF_CEN (Y|N)`
	
	In VT5 Packet Mode:
	
	`STN station [A|B] EN (Y|N)`	

Arguments	station:	Station number
	ARM:	Arm descriptor
	ENABLE:	**Y**= AWC enabled (AWC feature is on) **N** = AWC disabled (AWC feature is off)
Description	Requests the Enable AWC command at individual stations.	

Example	Request the enable option for AWC calculation for station 1:
	
	`RQ STN 1 WAF_CEN SETTING ENABLE`
	`STN 1 A WAF_CEN SETTING Y`	

Copyright © 2023, Brooks Automation, Inc.

Store Enable		
Command Syntax	`STORE STN station [[ARM] (A|B)] WAF_CEN ENABLE`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
Description	Stores the Enable AWC command at individual stations.	
Example	Store the enable option for AWC calculation for station 1: `STORE STN 1 WAF_CEN ENABLE`	

Set/Request/Store Wafer Size

Sets, requests, and stores the wafer size for AWC at individual stations. Wafer size tells the system what size wafer to expect to find.

Table 11-1: Typical Wafer Sizes

Wafer Sizes in Millimeters	Wafer Sizes in Microns
200 mm	200000 µm
300 mm	300000 µm
450 mm	450000 µm

Set Wafer Size		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN WAF_SIZE size`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	WAF_SIZE:	Size of wafer in microns (6 digits).
Description	Sets the wafer size for AWC at individual stations.	

Copyright © 2023, Brooks Automation, Inc.

Set Wafer Size	
Example	Set the centering wafer size for station 1 to 200 mm: `SET STN 1 WAF_CEN WAF_SIZE 200000`

Request Wafer Size		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN WAF_SIZE`	
Response Syntax	`STN station [A|B] WAF_CEN waf_size`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	WAF_SIZE:	Size of wafer in microns (6 digits).
Description	Requests the wafer size for AWC at individual stations.	

Example	Request the centering wafer size for station 1:
	`RQ STN 1 WAF_CEN WAF_SIZE`
	`STN 1 A WAF_CEN 200000`	

Store Wafer Size		
Command Syntax	`STORE STN station [[ARM] A|B] WAF_CEN WAF_SIZE`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	WAF_SIZE:	Size of wafer in microns (6 digits).
Description	Stores the wafer size for AWC at individual stations.	
Example	Store the centering wafer size for station 1: `STORE STN 1 WAF_CEN WAF_SIZE`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Maximum Deviation

Sets, requests, and stores the maximum wafer size deviation for individual stations. The maximum deviation command sets the maximum amount of deviation allowed for AWC parameter option for a particular station. Deviation corresponds to the maximum value, in microns, that the substrate center is allowed to deviate from the station center (wafer center and end effector center) during a place, transfer, or swap command. If the WAF_CEN function determines that the wafer deviation has exceeded this number, the place, transfer, or swap command is aborted, the robot returns to the retract position, and an error is reported. This value is also used during the calibration routine. The default maximum deviation value is 6000 microns.

The AWC parameter names are displayed in long and short variations. Short name is preferred. Long name is used for backward compatibility.

Set Maximum Deviation		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN (MAX_DEV|MAX_DEVIATION) max_dev`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	MAX_DEV:	Maximum deviation (in microns)
Description	Sets the maximum wafer size deviation for individual stations.	
Example	Set the maximum wafer offset for PLACE command to 5000 microns (5mm): `SET STN 1 WAF_CEN MAX_DEV 5000`	

Request Maximum Deviation		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN (MAX_DEV|MAX_DEVIATION)`	
Response Syntax	`STN station [A|B] WAF_CEN max_deviation`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	MAX_DEV:	Maximum deviation (in microns)
Description	Requests the maximum wafer size deviation for individual stations.	

Copyright © 2023, Brooks Automation, Inc.

Request Maximum Deviation	

Example	Request the maximum wafer offset for station 1, arm B, for the PLACE command:
	
	`RQ STN 1 ARM B WAF_CEN SETTING MAX_DEV`
	`STN 1 A WAF_CEN SETTING 5000`

Store Maximum Deviation		
Command Syntax	`STORE STN station [[ARM] (A|B)] WAF_CEN (MAX_DEV|MAX_DEVIATION)`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	MAX_DEV:	Maximum deviation (in microns)
Description	Stores the maximum wafer size deviation for individual stations.	
Example	Store the maximum wafer offset for PLACE command: `STORE STN 1 WAF_CEN MAX_DEV`	

Set/Request/Store Check on Pick Enable

Sets, requests, and stores the substrate deviation to be calculated during the retract move after pick. The check on pick enable command will perform a check of the substrate deviation for the wafer that is on the end effector during the retract move after a successful pick operation. If the check on pick enable function determines that the wafer deviation has exceeded the maximum pick deviation value, the pick command is completed and an error is reported.

Since the substrate position in this station is already corrected by AWC, the check on pick enable value is primarily dedicated to check the AWC precision and is also used to ensure wafer is well centered before placing in a station that has tight tolerances. If set to N, the check on pick will not be executed and the data will not be available.

Set Check on Pick Enable	
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN CK_ON_PICK (Y|N)`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Check on Pick Enable		
Arguments	station:	Station number
	ARM:	Arm descriptor
	CK_ON_PICK:	**Y**: check during retract after pick **N**: do not check during retract after PICK
Description	Sets the substrate deviation to be calculated during the retract move after pick.	
Example	Set the parameter to allow for checking the maximum wafer offset for a PICK command: `SET STN 1 WAF_CEN CK_ON_PICK Y`	

Request Check on Pick Enable		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN CK_ON_PICK`	
Response Syntax	`STN station [A|B] WAF_CEN (Y|N)`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	CK_ON_PICK:	**Y**: check during retract after pick **N**: do not check during retract after PICK
Description	Requests the substrate deviation to be calculated during the retract move after pick.	

Example	Request the option for checking the maximum wafer offset for PICK command for station 11:
	`RQ STN 11 WAF_CEN CK_ON_PICK`
	`STN 11 A WAF_CEN Y`	

Store Check on Pick Enable		
Command Syntax	`STORE STN station [[ARM] (A|B)] WAF_CEN CK_ON_PICK`	
Description	Stores the substrate deviation to be calculated during the retract move after pick.	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	CK_ON_PICK:	**Y**: check during retract after pick **N**: do not check during retract after PICK

Copyright © 2023, Brooks Automation, Inc.

Store Check on Pick Enable	
Description	Stores the substrate deviation to be calculated during the retract move after pick.
Example	Store the option for checking the maximum wafer offset for PICK command for station 1: `STORE STN 1 WAF_CEN CK_ON_PICK`

Set/Request/Store Check on Pick Max Pick Deviation

Sets, requests, and stores maximum deviation parameter for check on pick. The maximum pick deviation command sets the maximum amount of deviation allowed for the check on pick option for the designated station. Deviation corresponds to the maximum value, in microns, that the substrate center is allowed to deviate from the end effector center during retract after a successful pick. The default is 3000. If the deviation is greater than maximum deviation, an error will be reported.

Set Check on Pick Max Pick Deviation		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN MAX_PICK_DEV|MAX_PICK_DEVIATION max_pick_dev`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	MAX_PICK_DEV:	Maximum deviation (in microns)
Description	Sets the maximum deviation parameter for check on pick.	
Example	Set the maximum wafer offset for pick command to 5000 microns (5 mm): `SET STN 1 WAF_CEN MAX_PICK_DEV 5000`	

Request Check on Pick Max Pick Deviation	
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN MAX_PICK_DEV|MAX_PICK_DEVIATION`
Response Syntax	`STN station [A|B] max_pick_dev`

Copyright © 2023, Brooks Automation, Inc.

Request Check on Pick Max Pick Deviation		
Arguments	station:	Station number
	ARM:	Arm descriptor
	MAX_PICK_DEV:	Maximum deviation (in microns)
Description	Requests the maximum deviation parameter for check on pick.	
Example	`RQ STN 32 B WAF_CEN MAX_PICK_DEV` `STN 32 B WAF_CEN 5000`	

Store Check on Pick Max Pick Deviation		
Command Syntax	`STORE STN` *`station`* `[[ARM] (A|B)] WAF_CEN MAX_PICK_DEV|MAX_PICK_DEVIATION`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	MAX_PICK_DEV:	Maximum deviation (in microns)
Description	Stores the maximum deviation parameter for check on pick.	
Example	Store the maximum wafer offset for PICK command for station 1: `STORE STN 1 WAF_CEN MAX_PICK_DEV`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Breakage Threshold

Sets, requests, and stores the wafer defect or breakage detection threshold value in microns. If one of the sensed locations for the wafer exceeds the nominal wafer dimension by the Break Threshold value, the wafer is considered to be a broken wafer and the system will report an error. Broken Wafer Threshold should be greater than the Notch Threshold. The default value is 5000 microns, and should be acceptable for most applications. If errors are falsely reported, the value should be increased.

Command format can be long or short variant. Short variant is preferred. Longer names are for compatibility and should only be used if necessary and only with RL_SENS mode.

Set Breakage Threshold		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN BREAK_THR|BREAK_THRESHOLD break_thr`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	break_thr:	Maximum variation allowed before the wafer is classified as broken.
Description	Sets the wafer defect or breakage detection threshold value in microns.	
Example	Set the maximum break threshold to 4000 microns (4 mm): `SET STN 1 WAF_CEN BREAK_THR 4000`	

Request Breakage Threshold		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN BREAK_THR|BREAK_THRESHOLD`	
Response Syntax	`STN station [A|B] WAF_CEN break_thr`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	BREAK_THR:	Maximum variation allowed before the wafer is classified as broken.
Description	Requests the wafer defect or breakage detection threshold value in microns.	

Example	Request the maximum wafer break threshold value for station 8:
	`RQ STN 8 WAF_CEN BREAK_THR`
	`STN 8 A WAF_CEN 3000`	

Copyright © 2023, Brooks Automation, Inc.

Store Breakage Threshold		
Command Syntax	`STORE STN station [[ARM] (A|B)] WAF_CEN BREAK_THR|BREAK_THRESHOLD`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	BREAK_THR:	Maximum variation allowed before the wafer is classified as broken.
Description	Stores the wafer defect or breakage detection threshold value in microns.	
Example	Store the maximum wafer break threshold value for station 2: `STORE STN 2 WAF_CEN BREAK_THR`	

Find/Set/Request/Store Station Notch Threshold

Finds, sets, requests, or stores the automated calibration routine of the threshold value used to detect and tolerate the alignment fiducials when aligning the wafers. The notch threshold value command sets the AWC notch threshold parameter for a particular station; where *value* corresponds to the amount of notch detection used during a wafer centering operation. This parameter depends on the optical sensor locations and is usually set up automatically by a calibration procedure. If this value cannot be set up by automatic calibration, then the correct value needs to be determined and set up by the operator. Default value is 150 microns.

Threshold prevents measurement noise from falsely eliminating a sensor. Small threshold values make the AWC measurement more sensitive to noise. If the threshold is too small, points on the circumference of the substrate may be rejected more frequently than necessary, causing the AWC precision to decrease. If the notch value is too large, the AWC process will not reject readings affected partially by deviations from the ideal circular shape of the substrate and the AWC precision will deteriorate.

> ***NOTE:*** It is not recommended that `CAL_WITH_PICK` is set to Y for this procedure for the following.

This find command performs an automated calibration routine to determine the appropriate threshold to be used for detecting alignment fiducials on the substrate edge. The fiducial must be placed so that it can be found during one of the 8 passes in this operation. During the calibration process, the substrate is placed several times to the station. The procedure finishes with the substrate placed on the end effector.

Copyright © 2023, Brooks Automation, Inc.

Notch Threshold Calibration requires an ideal wafer placement on the end effector and calibrated sensors. Calibration will fail if a physical place with an eccentricity within the `MAX_DEV` value cannot be done. This means that a Z-less robot or station with edge-grip is not supported by this routine.

Find Station Notch Threshold		
Command Syntax	`FIND STN station [[ARM] (A|B)] WAF_CEN NOTCH_THR|NOTCH_THRESHOLD`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	NOTCH_THR:	Maximum size of the notch (in microns)
Description	Finds the automated calibration routine of the threshold value used to detect and tolerate the alignment fiducials when aligning the wafers.	
Example	Find the data for the station 1 AWC sensors to get data for the notch threshold: `FIND STN 1 WAF_CEN NOTCH_THR`	

Set Station Notch Threshold		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN NOTCH_THR|NOTCH_THRESHOLD`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	NOTCH_THR:	Maximum size of the notch (in microns)
Description	Sets the automated calibration routine of the threshold value used to detect and tolerate the alignment fiducials when aligning the wafers.	
Example	Set the maximum wafer notch threshold to 300 microns (0.3 mm): `SET STN 1 WAF_CEN NOTCH_THR 300`	

Request Station Notch Threshold	
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN NOTCH_THR|NOTCH_THRESHOLD`
Response Syntax	`STN station[A|B] WAF_CEN notch_thr`

Copyright © 2023, Brooks Automation, Inc.

Request Station Notch Threshold		
Arguments	station:	Station number
	ARM:	Arm descriptor
	NOTCH_THR:	Maximum size of the notch (in microns)
Description	Requests the automated calibration routine of the threshold value used to detect and tolerate the alignment fiducials when aligning the wafers.	

Example	Request the station 1 maximum wafer notch threshold:
	
	`RQ STN 1 WAF_CEN SETTING NOTCH_THR`
	`STN 1 A WAF_CEN 150`	

Store Station Notch Threshold		
Command Syntax	`STORE STN` *`station`* `[[ARM] (A|B)] WAF_CEN NOTCH_THR|NOTCH_THRESHOLD`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	NOTCH_THR:	Maximum size of the notch (in microns)
Description	Stores the automated calibration routine of the threshold value used to detect and tolerate the alignment fiducials when aligning the wafers.	
Example	Store the maximum wafer notch threshold: `STORE STN 1 WAF_CEN NOTCH_THR`	

Set/Request/Store Open Extension Parameter

Sets, requests, and stores parameters to indicate the hardware configuration of the end effector in relation to the wafer. The open extension parameter command sets binary parameters reflecting the hardware configuration of the AWC station. In most cases, it needs to be set to “Y”.

The Y setting means that the sensors will detect the edges of the wafer/substrate first, and not the end effector, robot, or manipulator arm. See A and B in Figure 11-1.

The N setting means that the sensors will detect the edges of the end effector, robot, or manipulator arm first, and not the wafer/substrate. See C in Figure 11-1.

Copyright © 2023, Brooks Automation, Inc.

Hardware configurations where some or all of the first detected edges do not belong to the substrate during both extend and retract moves are not supported in WAVE II AWC implementation. As an example, refer to a combination of B and C in Figure 11-1.

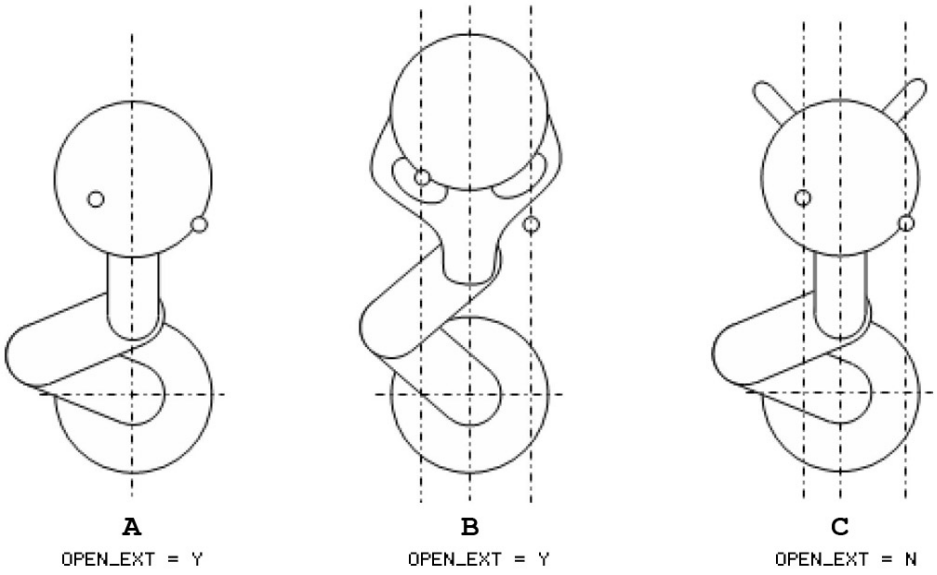


Figure 11-1: AWC Configuration OPEN_EXT (open extension) Parameter

Set Open Extension Parameter		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN OPEN_EXT (Y|N)`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	OPEN_EXT:	Y: Wafer/Substrate edges are detected first. N: End effector, robot, or manipulator arm will be detected first.
Description	Sets the parameters to indicate the hardware configuration of the end effector in relation to the wafer.	
Example	Set the definition of the AWC to use the two first sets of captured data for wafer offset calculation: `SET STN 1 WAF_CEN OPEN_EXT Y`	

Copyright © 2023, Brooks Automation, Inc.

Request Open Extension Parameter		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN OPEN_EXT`	
Response Syntax	`STN station ARM OPEN_EXT (Y|N)`	
Arguments	station:	Station number
	ARM/arm:	Arm descriptor
	OPEN_EXT:	Y: Wafer/Substrate edges are detected first. N: End effector, robot, or manipulator arm will be detected first.
Description	Requests the parameters to indicate the hardware configuration of the end effector in relation to the wafer.	
Example	`RQ STN 8 ARM B WAF_CEN OPEN_EXT` `STN 8 ARM B OPEN_EXT Y`	

Store Open Extension Parameter		
Command Syntax	`STORE STN station [[ARM] (A|B)] WAF_CEN OPEN_EXT`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	OPEN_EXT:	Y: Wafer/Substrate edges are detected first. N: End effector, robot, or manipulator arm will be detected first.
Description	Stores the parameters to indicate the hardware configuration of the end effector in relation to the wafer.	
Example	Store station 1 OPEN_EXT option: `STORE STN 1 WAF_CEN OPEN_EXT`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Station Calibration With Pick Procedure

Sets, requests, and stores a pick option prior to the calibration routine. This parameter allows or disallows the arm to pick a wafer prior to doing a calibration routine using the command FIND WAF POS. See "Find Station Sensor Parameters" on page 247. If a wafer is on the arm pan an error is reported. This command requires that a wafer be manually placed and precisely aligned on the end effector prior to calibration. The default setting is N, the robot will not pick a wafer with either arm pans prior to a calibration procedure.

Set Station Calibration With Pick Procedure		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN CAL_WITH_PICK (Y|N)`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM/arm:	Arm descriptor
	CAL_WITH_PICK:	Y: robot can pick a wafer prior to the calibration routine. N: robot will not pick a wafer with either arm pans prior to a calibration procedure.
Description	Sets a pick option prior to the calibration routine.	

Example	Set the Pick option so the robot is expecting to start the calibration procedure with empty pans but a wafer is present:
	`SET STN 2 WAF_CEN CAL_WITH_PICK Y`
	`FIND WAF POS N 2`
	`ERR = Error prior to PICK: Wafer sensed. error number 740`	

Request Station Calibration With Pick Procedure		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN CAL_WITH_PICK`	
Response Syntax	`STN station [A|B] WAF_CEN (Y|N)`	
Arguments	station:	Station number
	ARM/arm:	Arm descriptor
	CAL_WITH_PICK:	Y: robot can pick a wafer prior to the calibration routine. N: robot will not pick a wafer with either arm pans prior to a calibration procedure.
Description	Requests a pick option prior to the calibration routine.	

Example	Request the Pick option for station 12 arm B AWC calibration:
	`RQ STN 12 B WAF_CEN CAL_WITH_PICK`
	`STN 12 B WAF_CEN N`	

Copyright © 2023, Brooks Automation, Inc.

Store Station Calibration With Pick Procedure		
Command Syntax	`STORE STN station [[ARM] (A|B)] WAF_CEN CAL_WITH_PICK`	
Response Syntax	There is no response tothis command.	
Arguments	station:	Station number
	ARM/arm:	Specifies arm being stored
	CAL_WITH_PICK:	Y: robot can pick a wafer prior to the calibration routine. N: robot will not pick a wafer with either arm pans prior to a calibration procedure.
Description	Stores a pick option prior to the calibration routine.	
Example	Store the Pick option for station 1 AWC calibration: `STORE STN 1 WAF_CEN CAL_WITH_PICK`	

Set/Request/Store Enable Offset Correction Option

Sets, requests, and stores the parameter that enables or disables the physical AWC Offset Correction move during a place. This command enables or disables the AWC Offset Correction moves during a place. AWC calculates the wafer centering data regardless of whether this option is enabled or disabled. The AWC `WAF_CEN_DATA` values are available for the user after the completion of a place command and it is up to the user to decide when to use them. The default value for this parameter is “Y,” which means that the Offset Corrections are performed when the substrate is placed.

When AWC Correction is disabled for a station, the robot does not perform any correction motions during a place, swap, or transfer command, even though the AWC offsets are calculated. When the AWC Correction is enabled for a station, the robot performs the place, swap, or transfer command and applies the wafer centering correction moves.

Set Enable Offset Correction Option	
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN OFFSET_CORR [Y|N]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Enable Offset Correction Option		
Arguments	station:	Station number
	ARM:	Arm descriptor
	Y|N:	Y = Enable offset correction. N= Disable offset correction.
Description	Sets the parameter that enables or disables the physical AWC Offset Correction move during a place.	

Example	Set station AWC Offset Correction to be disabled:
	`SET STN 1 WAF_CEN OFFSET_CORR N`
	Set station AWC Offset Correction to be enabled:
	`SET STN 1 WAF_CEN OFFSET_CORR Y`	

Request Enable Offset Correction Option		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN OFFSET_CORR`	
Response Syntax	`STN station[A|B] WAF_CEN (Y|N)`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	Y|N:	Y = Offset correction is enabled. N= Offset correction is disabled.
Description	Requests the parameter that enables or disables the physical AWC Offset Correction move during a place.	

Example	Request station AWC Offset Correction be displayed:
	`RQ STN 2 WAF_CEN OFFSET_CORR`
	`STN 2 A WAF_CEN Y`	

Store Enable Offset Correction Option	
Command Syntax	`STORE STN station [[ARM] (A|B)] WAF_CEN OFFSET_CORR`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Enable Offset Correction Option		
Arguments	station:	Station number
	ARM:	Arm descriptor
	OFFSET_CORR:	Correction offset (in microns)
Description	Stores the parameter that enables or disables the physical AWC Offset Correction move during a place.	
Example	Store offset correction for station 2 arm B: `STORE STN 2 B WAF_CEN OFFSET_CORR`	

Set/Request/Store Offset Correction Threshold Parameter

Sets, requests, and stores the offset correction threshold value. If the calculated wafer correction offset is less than the correction threshold the wafer location will not be corrected by the AWC correction data when the robot places the wafer at the station. If the calculated wafer correction offset is greater than the threshold, the offset data will be used to correct the wafer placement in the station. Regardless of the correction threshold, the AWC calculates the WAF_CEN DATA and it is available to the user after the completion of the place command. The default correction value is 0.

Set Offset Correction Threshold Parameter		
Command Syntax	`SET STN` *`station`* `[[ARM] (A|B)] WAF_CEN CORR_THR|CORR_THRESHOLD` *`corr_thr`*	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	CORR_THR:	Threshold for offset correction value (in microns)
Description	Sets the offset correction threshold value.	

Copyright © 2023, Brooks Automation, Inc.

Set Offset Correction Threshold Parameter	

Example	Set the AWC offset correction threshold to 100 microns:
	
	This means that if the calculated wafer deviation is less then the threshold, the wafer position will not be corrected. Otherwise, the robot moves to the correct wafer location before placing the wafer into the station.
	
	`SET STN 2 WAF_CEN CORR_THR 100`
	
	Reset the AWC offset correction threshold:
	
	`SET STN 2 WAF_CEN CORR_THR 0`

Request Offset Correction Threshold Parameter		
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN CORR_THR|CORR_THRESHOLD`	
Response Syntax	`STN station [A|B] WAF_CEN corr_thr`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	CORR_THR:	Threshold for offset correction value (in microns)
Description	Requests the offset correction threshold value.	

Example	Request the AWC offset correction threshold for station 7 and arm B:
	
	`RQ STN 7 B WAF_CEN CORR_THR`
	`STN 7 B WAF_CEN 100`	

Store Offset Correction Threshold Parameter		
Command Syntax	`STORE STN station [[ARM] (A|B)] WAF_CEN CORR_THR|CORR_THRESHOLD`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	CORR_THR:	Threshold for offset correction value (in microns)
Description	Stores the offset correction threshold value.	
Example	Store the AWC offset correction threshold of 100 microns for station 7 and arm B: `STORE STN 7 B WAF_CEN CORR_THR`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Enable Steady State Offset

Sets, requests, and stores a known steady state offset. This option allows the user to specify whether or not to apply a steady state offset to the wafer centering data calculated during a place command. When the AWC Steady State Offset option is disabled for a particular station, the robot performs the place, swap, or transfer command to that station using the calculated wafer centering correction offset but not the Steady State Offset.

If the station has values entered for the Steady State Offset X and Y parameters, these values are not used during the place, swap, or transfer command unless the feature is enabled. Enabling this feature adds the Steady State Offset to the calculated AWC wafer centering offset during a place, swap, or transfer if a Steady State Offset is set for the station.

The set command enables or disables the parameter for the Wafer Centering Steady State Offset. Use the store command to save this option setting to memory. The request command returns information about whether the Steady State Offset is enabled or disabled.

Set Enable Steady State Offset		
Command Syntax	`SET STN station [[ARM] (A|B)] WAF_CEN ENABLE_SS_OFFSET [Y|N]`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	ENABLE_S_OFFSET:	Y = Steady state offset will be used for all place commands. N = Steady state offset will not be added to the calculated AWC offset.
Supported Devices	MagnaTran LEAP robots with Right/Left AWC sensors.	
Description	Sets a known steady state offset.	
Example	Set the Steady State Offset to enable: `SET STN 2 WAF_CEN ENABLE_SS_OFFSET Y`	

Request Enable Steady State Offset	
Command Syntax	`RQ STN station [[ARM] (A|B)] WAF_CEN ENABLE_SS_OFFSET`
Response Syntax	`STN station [A|B] WAF_CEN [Y|N]`

Copyright © 2023, Brooks Automation, Inc.

Request Enable Steady State Offset		
Arguments	station:	Station number
	ARM:	Arm descriptor
	ENABLE_S_OFFSET:	Y = Steady state offset is used for all PLACE commands. N = Steady state offset is not being added to the calculated AWC offset.
Supported Devices	MagnaTran LEAP robots with Right/Left AWC sensors.	
Description	Requests a known steady state offset.	
Example	`RQ STN 3 ARM B WAF_CEN ENABLE_SS_OFFSET` `STN 3 B WAF_CEN Y`	

Store Enable Steady State Offset		
Command Syntax	`STORE STN` *`station`* `[[ARM] (A|B)] WAF_CEN ENABLE_SS_OFFSET`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	ENABLE_S_OFFSET:	AWC settings being stored
Supported Devices	MagnaTran LEAP robots with Left/Right AWC sensors.	
Description	Stores a known steady state offset.	
Example	Store the Steady state Offset setting for station 13: `STORE STN 13 WAF_CEN ENABLE_SS_OFFSET`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Steady State Offset Parameters

Sets, requests, and stores steady state offset parameters. Steady State Offset adds the programmed X and Y value to the AWC correction for the specified pan and the specified station during place, swap, and transfer commands.

To use this feature, the user must first enable the option. See " Set/Request/Store Enable Steady State Offset" on page 237. Then, the user must program the Steady State Offset X and Y values to be applied during every place, swap, and transfer into the specified station.

When the Steady State Offset option is enabled and X and Y offsets are programmed for a particular station, the robot calculates the AWC offset during a place, swap, or transfer command and then adds the AWC calculated offsets to the Steady State offset and uses the resulting values to perform the center correction motions for the substrate placement. The results of the AWC offsets, including the steady state offset, can be requested using the `RQ WAF_CEN <PAN L|R> DATA` command. (See "Request Data " on page 248.)

NOTE: `[[PAN] (L|R)]` is only used with the dual pan robot.

Set Steady State Offset Parameters		
Command Syntax	`SET STN station [[ARM] A|B] WAF_CEN [[PAN] L|R] SS_OFFSET [X xVal|Y yVal]`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	PAN L|R:	L = Left pan R = Right pan
	xVal:	The 6 digit value of the Steady State Offset in the X direction in microns for the selected pan.
	yVal:	The 6 digit value of the Steady State Offset in the Y direction in microns for the selected pan. This is reported as a 6 digit value.
Supported Devices	MagnaTran LEAP robots with Right/Left AWC sensors.	
Description	Sets steady state offset parameters.	

Copyright © 2023, Brooks Automation, Inc.

Set Steady State Offset Parameters	

Example	SET commands for standard robot station 1, arm B:
	`SET STN 1 ARM B WAF_CEN SS_OFFSET X 1000 Y -250`
	
	SET commands for dual pan station 1, arm B, pan left and station 1, arm B, pan right:
	`SET STN 1 ARM B WAF_CEN PAN L SS_OFFSET X 1000 Y -250`
	
	`SET STN 1 ARM B WAF_CEN PAN R SS_OFFSET X 500 Y 800`

Request Steady State Offset Parameters		
Command Syntax	`RQ STN` *`station`* `[[ARM] A|B] WAF_CEN [[PAN] L|R] SS_OFFSET`	
Response Syntax	`STN` *`station`* `ARM [A|B] WAF_CEN [[PAN] (L|R)] SS_OFFSET` *`xVal yVal`* `PAN R SS_OFFSET` *`xVal yVal`*	
Arguments	station:	Station number
	ARM:	Arm descriptor
	PAN L|R:	L = Left pan R = Right pan.
	xVal:	The value of the Steady State Offset in the X direction in microns for the selected pan.
	yVal:	The value of the Steady State Offset in the Y direction in microns for the selected pan.
Supported Devices	MagnaTran LEAP robots with Left/Right AWC sensors.	
Description	Requests steady state offset parameters.	

Example	Request for standard robot Steady State Offset parameters, station 1, arm B:
	`RQ STN 1 ARM B WAF_CEN SS_OFFSET`
	`STN 1 B WAF_CEN PAN L SS_OFFSET PAN L SS_OFFSET 001000 -000250`
	
	Request for dual pan Steady State Offset parameters, station 1, arm B, pan L, pan R:
	`RQ STN 1 ARM B WAF_CEN PAN L SS_OFFSET`
	`STN 1 B WAF_CEN PAN L SS_OFFSET PAN L SS_OFFSET 001000 -000250`
	
	`RQ STN 1 ARM B WAF_CEN PAN R SS_OFFSET`
	`STN 1 B WAF_CEN PAN R SS_OFFSET PAN R SS_OFFSET 000500 000800`	

Copyright © 2023, Brooks Automation, Inc.

Store Steady State Offset Parameters		

Command Syntax	`STORE STN station [[ARM] A|B] WAF_CEN SS_OFFSET PAN L SS_OFFSET xVal yVal`
	`PAN R SS_OFFSET xVal yVal`
	
	`STORE STN station [[ARM] A|B] WAF_CEN SS_OFFSET xVal yVal`	

Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	PAN L|R:	L = Left pan R = Right pan
Supported Devices	MagnaTran LEAP robots with Right/Left AWC sensors.	
Description	Stores steady state offset parameters.	
Example	Store command for Steady State Offset parameters for station 6, arm B: `STORE STN 6 ARM B WAF_CEN SS_OFFSET`	

Copyright © 2023, Brooks Automation, Inc.

AWC Sensor Calibration

Request/Store Sensor Parameters for Specified Sensor in Station

Requests and stores all AWC parameters for the specified sensor in a specific station.

See also `STORE STN WAF_CEN SENS ALL` and `STORE STN WAF_CEN SETTING ALL`.

Commands stored for SINGLE_SENS ALL command are:

```
SENS io
SENS sensorNumber R T
SENS DEL_EX LD TRL DEL_RE LD TRL
```

Commands stored for RIGHT_SENS ALL command are:

```
RIGHT_SENS io
RIGHT_SENS sensorNumber R T
RIGHT_SENS DEL_EX LD TRL DEL_RE LD TRL
```

Commands stored for LEFT_SENS ALL command are:

```
LEFT_SENS io
LEFT_SENS sensorNumber R T
LEFT_SENS DEL_EX LD TRL DEL_RE LD TRL
```

NOTE: `[[PAN] L|R)]` is only used with the dual pan.

Request Sensor Parameters for Specified Sensor in Station	

Command Syntax	`RQ STN station [[ARM] (A|B)] [[PAN] (L|R)] WAF_CEN SENS ALL`
	`RQ STN station [[ARM] (A|B)] [[PAN] (L|R)] WAF_CEN RIGHT_SENS ALL`
	`RQ STN station [[ARM] (A|B)] [[PAN] (L|R)] WAF_CEN LEFT_SENS ALL`

Response Syntax	`STN station arm WAF_CEN PAN (L|R) (LEFT_SENS|RIGHT_SENS) pos r pos t del_ex_ld del_ex trl del_re ld del_rl trl`

Copyright © 2023, Brooks Automation, Inc.

Request Sensor Parameters for Specified Sensor in Station		
Arguments	station:	Station number
	ARM:	Arm descriptor
	PAN L|R:	L: Left pan R: Right pan
	RIGHT_SENS:	Requests the parameters for the right sensor.
	LEFT_SENS:	Requests the parameters for the left sensor.
	pos_r:	Radial position
	pos_t:	Theta position
	del_ex_ld:	Leading edge delay extend value
	del_ex_trl:	Trailing edge delay extend value
	del_re_ld:	Leading edge delay retract value
	del_re_trl:	Trailing edge delay retract value
Description	Requests all AWC parameters for the specified sensor in a specific station.	

Example	Request the left sensor station 13 parameters:
	
	`RQ STN 13 WAF_CEN LEFT_SENS ALL`
	`STN 13 A WAF_CEN PAN L LEFT_SENS 04 562490 192451 000871 001024 000915 000756`	

Store Sensor Parameters for Specified Sensor in Station		

Command Syntax	`STORE STN` *`station`* `<<ARM> A|B> <<PAN> L|R> WAF_CEN SENS ALL`
	`STORE STN` *`station`* `<<ARM> A|B> <<PAN> L|R> WAF_CEN RIGHT_SENS ALL`
	`STORE STN` *`station`* `<<ARM> A|B> <<PAN> L|R> WAF_CEN LEFT_SENS ALL`	

Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	PAN L|R:	L: Left pan R: Right pan
	RIGHT_SENS:	Stores the parameters for the right sensor.
	LEFT_SENS:	Stores the parameters for the left sensor.

Copyright © 2023, Brooks Automation, Inc.

Store Sensor Parameters for Specified Sensor in Station	
Description	Stores all AWC parameters for the specified sensor in a specific station.
Example	Store all left sensor station 13 parameters: `STORE STN 13 WAF_CEN LEFT_SENS ALL`

Request/Store All Sensor Settings

Requests and stores all wafer centering parameters for all sensors in the specified station. This command requests and stores all AWC sensor parameters for the specified station for each of the sensors specified by the MODE command. Format is the same for SINGLE_SEN as well as for RL_ SENS.

Commands stored are:

```
LEFT_SENS (io)
LEFT_SENS R T
LEFT_SENS DEL_EX LD TRL DEL_RE LD TRL
RIGHT_SENS (io)
RIGHT_SENS R T
RIGHT_SENS DEL_EX LD TRL DEL_RE LD TRL
```

See also STORE STN WAF_CEN RIGHT SENS ALL, STORE STN WAF_CEN LEFT SENS ALL, and STORE STN WAF_CEN SETTING ALL.

Request All Sensor Parameters for All Sensors in Station	
Command Syntax	`RQ STN station [[ARM] (A|B) [PAN (L|R)] WAF_CEN SENS ALL`
Response Syntax	`STN station arm WAF_CEN [PAN (L|R)] LEFT_SENS pos_r pos_t del_ex_ld del_ex_trl del_re_ld del_rl_trl RIGHT_SENS pos_r pos_t del_ex_ld del_ ex_trl del_re_ld del_rl_trl`

Copyright © 2023, Brooks Automation, Inc.

Request All Sensor Parameters for All Sensors in Station		
Arguments	station:	Station number
	ARM:	Arm descriptor
	PAN:	Specifies the left or right pan, if applicable
	pos_r:	Radial position
	pos_t:	Theta position
	del_ex_ld:	Leading edge delay extend value
	del_ex_trl:	Trailing edge delay extend value
	del_re_ld:	Leading edge delay retract value
	del_re_trl:	Trailing edge delay retract value
Description	Requests all wafer centering parameters for all sensors in the specified station.	

Example	Request the AWC settings for station 1, arm B:
	
	`RQ STN 1 ARM B WAF_CEN SENS ALL`
	`STN 1 B WAF_CEN LEFT_SENS 02 497120 001823 000356 000784 000517 001011 RIGHT_SENS 05 505634 002118 000409 000809 000498 000961`	

Store All Sensor Parameters for All Sensors in Station		
Command Syntax	`STORE STN station [[ARM] (A|B)] [PAN (L|R)] WAF_CEN SENS ALL`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
Description	Stores all wafer centering parameters for all sensors in the specified station.	
Example	Store all station 1, arm B sensor parameter: `STORE STN 1 ARM B WAF_CEN SENS ALL`	

Copyright © 2023, Brooks Automation, Inc.

Find/Request/Store Wafer Centering All

The find command starts the automated calibration of sensor location and notch threshold for the specified station. Refer to "Request/Store All Sensor Settings " on page 244 to request and store the station parameters for this command.

The find command starts automated calibration routines to determine the locations of the sensors and estimate the delays in the system for AWC station. After the location calibration routine successfully finishes, the calibration routine of threshold for detection of alignment fiducials starts. The wafer is placed several times to the station during the threshold calibration routine.

The procedure finishes with the wafer placed on the end effector. Threshold calibration routine expects standard robot/station setup and fails if physical place with an eccentricity (within `MAX_DEV`) cannot be done. Z-less robots are not supported by this routine.

Find Calibrate Wafer Centering All		
Command Syntax	`FIND STN station [[ARM] A|B] WAF_CEN ALL`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
Description	Starts the automated calibration of sensor location and notch threshold for the specified station.	
Example	Calibrate wafer at Station 1: `FIND STN 1 ARM A WAF_CEN ALL`	

Copyright © 2023, Brooks Automation, Inc.

Find Station Sensor Parameters

Finds the location of the sensors and estimates the delays in the system for AWC. This find command performs an automated calibration routine to determine the locations of the sensors and estimate the delays in the system for the specified AWC station and the specified arm. This calibration must not find the wafer fiducials.

When the AUTO_NOTCH setting is set to Y, then, after the sensor location calibration routine has successfully completed, the calibration routine for the threshold for the detection of the fiducial will start.

The Threshold calibration routine expects a standard robot and station setup and fails if a physical place with an eccentricity within MAX_DEV cannot be done. This means that a Z-less robot or station with edge-grip is not supported by this routine.

See "Request/Store All Sensor Settings " on page 244.

Find Station Sensor Parameters		

Command Syntax	`FIND WAF POS N station [[ARM] (A|B)]`
	
	`FIND STN station [[ARM] (A|B)] WAF_CEN SENS`	

Response Syntax	There is no response to this command. To review a response to the command, refer to the RQ STN WAF_CEN SENS ALL command.	
Arguments	station:	Station number
	ARM/arm:	Arm descriptor
Description	Finds the location of the sensors and estimates the delays in the system for AWC.	
Example	Find the calibration for station 1 AWC sensors: `FIND STN 1 WAF_CEN SENS`	

Copyright © 2023, Brooks Automation, Inc.

AWC Operation

Request Data

Requests the current calculated AWC wafer centering offset. This command is used to display the position capture data and the Radial and Theta offsets calculated during the previous WAF_CEN function. Only valid data is displayed. If valid data is not available, an error is reported.

Request Data	
Command Syntax	`RQ WAF_CEN DATA`
Response Syntax	`WAF_CEN RT r_pos_leading_edge t_pos_leading_edge r_pos_trailing_edge t_pos_trailing_edge LFT r_pos_leading_edge t_pos_leading_edge r_pos_trailing_edge t_pos_trailing_edge r_offset t_offset`
Arguments	None
Description	Requests the current calculated AWC wafer centering offset.

Example	Request current calculated wafer offset reports:
	
	`RQ WAF_CEN DATA`
	`WAF_CEN RT 498152 290861 542109 290798 LFT 379105 290802 620109 290901 001562 000145`

Copyright © 2023, Brooks Automation, Inc.

12. Workspaces

The Workspace Overview

The user can define workspaces – as three dimension zones - in which the robot is allowed to move. A motion error is reported if the user attempts to move a robot end effector outside any active workspaces. The purpose is to reject moves whose path is not within any defined and enabled (active) workspaces, to prevent collisions between the robot end effectors, material, and enclosures. Workspaces can be enabled/disabled globally. Workspaces can be enabled by arm/end effector. By enabling workspace mode the user is locking down the permissible robot end effector move paths that are supported.

Once the workspace feature is enabled, one predefined workspace will exist called the HOME workspace. This workspace defines the safe travel area around the robot home position. The user then defines the workspaces around the robot's total movement to each workstation. This workspace may be created by one of two methods: the user may define the desired workspace by manually entering desired values or the workspace may be automatically set around taught workstations.

Figure 12-1 shows active workspaces for each station (1-6) and also shows the HOME workspace as the circle in the middle around the robot.

NOTE: Motion Interlock (MTN_Intlck) is not supported in LEAP.

Copyright © 2023, Brooks Automation, Inc.

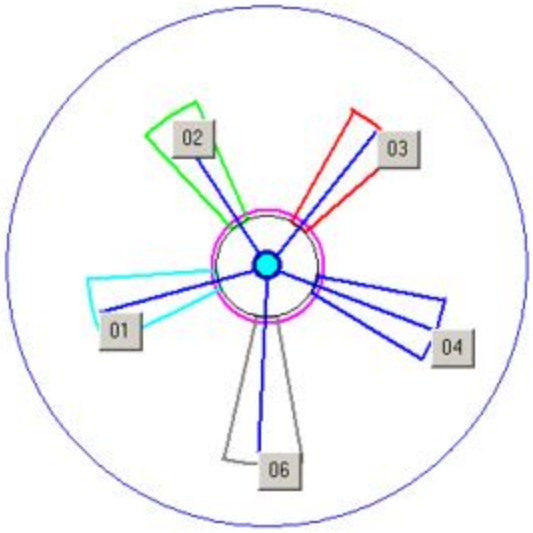


Figure 12-1: Workspace (in 2 dimensions)

A workspace is defined by the following parameters:

Parameter	Description
Name	A maximum 20-character, alphanumeric name unique to the workspace.
State	Specifies whether the workspace is active or inactive.
Arm	Associates an arm to the workspace; may be A, B, or both.
Slot Valve Interlock	An I/O mapped Slot Valve Interlock name to be checked before movement is allowed. Default = NONE.
Station	Indicated which station, if any, is associated with the workspace.
Rmin	The minimum radial axis limit for the workspace.
Tmin	The minimum theta axis limit for the workspace.
Zmin	The minimum Z-axis limit for the workspace.
Rmax	The maximum radial axis limit for the workspace.
Tmax	The maximum theta axis limit for the workspace.
Zmax	The maximum Z-axis limit for the workspace.
W-axis	Specifies the flipping state for the workspace.

Copyright © 2023, Brooks Automation, Inc.

HOME Workspace

The HOME workspace is a predefined volume of space enclosing the robot around the home position. Within this workspace, the robot is limited to an extend length equal to its retract value for the radial axis, a vertical lift height equal to the maximum Z-axis height, and no rotational limit for the theta axis. The HOME workspace is automatically created by the firmware and may be different in volume from robot to robot since robots vary in arm sizes. The HOME workspace can not be changed or deleted. If the HOME workspace is the only Workspace in existence, the robot will only be permitted to home and move about its home volume.

Other Workspaces

Moving from workspace to workspace can be achieved only if the workspaces are overlapping or have a common region.



Figure 12-2: Workspace Definitions

Movement between disjointed workspaces always requires a common region to be defined.

The only exception to this is when stations are being taught from the CDM interface. Since workspaces may not yet be defined for a station because the coordinates for the station are unknown, movement of the robot from the Learn Station menus on the CDM will permit movement into the yet to be defined workspaces.

Copyright © 2023, Brooks Automation, Inc.

Creating Workspaces

Manually Create Workspace

Creating workspaces manually is always available to the user. The manual creation of a workspace requires the use of several commands. The first command is CREATE WSPACE, which requires a unique workspace name as part of the command. Once a workspace has been created, the default values associated with all the remaining parameters are as follows:

Create Workspace Parameters	
STATE	INACTIVE
Interlock	NONE
Arm	NONE
Station	0 (indicating no stations associated with this workspace)
Rmin	0 microns
Tmin	0 millidegrees
Zmin	0 microns
Rmax	0 microns
Tmax	0 millidegrees
Zmax	0 microns
Mtn_Intlck	DIS

NOTE: See "Workspace (Create/Remove)" for the command syntax necessary to create and remove workspaces.

The user can now change the default parameters with the SET WSPACE command. Again the name of the newly created workspace must be a part of the command. This command only updates the active workspace definition. To store the workspace definition permanently, the user must use the STORE WSPACE command, which also requires the workspace name as part of the command. When changing the radial and Z-axis parameters, their respectively minimum values must be equal to or less than their respective maxi- mum values. For the theta axis, the minimum can be greater than the maximum.

Copyright © 2023, Brooks Automation, Inc.

Defining Tmin and Tmax

Tmin can be set to a value less than, equal to, or greater than Tmax. The robot will consider the area starting from Tmin and rotating clockwise to Tmax as the valid theta workspace. By this definition, workspaces starting from Tmax and rotating clockwise to Tmin, or starting from Tmin and rotating counterclockwise to Tmax are outside of the workspace.

Workspace of Modes of Operation

The workspace feature can be configured to operate for various modes.

Workspace Jog Mode – Restricts the motions allowed during CDM Jog to areas inside active workspaces only. This mode is disabled by default, allowing the CDM Jog unrestricted motion.

Workspace Load_Dependent Mode – Specifies when the workspace will be used. Y workspaces are used only when the robot has a load. N workspaces are always used, regardless of the load state.

Copyright © 2023, Brooks Automation, Inc.

Create/Remove Workspace

This is used to create or remove workspaces.

Create Workspace		
Command Syntax	`CREATE WSPACE name`	
Response Syntax	There is no response to this command.	
Arguments	name:	Specifies the alphanumeric name of the new workspace. Must be 20 characters or less. ***NOTE:*** *The Workspace will be set to the default values and stored permanently.*
Description	The CREATE command is used to create a new workspace.	
Example	`CREATE WSPACE test1`	

Remove Workspace		
Command Syntax	`REMOVE WSPACE name`	
Response Syntax	There is no response to this command.	
Arguments	name:	Specifies the alphanumeric name of the workspace. Must be 20 characters or less. ***NOTE:*** *The workspace will be set to the default values and stored permanently.*
Description	The REMOVE command is used to remove all or specific workspaces previously defined.	
Example	`REMOVE WSPACE test1`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Workspace

Sets, requests, and stores workspace parameters.

Set Workspace		
Command Syntax	`STORE WSPACE name [ARM] [STN STATION] [ALL| (STATE |INTLCK |RMIN |RMAX |TMIN |TMAX|ZMIN |ZMAX |MTN_INTLCK]`	
Response Syntax	There is no response to this command.	
Arguments	name:	Specifies the workspace name.
	rmin:	Robot retract value to robot minimum retract value (in microns)
	rmax:	Robot retract value to robot maximum extension value (in microns)
	tmin:	0 - 360000 (millidegrees)
	tmax:	0 - 360000 (millidegrees)
	zmin:	0 to robot minimum Z vertical height (in microns)
	zmax:	0 to robot maximum Z vertical height (in microns)
	STATE:	Current state of the workspace ACTIVE INACTIVE
Description	Sets the specified workspace parameter or parameters for the specified work space name.	
Example	`STORE WSPACE test1 ARM A STN 1 MIN R 60000 MAX R 150000 STATE ACTIVE`	

Copyright © 2023, Brooks Automation, Inc.

Request Workspace		
Command Syntax	RQ WSPACE name [ARM] [STN station] [MIN (R|T |Z))]| [MAX (R|T |Z))] | [ALL| (STATE |INTLCK |RMIN |RMAX |TMIN |TMAX|ZMIN |ZMAX]	
Response Syntax	RQ WSPACE name ALL WSPACE NAME ---------- name STATE ---------- (ACTIVE|INACTIVE) INTLCK ---------- svlv_name ARM(s) ---------- (ARM_A|ARM_B| ARM_BOTH_ STN ---------- station RMIN ---------- rmin RMAX ---------- rmax TMIN ---------- tmin TMAX ---------- tmax ZMIN ---------- zmin ZMAX ---------- zmax	
Arguments	name:	Specifies the work space name.
	rmin:	Robot retract value to robot minimum retract value in microns
	rmax:	Robot retract value to robot maximum extension value in microns
	tmin:	0 - 360000 (millidegrees)
	tmax:	0 - 360000 (millidegrees)
	zmin:	0 to robot minimum Z vertical height in microns
	zmax:	0 to robot maximum Z vertical height in microns
	STATE:	Current state of the Workspace: **ACTIVE** **INACTIVE**
Description	After the workspace area has been established, set, and stored, the Motion Interlock must be enabled for each region to force motions between each region to the overlap region.	
Example	RQ WSPACE ALL HOME_WORK_SPACE ACTIVE NONE BOTH 0 0000000 0378469 000000 360000 -00750 050000 DIS TEST INACTIVE NONE NONE 0 0344063 0344063 000000 000000 000000 000000 DIS	

Copyright © 2023, Brooks Automation, Inc.

Store Workspace		

Command Syntax	`STORE WSPACE name [ARM] [STN station]`
	`[MIN (R|T |Z))]| [MAX (R|T |Z))] |`
	`[ALL| (STATE |INTLCK |RMIN |RMAX |TMIN |TMAX|ZMIN |ZMAX]`	

Response Syntax	There is no response to this command.	
Arguments	name:	Specifies the workspace name.
	rmin:	Robot retract value to robot minimum retract value in microns
	rmax:	Robot retract value to robot maximum extension value in microns
	tmin:	0 - 360000 (millidegrees)
	tmax:	0 - 360000 (millidegrees)
	zmin:	0 to robot minimum Z vertical height in microns
	zmax:	0 to robot maximum Z vertical height in microns
	STATE:	Current state of the Workspace: **ACTIVE** **INACTIVE**
Description	After the workspace area has been established, set, and stored, the Motion Interlock must be enabled for each region to force motions between each region to the overlap region.	
Example	`STORE WSPACE TEST1 B STN 2`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Workspace Mode

Sets, requests, and stores the workspace mode of operation to ON or OFF.

Set Workspace		
Command Syntax	`SET WSPACE MODE [(ON|OFF)] [JOG (ON|OFF)] [LOAD_DEPENDENT (Y|N)]`	
Response Syntax	There is no response to this command.	
Arguments	MODE_DEVOUT: 33:34	Specifies the mode of operation. **ON**: uses workspace for high level moves. **OFF**: Workspaces are ignored.
	JOG:	Specifies the mode of operation. **ON**: uses workspace are honored during Jog. **OFF**: Workspaces are ignored during Jog.
	LOAD_ DEPENDENT:	Specifies when the workspace will be used. **Y**: Workspaces are used ONLY when the robot has a load. **N**: Workspaces are always used, regardless of the load state.
Description	Turns the safe workspace area OFF or ON. The Jog option turns on the workspaces when using the Move->Jog control from the CDM. This helps the user avoid collisions during teaching. If the user enters an area outside of the Workspace while in Mode Jog, turn the Mode Jog to OFF to recover. The default for the Jog option is OFF. The Load Dependent option allows workspace rules to be applied only when the robot has a wafer, or at all times.	
Example	`SET WSPACE MODE ON`	

Copyright © 2023, Brooks Automation, Inc.

Request Workspace		
Command Syntax	`RQ WSPACE MODE [JOG] [LOAD_DEPENDENT]`	
Response Syntax	`WSPACE MODE ON|OFF`	
Arguments	MODE:	Specifies the mode of operation. **ON**: uses workspace for high level moves. **OFF**: Workspaces are ignored.
	JOG:	Specifies the mode of operation. **ON**: uses workspace are honored during Jog. **OFF**: Workspaces are ignored during Jog.
	LOAD_ DEPENDENT:	Specifies when the workspace will be used. **Y**: Workspaces are used ONLY when the robot has a load. **N**: Workspaces are always used, regardless of the load state.
Example	`RQ WSPACE MODE` `WSPACE MODE ON`	

Copyright © 2023, Brooks Automation, Inc.

Store Workspace		
Command Syntax	`STORE WSPACE MODE [JOG] [LOAD_DEPENDENT]`	
Response Syntax	There is no response to this command.	
Arguments	MODE:	Specifies the mode of operation. **ON**: Uses Workspace for high level moves. **OFF**: Workspaces are ignored.
	JOG:	Specifies the mode of operation. **ON**: uses workspace are honored during Jog. **OFF**: Workspaces are ignored during Jog.
	LOAD_ DEPENDENT:	Specifies when the Workspace will be used. **Y**: Workspaces are used ONLY when the robot has a load. **N**: Workspaces are always used, regardless of the load state.
Example	`STORE WSPACE MODE`	

Copyright © 2023, Brooks Automation, Inc.

Workspace Error Codes

Error Code	Definition
Error 1100	Current position not within work space.
Error 1101	Destination position not within work space.
Error 1102	Workspaces do not overlap.
Error 1103	Workspace interlock occurred.
Error 1104	No more workspaces available.
Error 1105	The workspace volume must be specified.
Error 1106	Radial maximum is less than radial minimum.
Error 1107	Theta maximum is less than theta minimum.
Error 1108	Z maximum is less than Z minimum.
Error 1109	Radial minimum is greater than stored radial max.
Error 1110	Radial maximum is less than stored radial min.
Error 1111	Theta minimum is greater than stored theta maximum.
Error 1112	Theta maximum is less than stored theta minimum.
Error 1113	Z minimum is greater than stored Z maximum.
Error 1114	Z maximum is less than stored Z minimum.
Error 1115	Workspace name does not exist.
Error 1116	The arm for the workspace must be specified.
Error 1117	The workspace already exists.
Error 1118	Invalid station number.
Error 1119	Reserved workspace name used.

Copyright © 2023, Brooks Automation, Inc.

13. Atmospheric Robots

There are additional commands and command modifiers, specific to atmospheric robots, that work alongside the other commands in this manual.

Additional Axes

Atmospheric robots support additional axes compared to vacuum robots. Some support a linear track axes that moves the base of the robot horizontally. In many cases, the Z-axis has a much greater range than vacuum robots. Atmospheric robots also have wrists, referred to as the W-axis.

Traverser/S-axis: The traverser axis, also known as the S-axis or track, is used to move the robot between the doors of an EFEM. The traverser axis has a brake mechanism that keeps it from moving when the robot is not powered.

Z-axis Differences: The Z-axis for atmospheric robots has a much greater reach than a typical vacuum robot. The Z-axis can typically reach over the range of 25 wafers in a FOUP carrier.

Wrist/W-axis: The wrist axis, or W-axis, allows the end effector to move at the wrist.

> ***NOTE:*** In the following command tables, non-bold parameters are common to vacuum and atmospheric robots and are explained in Chapter 1: Station Setup. New parameters for atmospheric robots are designated in **bold**.

Set Station	
Command Syntax	`SET STN station <<(ARM)>[(A|B)]> [R rVal] [T tVal] [Z zVal] [LOWER lowDist][Z_WAF zWafVal] [NSLOTS numSlots] [PITCH pitch]` **`[W WA] [S S]`**
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Station		
Arguments	STN:	Station number. Range 1 - 99
	ARM:	Arm descriptor. `A`: Arm A `B`: Arm B Default: A The arm descriptor must be specified to transfer wafers with Arm B. The “ARM” identifier is optional.
	R:	`rVal`: The station’s radial extend location in microns. Response = 7 digits
	T:	`tVal`: The station’s rotational axis location in units of millidegrees. Response = 6 digits
	Z:	`zVal`: The BTO Z-axis of the wafer Transfer Plane relative to Home in microns. Response = 6 digits
	LOWER:	`lowDist`: The distance in microns below the wafer Transfer Plane that becomes the down position location. Response = 6 digits
	NSLOTS:	`numSlots`: The number of slots at the specified station. A value of 0 or 1 indicates a non-cassette type station. Range= 0 - 100 Default = 1 Response = 6 digits ***NOTE:*** *0 and 1 both mean that the station has a single slot.*
	PITCH:	`pitch`: The pitch between slots in microns at the specified station Default = 0 Response = 6 digits
	W:	The wrist's rotational axis location in units of millidegrees.
	S:	The track position relative to Home in microns.
	W UN:	*For rotating wrists only.* `wUNFlpVal`: Specifies unflip position of the end effector Rotated position axis location in units of millidegrees Response = 6 digits
	W FL:	*For rotating wrists only*. `wFlpVal`: Specifies flip position of the end effector Rotated position axis location in units of millidegrees Response = 6 digits

Copyright © 2023, Brooks Automation, Inc.

Set Station		
	WA:	*This command is used with the 5-axis Armset.* `waVal`: Specifies the stations wrist location in millidegrees for the upper end effector. Response = 6 digits
	WB:	*This command is used with the 5-axis Armset.* `wbVal`: Specifies the stations wrist location in millidegrees for the lower end effector. Response = 6 digits
	Z_WAF:	*This command is used with the Z-axis option.* `zWafVal`: The value, in microns, set for the third WF position, wafer position, at the station. The value must be an absolute value between the UP position and the DN position. For example, if the BTO=50000 and LWR=1000, then a valid Z_WAF value is between 49001 and 49999. The 3_Z Station Option for Z position must be set to Yes. Response = 6 digits
Example	`SET STN 5` `R 703858 T 150716 Z 35400 LOWER 5000 NSLOTS 25 PITCH 0` **`W 179275 S 250000`**	

Request Station	
Command Syntax	`RQ STN` *`station`* `[[ARM]|[(A|B)]] [ALL|{R|T|Z|LOWER|Z_WAF|PITCH|NSLOTS|`**`W|S`**`}]`
Response Syntax	`STN` *`station`* `[ARM (A|B)] R` *`R_Value`* `T` *`T_Value`* `Z` *`Z_Value`* `LOWER` *`Lower_Distance`* `NSLOTS` *`Number_Slots`* `PITCH` *`Pitch_Value`* **`W W_Value S S_Value`**

Copyright © 2023, Brooks Automation, Inc.

Request Station		
Arguments	STN:	`station`: Station number Range 1 - 99
	ARM:	Arm descriptor `A`: Arm A `B`: Arm B Default: A The arm descriptor must be specified to transfer wafers with Arm B. The “ARM” identifier is optional.
	R:	Requests the station’s radial extend location in microns. Response = 7 digits
	T:	Requests the station’s rotational axis location in units of mil-lidegrees. Response = 6 digits
	Z:	Requests the BTO Z-axis of the wafer Transfer Plane relative to Home in microns. Response = 6 digits
	LOWER:	Requests the distance in microns below the wafer Transfer Plane that becomes the down position location. Response = 6 digits
	NSLOTS:	Requests the number of slots at the specified station. A value of 0 or 1 indicates a non-cassette type station. Range= 0 - 100 Default = 1 Response = 6 digits
	PITCH:	Requests the pitch between slots in microns at the specified station. Default = 0 Response = 6 digits
	W:	The wrist's rotational axis location in units of millidegrees.
	S:	The track position relative to Home in microns.
	W UN:	*For rotating wrists only.* `wUNFlpVal`: Specifies unflip position of the end effector. Rotated position axis location in units of millidegrees Response = 6 digits
	W FL:	*For rotating wrists only*. `wFlpVal`: Specifies flip position of the end effector. Rotated position axis location in units of millidegrees Response = 6 digits
	WA:	*This command is used with the 5-axis Armset.* `waVal`: Specifies the stations wrist location in millidegrees for the upper end effector. Response = 6 digits

Copyright © 2023, Brooks Automation, Inc.

Request Station		
	WB:	*This command is used with the 5-axis Armset.* `wbVal`: Specifies the stations wrist location in millidegrees for the lower end effector. Response = 6 digits
	Z_WAF:	*This command is used with the Z-axis option.* `zWafVal`: The value, in microns, set for the third WF position, wafer position, at the station. The value must be an absolute value between the UP position and the DN position. For example, if the BTO=50000 and LWR=1000, then a valid Z_WAF value is between 49001 and 49999. The 3_Z Station Option for Z position must be set to Yes. Response = 6 digits
Example	`RQ STN 5` `ALL STN 0005 0703858 150716 035400 005000 10000 000025 179275 250000`	

Store Station	
Command Syntax	`STORE STN` *`station`* `<<(ARM)>[(A|B)]> [ALL|{R|T|Z|LOWER|Z_WAF|PITCH|NSLOTS|`**`W|S`**`}]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Station		
Arguments	STN:	`station`: Station number Range 1 - 99
	ARM:	Arm descriptor `A`: Arm A `B`: Arm B Default: A The arm descriptor must be specified to transfer wafers with Arm B. The “ARM” identifier is optional.
	R:	Stores the radial axis value.
	T:	Stores the theta axis value.
	Z:	Stores the Z-axis value.
	LOWER:	Stores the Lower value.
	NSLOTS:	Stores the number of slots assigned to the station.
	PITCH:	Stores the pitch value.
	W:	The wrist's rotational axis location in units of millidegrees.
	S:	The track position relative to Home in microns.
	Z_WAF:	*This command is used with the Z-axis option.* `zWafVal`: The value, in microns, set for the third WF position, wafer position, at the station. The value must be an absolute value between the UP position and the DN position. For example, if the BTO=50000 and LWR=1000, then a valid Z_WAF value is between 49001 and 49999. The 3_Z Station Option for Z position must be set to Yes. Response = 6 digits
Example	`STORE STN 5` `R 703858 T 150716 Z 35400 LOWER 5000 NSLOTS 25 PITCH 0 W 179275 S 250000`	

Copyright © 2023, Brooks Automation, Inc.

Sensor Types

Reflective Sensor Mapping

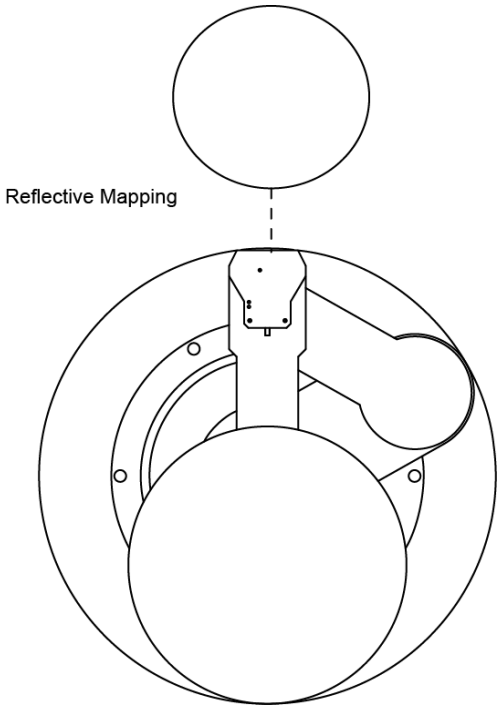


Figure 13-1: Wafer Mapping Sensors - Reflective Mapping

Copyright © 2023, Brooks Automation, Inc.

Through-Beam Mapping

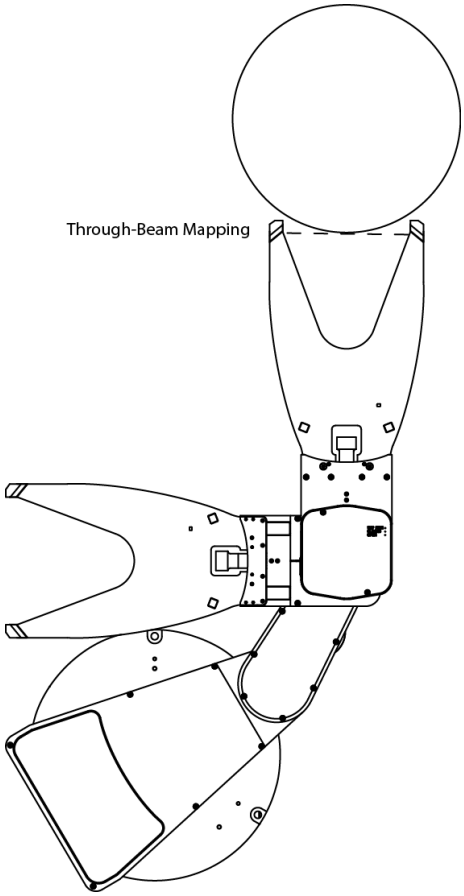


Figure 13-2: Wafer Mapping Sensors - Through-Beam Mapping

Wafer Mapping

Wafer Mapping Overview

a. Teach and Store the station coordinates for PICK and PLACE. See Chapter 1: Station Setup.
b. Align the Wafer Sensor.
c. Teach and Store the station coordinates for mapping. See "Teach Mapping Values" on page 272.

For Through-Beam ONLY:

The mechanical design of the Through-Beam sensor increases the risk of possible collision with a wafer that has slid out of the cassette. To avoid this, additional setup values must be entered.

a. Request the current Map Clearance value:

```
RQ CONFIG DATA SUPINFO MAPCLEARANCE NONE NONE
```

Copyright © 2023, Brooks Automation, Inc.

The response is:

`CONFIG DATA SUPINFO MAPCLEARANCE NONE NONE`*`mapClearanceValue`*

See "Set/Request/Store Station Map PreScan" on page 299.

b. Calculate the maximum PRESCAN_TO_MAP_DIST value:

`PRESCAN_TO_MAP_DIST = 0.9 multiplied by `*`mapClearanceValue`*

See "Set/Request/Store Station Map PreScan" on page 299.

c. Set the calculated value to the Map station:

`SET STN `*`station`*` MAP PRESCAN_TO_MAP_DIST `*`value`*

See "Set/Request/Store Station Map PreScan" on page 299.

d. Find the Bias. See "Find Bias" on page 270.

e. Verify correct Mapping of wafers by mapping a cassette and verifying that the correct information is reported. See "Set/Request/Store Map Mode" on page 278.

Find Bias

Determines the operational value of the Find Bias.

NOTE: This command is used with Through-Beam and Reflective Mapping Sensors.

The Find Bias is the distance between the beam and the BTO. The robot uses the Find Bias distance as an offset that is added to the physical location of a specific slot to calculate the distance to be moved to determine wafer presence in that slot. A successful Find Bias must be performed before for each station before a MAP command can be issued.

When the Find Bias command is issued, the robot moves so that the wafer in the cassette interrupts the WPS laser beam.

When responding to this command, the robot expects a wafer in the correct slot with the command referencing the slot.

Two values are determined: the Find Bias value and the Find Bias Thickness.

The Find Bias command does not display the values it determines. A Request command must be issued to view the Find Bias value, and a Store command must be issued to save the determined Find Bias value in non-volatile memory.

Find Bias	
Command Syntax	`FB station <<(ARM)>[(A|B)]> [SLOT `*`fbslot`*`]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Find Bias		
Arguments	station:	Station number
	ARM:	Arm descriptor
	fbslot:	Slot number that contains a wafer. Range = 3 to number of slots in the cassette minus 2 Default = 5 Two slots below this slot and two slots above this slot must be empty.
Description	Determines the operational value of the Find Bias.	
Example	`FB 1 ARM A`	

Detecting Wafers in a Cassette

During a Map, the sensor scans a cassette of wafers and reports the status of wafers in each slot.

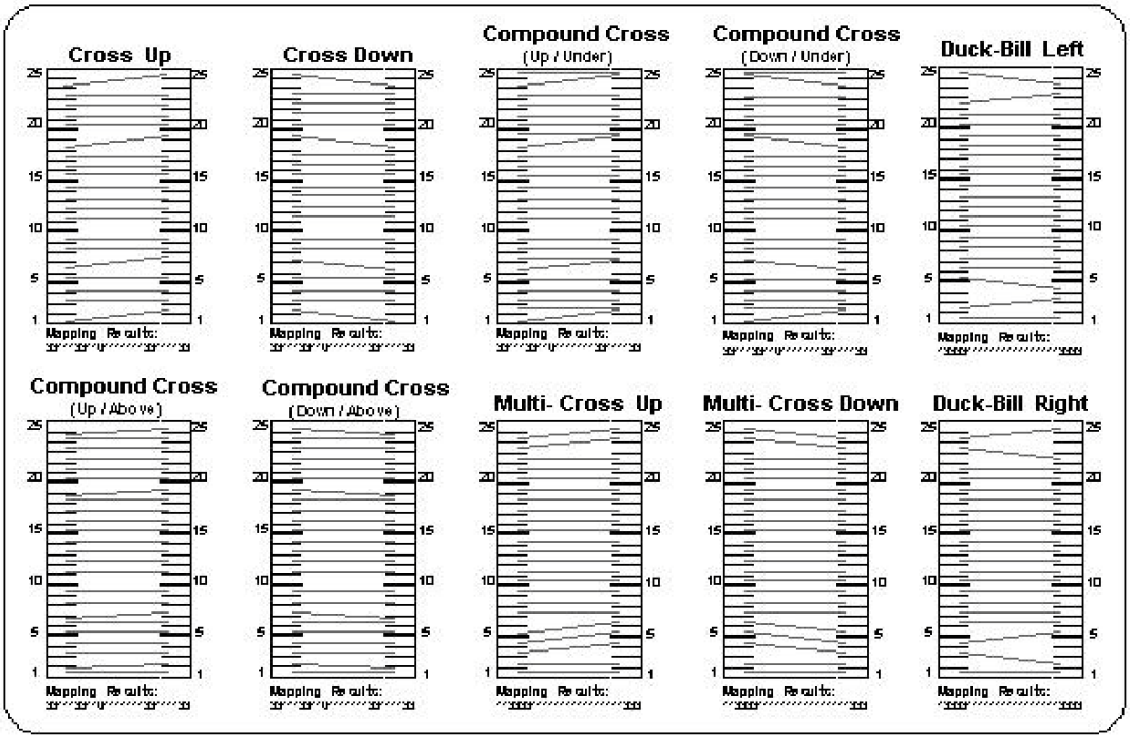
The WPS will accurately characterize the following conditions:

a. Properly slotted wafers
b. Missing wafers
c. Cross slotted wafers with or without a single adjacent wafer in a shared slot as shown (it is expected that some concessions may be necessary for accurate characterization of an upper adjacent wafer, since the configuration and nature of the system may conflict with doubles characterization).
d. Double wafers
e. Any combination of the above errors, including the presence of any combination of errors in the absence of properly placed wafers.

The mapper is not required to correctly characterize the following cases although it must indicate the results that would prevent product risk:

a. A cross slotted wafer with properly slotted wafers sharing slots on both sides. The mapper must characterize the cross slot correctly.
b. A double wafer slot shared with one edge of a cross slot. The mapper must indicate an error involving the shared slot.

Copyright © 2023, Brooks Automation, Inc.



Teach Mapping Values

To teach a robot how to Map a cassette, the following parameters are definable by the user:

a. Cross slot sensitivity parameters, relative and absolutes (i.e. single cross, hard deck and hard ceiling).
b. Double slot sensitivity parameters, relative and absolute.
c. Number of slots in the Cassette and the Cassette pitch.
d. Vertical offset below the lowest slot and above the highest slot during the map scan.

Set/Request/Store Absolute Threshold

Sets, requests, and stores the default thresholds for mapping. Setting the minimum and maximum values to the same value will effectively disable the function.

NOTE: This command is used with Reflective Mapping Sensors.

The Absolute Delta MIN is the smallest delta a wafer can have and still be called a cross. The Absolute Delta MAX is the largest delta a wafer can have and still be called a single. Between these values, the wafer could be called a cross or a single depending on the average delta of the cassette and the value set for the SINGLE CROSS parameter.

Copyright © 2023, Brooks Automation, Inc.

The Absolute Dual value is the hard deck or hard ceiling for dual delta reflective mapping. The minimum and maximum values for the dual delta can be set by the user.

- Hard Deck = MIN ABS_WIDTH x (FB Thickness) / FB_THICKNESS_NOMINAL
- Hard Ceiling =MAX ABS_WIDTH x (FB Thickness) / FB_THICKNESS_NOMINAL

Set Absolute Threshold		
Command Syntax	`SET (MIN|MAX) ABS_WIDTH` *`threshold`* `SET (MIN|MAX) ABS_DELTA` *`threshold`* `SET (MIN|MAX) ABS_DUAL` *`threshold`*	
Response Syntax	There is no response to this command.	
Arguments	ABS_WIDTH:	threshold: Threshold for Absolute thickness of the wafer MIN: default = 10000 (hard deck) MAX: default = 10000 (hard ceiling) ***NOTE:*** *Setting the minimum and maximum values to the same value will effectively disable the function.*
	ABS_DELTA:	threshold: Threshold for Absolute Delta MIN: default = 750 MAX: default = 2000
	ABS_DUAL:	threshold: Hard deck or hard ceiling threshold for dual delta MIN: default = 275 (hard deck) MAX: default = 750 (hard ceiling)
Description	Sets the default thresholds for mapping.	
Example	`SET MAX ABS_WIDTH 10000`	

Copyright © 2023, Brooks Automation, Inc.

Request Absolute Threshold		

Command Syntax	`RQ (MIN|MAX) ABS_WIDTH`
	`RQ (MIN|MAX) ABS_DELTA`
	`RQ (MIN|MAX) ABS_DUAL`	

Response Syntax	There is no response to this command.	
Arguments	ABS_WIDTH:	MIN: default = 10000 (hard deck) MAX: default = 10000 (hard ceiling) ***NOTE:*** *Setting the minimum and maximum values to the same value will effectively disable the function.*
	ABS_DELTA:	MIN: default = 750 MAX: default = 2000
	ABS_DUAL:	MIN: default = 275 (hard deck) MAX: default = 750 (hard ceiling)
Description	Requests the default thresholds for mapping.	

Example	`RQ MAX ABS_WIDTH`
	`MAX ABS WIDTH 10000`
	
	`RQ MIN ABS_WIDTH`
	`MIN ABS WIDTH 10000`
	
	`RQ MAX ABS_DELTA`
	`MAX ABS DELTA 2000`
	
	`RQ MIN ABS_DELTA`
	`MIN ABS DELTA 750`	

Store Absolute Threshold		

Command Syntax	`STORE (MIN|MAX) ABS_WIDTH`
	`STORE (MIN|MAX) ABS_DELTA`
	`STORE (MIN|MAX) ABS_DUAL`	

Response Syntax	There is no response to this command.	
Arguments	ABS_WIDTH:	MIN: default = 10000 (hard deck) MAX: default = 10000 (hard ceiling) ***NOTE:*** *Setting the minimum and maximum values to the same value will effectively disable the function.*
	ABS_DELTA:	MIN: default = 750 MAX: default = 2000
	ABS_DUAL:	MIN: default = 275 (hard deck) MAX: default = 750 (hard ceiling)
Description	Stores the default thresholds for mapping.	
Example	`STORE MAX ABS_WIDTH`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Cross

Sets, requests, and stores the default thresholds for detecting cross-slotted wafers in a cassette.

NOTE: This command is used with Through-Beam and Reflective Mapping Sensors.

If a cross-slotted wafer was incorrectly mapped, adjust the Dual Cross Threshold. If a double-slotted wafer was incorrectly mapped, adjust the Double Thickness.

During mapping, the first scan detects Single Cross and the second scan combined with the first scan determines Dual Cross. By setting the Single Cross to 20, only the Dual Cross threshold settings will affect cross-slot detection.

As the Dual Cross threshold is increased, the probability that a cross-slotted wafer will be classified as a single wafer increases; as the threshold is decreased, the probability that a single wafer will be classified as a cross-slotted increases.

Set Cross		
Command Syntax	`SET (SINGLE|DUAL) CROSS` *`value`*	
Response Syntax	There is no response to this command.	
Arguments	value:	Sets the delta for cross-slotted wafers. SINGLE CROSS: Default = 20 DUAL CROSS: Default = 510
Description	Sets the default thresholds for detecting cross-slotted wafers in a cassette.	
Example	`SET SINGLE CROSS 14`	

Request Cross		
Command Syntax	`RQ (SINGLE|DUAL) CROSS`	
Response Syntax	There is no response to this command.	
Arguments	value:	Sets the delta for cross-slotted wafers. SINGLE CROSS: Default = 20 DUAL CROSS: Default = 510
Description	Requests the default thresholds for detecting cross-slotted wafers in a cassette.	

Copyright © 2023, Brooks Automation, Inc.

Request Cross	

Example	`RQ DUAL CROSS`
	`DUAL CROSS 1`
	
	`RQ SINGLE CROSS`
	`SINGLE CROSS 14`

Store Cross		
Command Syntax	`STORE (SINGLE|DUAL) CROSS`	
Response Syntax	There is no response to this command.	
Arguments	value:	Sets the delta for cross-slotted wafers. SINGLE CROSS: Default = 20 DUAL CROSS: Default = 510
Description	Stores the default thresholds for detecting cross-slotted wafers in a cassette.	
Example	`STORE SINGLE CROSS`	

Set/Request/Store Double Thickness

Sets, requests, and stores the default thresholds for detecting double wafers in a cassette.

Double-stacked detection cannot be guaranteed due to variables such as substrate material, substrate thickness, coatings, and other factors.

NOTE: This command is used with Through-Beam and Reflective Mapping Sensors.

Double Thickness is the minimum width of two wafers in a slot to the maximum width of a single wafer. The threshold is determined using the following equation:

As the Double Thickness threshold is increased, the probability that a double wafer will be classified as a single wafer increases; as the threshold is decreased, the probability that a single wafer will be classified as a double increases.

Set Double Thickness		
Command Syntax	`SET DOUBLE THICKNESS` *value*	
Response Syntax	There is no response to this command.	
Arguments	value:	Threshold for double thickness Default = 5000
Description	Sets the default thresholds for detecting double wafers in a cassette.	

Copyright © 2023, Brooks Automation, Inc.

Set Double Thickness		
Example	`SET DOUBLE THICKNESS 150`	

Request Double Thickness		
Command Syntax	`RQ DOUBLE THICKNESS`	
Response Syntax	`RQ DOUBLE THICKNESS` *value*	
Arguments	value:	Threshold for double thickness Default = 5000
Description	Requests the default thresholds for detecting double wafers in a cassette.	
Example	`RQ DOUBLE THICKNESS` `DOUBLE THICKNESS 150`	

Store Double Thickness		
Command Syntax	`STORE DOUBLE THICKNESS`	
Response Syntax	There is no response to this command.	
Arguments	value:	Threshold for double thickness. Default = 5000
Description	Stores the default thresholds for detecting double wafers in a cassette.	
Example	`STORE DOUBLE THICKNESS`	

Go To Map

Moves to a specified station-referenced location.

This command performs all interlocking necessary to maintain safe wafer handling.

NOTE: This command is used with all robots.

Go To Map	
Command Syntax	`GOTO[N station] <<(ARM)>[(A|B)]> [MAP(EX|RE)]` `GOTO [MAP N station] <<(ARM)>[(A|B)]> [VIA}`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Go To Map		
Arguments	station:	Station number
	ARM:	Arm descriptor Arm AB is used with Radius armsets only
	MAP N:	For use with Wafer Present Sensors. Moves radial position of arm at MAP position. **EX**:Move to station's MAP position and EXtend. **RE**: Move to station's MAP position.
	VIA:	Moves to the VIA point associated with the station.
Description	Moves to a specified station-referenced location.	
Example	`GOTO N 1 MAP`	

Set/Request/Store Map Mode

Sets, requests, and stores the method of Mapping.

NOTE: This command is used with Through-Beam and Reflective Mapping Sensors.

The Map Mode performs one or more scans of the cassette to determine wafer presence and wafer positioning. For SINGLE mode, the passes are separated by the ANGLE DISTANCE+; for SINGLE+, The robot performs one scan, traveling up, and in the center. This scan is used to count wafers only. The SINGLE+ scan performs a center scan followed by a second scan performed to the right of the center scan at ANGLE DISTANCE. If the two scans do not agree, two more scans will be performed.

DUAL,DUAL+, or TRIPLE, the passes are separated by the ANGLE DISTANCE of the Map Station command.

Different modes run at different speeds. These speeds are factory set and are not user definable.

Set Map Mode	
Command Syntax	`SET MAP MODE [SINGLE|DUAL|TRIPLE|SINGLE+|DUAL+]`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Map Mode		
Arguments	station:	Station number
	ARM:	Arm descriptor
	fbslot:	Slot number that contains a wafer. Range = 3 to number of slots in the cassette minus 2 Default = 5 Two slots below this slot and two slots above this slot must be empty.
Description	Sets the method of Mapping.	
Example	`SET MAP MODE DUAL`	

Request Map Mode		
Command Syntax	`RQ MAP MODE`	
Response Syntax	`MAP MODE: MODE`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	fbslot:	Slot number that contains a wafer. Range = 3 to number of slots in the cassette minus 2 Default = 5 Two slots below this slot and two slots above this slot must be empty.
Description	Requests the method of Mapping.	
Example	`RQ MAP MODE` `MAP MODE: DUAL`	

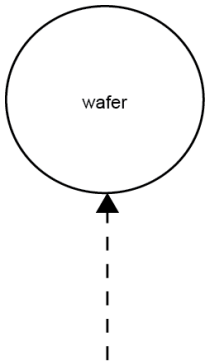
Store Map Mode	
Command Syntax	`STORE MAP MODE`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Map Mode		
Arguments	SINGLE:	*Used with all Through-Beam Sensors and some Reflective Sensors.* Robot performs a Single scan to MAP the cassette. *Normally used on all Through-Beam applications.*
	SINGLE+:	*Used with Reflective Sensors.* Robot performs a Single scan up, and to the left of the center of the cassette followed by another scan at the ANGLE DISTANCE to the right of the center of the cassette.
	DUAL:	*Used with Reflective Sensors.* Performs two scans of the same cassette. The DUAL scan moves up for one pass, then down, and up for the second pass. The number of wafers seen at each pass must be identical. If not, an error is generated.
	DUAL+:	*Used with Reflective Sensors.* Dual+ performs DUAL scan with enhanced detection capability. Normally used on all Reflective applications.
	TRIPLE:	*Used with Reflective Sensors.* Performs three scans of same cassette. Default = SINGLE in Through-Beam applications DUAL in Reflective applications
Description	Stores the method of Mapping.	
Example	`STORE MAP MODE`	

SINGLE Map Mode

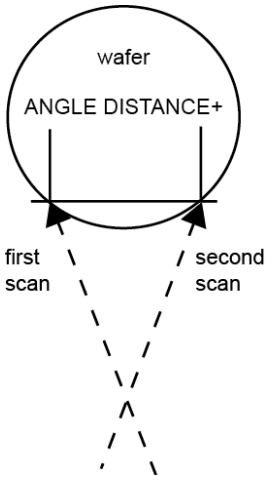
The robot performs one classification scan traveling up in the center.



Copyright © 2023, Brooks Automation, Inc.

SINGLE+Map Mode

Two scans are performed left and right of center, separated by ANGLE DISTANCE+. If the two scans do not agree, two more scans will be performed. Wafer data from the two scans is com-bined to form on Map classification.

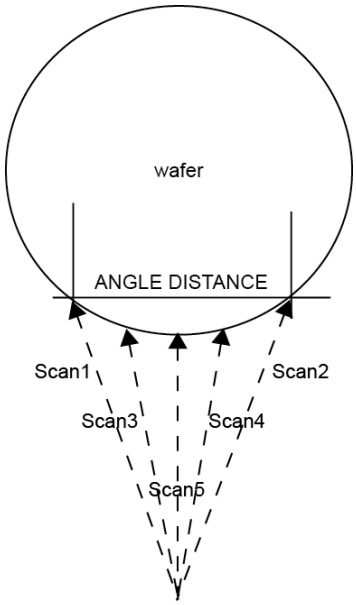


The distance between the first and second scans is an optional setting.

Copyright © 2023, Brooks Automation, Inc.

DUAL Map Mode

The robot performs two scans, each traveling up, to determine wafer count and wafer classification. The new side scan numbers are again compared. If the numbers still do not match, one scan is performed up the center. If the center scan does not match one of the previous four scans, an error is generated.

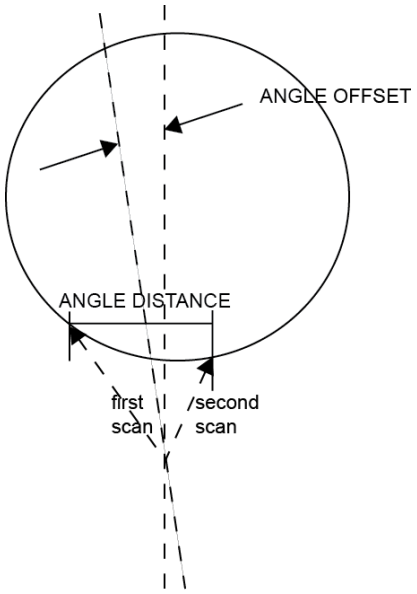


The distance between the first and second scans is an optional setting.

Copyright © 2023, Brooks Automation, Inc.

DUAL+ Map Mode

DUAL+ performs the same scans as DUAL with the addition of adding an offset. This offset can improve wafer classification.

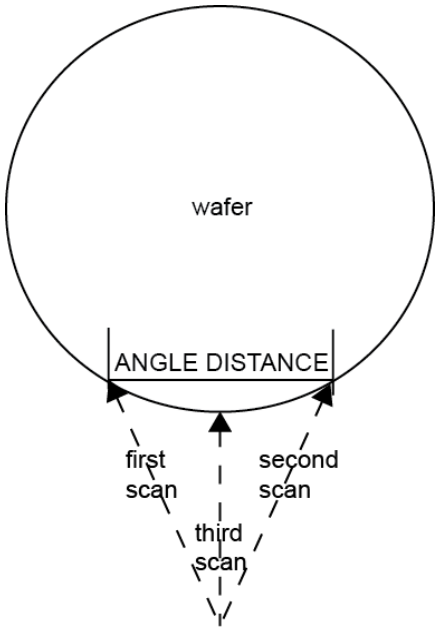


The offset distance from TVALMAP center-line is an optional setting.

Copyright © 2023, Brooks Automation, Inc.

TRIPLE Scan Mode

The robot performs three scans, each traveling up, in three different locations. The first scan is performed traveling up on the left side of the wafer. The second scan is also performed traveling up and on the right side of the wafer. The third scan is performed traveling up and in the center of the wafer. The center scan is used to count wafers. The two side scans are used for classification if the number of wafers seen by each side scan match the number seen by the center scan. The new side scan numbers are again compared to the center scan. If the numbers still do not match, an error is generated.



Set/Request/Store Map Speed Offset

Sets, requests, and stores the Mapping start position to allow constant velocity during Mapping.

NOTE: This command is used with Through-Beam Mapping Sensors.

The Map Speed command sets the distance from the Z start position to the Mapping start position to establish constant mapping velocity for better Mapping results. Map Speed allows adjusting the Z start position or lowest Z position for Mapping.

Calculation for the lowest MAP Z position:

ZStartMAP = BTO - FB - MAPmargin - SPEED_OFFSET,
where
ZStartMAP = the lowest Z position for the MAP command,
BTO = Z station parameter,
FB = distance between position where robot puts a wafer into the station and where robot sees this wafer,

Copyright © 2023, Brooks Automation, Inc.

MAPmargin = value of station Pitch divided by 2, and
SPEED_OFFSET = Z offset that allows the robot to reach a constant speed for scanning wafers.

> ***NOTE:*** The Speed_Offset parameter is a robot parameter and is common to all stations. It may be required to change this value if interference with the cassette is happening.

Set Map Speed Offset		
Command Syntax	`SET MAP SPEED_OFFSET` *value*	
Response Syntax	There is no response to this command.	
Arguments	value:	Z distance value Range = 0 to 3333 microns Default = 3333
Description	Sets the Mapping start position to allow constant velocity during Mapping.	
Example	`SET MAP SPEED_OFFSET 3000`	

Request Map Speed Offset		
Command Syntax	`RQ MAP SPEED_OFFSET`	
Response Syntax	`MAP SPEED_OFFSET` *value*	
Arguments	value:	Z distance value Range = 0 to 3333 microns Default = 3333
Description	Requests the Mapping start position to allow constant velocity during Mapping.	
Example	`RQ MAP SPEED_OFFSET` `MAP SPEED_OFFSET 3333`	

Store Map Speed Offset		
Command Syntax	`STORE MAP SPEED_OFFSET`	
Response Syntax	There is no response to this command.	
Arguments	value:	Z distance value Range = 0 to 3333 microns Default = 3333
Description	Stores the Mapping start position to allow constant velocity during Mapping.	
Example	`STORE MAP SPEED_OFFSET 3200`	

Copyright © 2023, Brooks Automation, Inc.

Map/Results/Clear Station

Maps the cassette at the specified station.

> ***NOTE:*** This command is used with Through-Beam and Reflective Mapping Sensors.
>
> The MAP feature is not supported in VT5 Compatibility.

This command updates the wafer Map for the cassette at the specified station. At power up of the robot, all map results are set to "?". After a Map command, the results will be based on the last Map with no dynamic updates. The Map results do not reflect movement of material until another Map of the station is done. If the EX_ENABLE, SBIT_ SVLV_SEN, or the VLV_SEN station interlocks are set, the robot will or will not be allowed to extend to the station depending on the state of the sensor.

All stations may be Mapped. There is no limit to the number of Maps that can be stored.

`RSR`: This command is used after a Map to read the results of the Mapping. The response is presented in digits for each slot.

`data`: The results of the last map where each digit represents a slot.
0 - no wafer detected
1- wafer detected
2 - double-slotted wafers detected
3 - cross-slotted wafer
? - not a valid slot

`SET MP CLEAR`: Clears the current wafer map information for the cassette at the specified station.

Map Station		
Command Syntax	`MAP` *`station`* `<<(ARM)>[(A|B)]>`	
Response Syntax	`MAP data`	
Arguments	station: ARM:	Station number Arm descriptor
Description	Maps the cassette at the specified station.	

Example	To map the cassette at station 2:
	`MAP 2`
	
	To map the cassette at station where the robot is presently located:
	`MAP`	

Copyright © 2023, Brooks Automation, Inc.

Results Station		
Command Syntax	`RSR [station] <<(ARM)>[(A|B)]>`	
Response Syntax	There is no response to this command.	
Arguments	station: ARM:	Station number Arm descriptor
Description	This command is used after a Map to read the results of the Mapping. The response is presented in digits for each slot.	

Example	The last map performed was at station 2. To request the last map performed:
	`RSR 2`
	`MAP 111110000011111111111111`	

Clear Station		
Command Syntax	`SET MP CLEAR station <<(ARM)>[(A|B)]>`	
Response Syntax	There is no response to this command.	
Arguments	station: ARM:	Station number Arm descriptor
Description	Resets the current wafer map information for the cassette at the specified station.	

Example	The following response reports wafers present and positioned correctly in slots 1 through 5 and 11 through 25. No wafers are in slots 6 through 10; no wafers are crossslotted.
	`MAP 111110000011111111111111`
	To clear the map results at station 2:
	`SET MP CLEAR 2`	

Set/Request/Store Map Wafer Presence Check

Sets, requests, and stores the interlock to check if a wafer is present before a mapping command is performed.

NOTE: This command is used with Through-Beam Mapping Sensors.

If the Wafer Presence Checking is enabled, the robot will verify that a wafer is not on the end effector before mapping. If a wafer is present, the robot will not execute the mapping command and an error will be reported. The robot will assume that a wafer is on the end effector at start-up.

Copyright © 2023, Brooks Automation, Inc.

Set Map Wafer Presence Check		
Command Syntax	`SET MAP WAF_PRS_CHK [ENB|DIS] <<(ARM)>[(A|B|BOTH)]>`	
Response Syntax	There is no response to this command.	
Arguments	WAF_PRS_CHK: ARM:	Wafer Presence Check verification ENB: Enabled DIS: Disabled (default) Arm descriptor
Description	Sets the interlock to check if a wafer is present before a mapping command is performed.	
Example	To set the robot to check for a wafer on the end effector before mapping: `SET MAP WAF_PRS_CHK ENB`	

Request Map Wafer Presence Check		
Command Syntax	`RQ MAP WAF_PRS_CHK <<(ARM)>[(A|B|BOTH)]>`	
Response Syntax	In Monitor Mode: `MAP WAF_PRS_CHK (ARM A|ARM B|ARM BOTH) (ENB|DIS) _RDY`	
Arguments	WAF_PRS_CHK: ARM:	Wafer Presence Check verification ENB: Enabled DIS: Disabled (default) Arm descriptor
Description	Requests the interlock to check if a wafer is present before a mapping command is performed.	
Example	To request the state of the Wafer Presence Interlock: `RQ MAP WAF_PRS_CHK MAP WAF_PRS_CHK ARM A ENB`	

Store Map Wafer Presence Check		
Command Syntax	`STORE MAP WAF_PRS_CHK <<(ARM)>[(A|B|BOTH)]>`	
Response Syntax	There is no response to this command.	
Arguments	WAF_PRS_CHK: ARM:	Wafer Presence Check verification ENB: Enabled DIS: Disabled (default) Arm descriptor
Description	Stores the interlock to check if a wafer is present before a mapping command is performed.	

Copyright © 2023, Brooks Automation, Inc.

Store Map Wafer Presence Check	
Example	To store the current mode: `STORE MAP WAF_PRS_CHK`

Set/Request/Store Map with VIA

Sets, requests, and stores the interlock to check if a wafer is present before a mapping command is performed.

NOTE: This command is used with Reflective Mapping Sensors.

Reflective Map Via: Robots with Reflective Mappers normally do not allow VIA Point Mapping. To allow VIA Point mapping with a Reflective Sensor, set the REFL_MAP_VIA option to enabled.

Set Map with VIA		
Command Syntax	`SET MAP REFL_MAP_VIA (DIS|ENB)`	
Response Syntax	There is no response to this command.	
Arguments	REFL_MAP_VIA:	Map VIA Point values with Reflective Sensor. ENB: Enabled DIS: Disabled (default)
Description	Sets the interlock to check if a wafer is present before a mapping command is performed.	
Example	`SET MAP REFL_MAP_VIA DIS`	

Request Map with VIA		
Command Syntax	`RQ MAP REFL_MAP_VIA`	
Response Syntax	`MAP REFL_MAP_VIA REFL_MAP_VIA`	
Arguments	REFL_MAP_VIA:	Map VIA Point values with Reflective Sensor. ENB: Enabled DIS: Disabled (default)
Description	Requests the interlock to check if a wafer is present before a mapping command is performed.	
Example	`RQ MAP REFL_MAP_VIA` `MAP REFL_MAP_VIA DIS`	

Copyright © 2023, Brooks Automation, Inc.

Store Map with VIA		
Command Syntax	`STORE MAP REFL_MAP_VIA`	
Response Syntax	`MAP REFL_MAP_VIA REFL_MAP_VIA`	
Arguments	REFL_MAP_VIA:	Map VIA Point values with Reflective Sensor. ENB: Enabled DIS: Disabled (default)
Description	Stores the interlock to check if a wafer is present before a mapping command is performed.	
Example	`STORE MAP DIS` `MAP REFL_MAP_VIA DIS`	

Set/Request/Store Slot Range

Sets, requests, and stores the relative percentage of pitch above the wafer during mapping.

NOTE: This command is used with Through-Beam and Reflective Mapping Sensors.

This command may be used to improve Mapping evaluation. When the software is determining which slot to put a scanned wafer in, it uses a range the size of one pitch. Part of this range is above an ideal wafer (determined from FB), and part is below. *Value* is the percentage of pitch that is used above the ideal wafer where the robot “locates” the wafer. The percentage of pitch used below the ideal wafer is (100 - *value*).

Set Slot Range		
Command Syntax	`SET SLOT_RANGE` *value*	
Response Syntax	There is no response to this command.	
Arguments	value:	Cassette Slot range. 5 < *value* < 95 Default=85
Description	Sets the relative percentage of pitch above the wafer during mapping.	
Example	`SET SLOT_RANGE 25`	

Request Slot Range	
Command Syntax	`RQ SLOT_RANGE`
Response Syntax	`SLOT RANGE 50`
Arguments	None

Copyright © 2023, Brooks Automation, Inc.

Request Slot Range	
Description	Requests the relative percentage of pitch above the wafer during mapping.
Example	`RQ SLOT_RANGE` `SLOT RANGE 50`

Store Slot Range	
Command Syntax	`STORE SLOT_RANGE`
Response Syntax	There is no response to this command.
Arguments	None
Description	Stores the relative percentage of pitch above the wafer during mapping.
Example	`STORE SLOT RANGE`

Set/Request/Store/Remove Station Map

Sets, requests, stores, and removes the absolute coordinate values of the various wafer mapping features.

NOTE: This command is used with Through-Beam and Reflective Mapping Sensors.

The robot uses the Mapping sensors to detect wafer presence. To correctly report the presence of a wafer, the Map operation values must be set.

The SET FIND BIAS command manually sets the Find Bias to the values specified by the user. Brooks does not recommend entering these values with this command. Instead, determine the bias values automatically by using the command "Find Bias" on page 270.

The FB THICKNESS is the wafer thickness. This value may be entered using the SET FB THICKNESS command; however, Brooks does not recommend entering these values with this command. Instead, determine the wafer thickness values by using the command "Find Bias" on page 270.

The ANGLE DISTANCE command will enter the user-defined value for the scan locations of the Dual and Triple scan modes ("Set/Request/Store Map Mode" on page 278). This distance is an actual physical measurement in microns as determined by the desired position of the first scan to the desired position of the second scan. If the ANGLE DISTANCE is changed for a station, the Find Bias for that station must be repeated. See "Find Bias" on page 270.

ANGLE OFFSET is an angle added to tValMap + 1/2 ANGLE DISTANCE before the scan angles are calculated. This is a signed number, so a positive value causes a clockwise shift, and a negative value causes a counterclockwise shift. The distance between the two mapping scans is determined

Copyright © 2023, Brooks Automation, Inc.

by ANGLE DISTANCE. For example, if the ANGLE DISTANCE between the scans for a certain station is 4000 milli-degrees and the tValMap for that station is 100000 milli-degrees the with the ANGLE OFFSET set to zero, the two scans would be 98000 and 102000. However, if the ANGLE OFFSET is set to 1500 milli-degrees, the new scans would be at 99500 and 103500 milli-degrees respectively.

The REMOVE MAP command will remove only the mapping station information (not the station information).

Table 13-1: Arguments

Argument	Description
station:	Station number If station is not entered, current station will be default.
findbias:	*FB for SINGLE Mode* *FB1 and FB2 for SINGLE +, DUAL, DUAL+, and TRIPLE.* Distance between the WTP and the sensor in microns See "Find Bias" on page 270. Field Size = 6
fbthick:	*FB THICKNESS for SINGLE Mode* *FB THICKNESS1 and FBTHICKNESS2 for SINGLE +, DUAL, DUAL+, and TRIPLE* Wafer thickness in units of microns See "Find Bias" on page 270. Field Size = 6
rValMap:	This command is used with Through-Beam and Reflective Mapping. Radial absolute value, in microns, for specified mapping station. Distance from the center of the robot to the center of the end effector when in the mapping position. Field Size = 7
tValMap:	*This command is used with Through-Beam and Reflective Mapping.* Theta absolute value, in millidegrees, for specified mapping station. Angle between taught HOME T position and the center of the end effector when in the mapping position. Field Size = 6
wValMap:	*This command is used with Through-Beam and Reflective Mapping on the 4-Axis Armset.* Theta wrist absolute value, in millidegrees, for specified mapping station. The absolute value is defined as the angle between the taught HOME T position and the center of the end effector in the mapping position pointed to the center of the wafer. Field Size = 6
waValMap:	*This command is used with Through-Beam and Reflective Mapping on the 5-Axis Armset.* Theta wrist absolute value, in millidegrees, of Arm A of the specified mapping station. The absolute value is defined as the angle between the taught HOME T position and the center of the end effector of Arm A in the mapping position pointed to the center of the wafer. Field Size = 6

Copyright © 2023, Brooks Automation, Inc.

Argument	Description
wbValMap:	*This command is used with Through-Beam and Reflective Mapping on the 5-Axis Armset.* Theta wrist absolute value, in millidegrees, of Arm B of the specified mapping station. The absolute value is defined as the angle between the taught HOME T position and the center of the end effector of Arm B in the mapping position pointed to the center of the wafer. Field Size = 6
ANGLE:	*This command is used with Reflective Mapping only.* `DISTANCE angleDist`: Dual `DISTANCE+ angleDistPlus`: Single+ `OFFSET angleOffset`: Dual+ Extra offset applied to the ANGLE DISTANCE Range = +/- 3000 milli-degrees Default = 0

Set Station Map	

Command Syntax	`SET STN station <<(ARM)>[(A|B)]> MAP`
	`[RVALMAP rValMap] [TVALMAP tValMap]`
	`[WVALMAP wValMap][WAVALMAP waValMap|WBVALMAP wbValMap] [FB findbias]`
	
	`[FB THICKNESS fbThick]`
	`[[(FB1 findbias1)|(FB THICKNESS1 fbThick1)]`
	`[(FB2 findbias2)|(FB THICKNESS2 fbThick2)]]`
	`[ANGLE (DISTANCE angleDist|DISTANCE+ angleDistPlus|OFFSET angleOffset)]`

Response Syntax	**SINGLE Mode Response:**
	`RVALMAP ---------- rValMap`
	`TVALMAP ---------- tValMap`
	`WVALMAP ---------- wValMap`
	`WAVALMAP ---------- waValMap`
	`WBVALMAP ---------- wbValMap`
	`FB ---------- findBias`
	`FB THICKNESS ------ fbThick`
	
	SINGLE+, DUAL, DUAL+, TRIPLE Mode Response:
	`RVALMAP -------- rValMap`
	`TVALMAP --------- tValMap`
	`WVALMAP --------- wValMap`
	`WAVALMAP -------- waValMap`
	`WBVALMAP -------- wbValMap`
	`FB1 ------------ findBias1`
	`FB THICKNESS1 --- fbThick1`
	`FB2 ------------ findBias2`
	`FB THICKNESS2 --- fbThick2`
	`ANGLE DISTANCE -- angleDist`
	`ANGLE DISTANCE+ - angleDistPlus`
	`ANGLE OFFSET ---- angleOffset`

Arguments	See Table 13-1.
Description	Sets the absolute coordinate values of the various wafer mapping features.
Example	`SET STN 1 MAP RVALMAP 410000 T VALMAP 180000 WVALMAP 180000`

Copyright © 2023, Brooks Automation, Inc.

Request Station Map	
Command Syntax	RQ STN *station* <<(ARM)>[(A|B)]> MAP [ALL] {[RVALMAP] [TVALMAP] [WVALMAP] [WAVALMAP|WBVALMAP]} [FB|FB THICKNESS] [FB1|FB2|FB THICKNESS1|FB THICKNESS2] [ANGLE (DISTANCE|DISTANCE+|OFFSET)]
Response Syntax	**SINGLE Mode Response:** RVALMAP ---------- rValMap TVALMAP ---------- tValMap WVALMAP ---------- wValMap WAVALMAP ---------- waValMap WBVALMAP ---------- wbValMap FB ---------- findBias FB THICKNESS ------ fbThick **SINGLE+, DUAL, DUAL+, TRIPLE Mode Response:** RVALMAP -------- rValMap TVALMAP --------- tValMap WVALMAP --------- wValMap WAVALMAP -------- waValMap WBVALMAP -------- wbValMap FB1 ---------- findBias1 FB THICKNESS1 --- fbThick1 FB2 ---------- findBias2 FB THICKNESS2 --- fbThick2 ANGLE DISTANCE -- angleDist ANGLE DISTANCE+ - angleDistPlus ANGLE OFFSET ---- angleOffset
Arguments	See Table 13-1
Description	Requests the absolute coordinate values of the various wafer mapping features.

Copyright © 2023, Brooks Automation, Inc.

Request Station Map	
Example	RQ MAP MODE MAP MODE: DUAL RQ STN 01 A MAP FB1 STN 0001 000000 RQ STN 01 A MAP FB THICKNESS1 STN 0001 000000 RQ STN 01 A MAP FB2 STN 0001 000000 RQ STN 01 A MAP FB THICKNESS2 STN 0001 000000 RQ STN 01 A MAP ANGLE DISTANCESTN 0001 038000 RQ STN 01 A MAP RVALMAP STN 0001 0000000 RQ STN 01 A MAP TVALMAP STN 0001 000000 RQ STN MAP ALL STN 0001 A MAP STN 0002 A MAP STN 0003 A MAP RQ STN 1 MAP ALL RVALMAP ------------- 0330847 TVALMAP ------------- 265027 WVALMAP ------------- 265027 FB ------------------ 05500 FB THICKNESS -------- 000600 VIA DIST ------------ 000000 VIA POSR ------------ 000000 VIA POST ------------ 000000 VIA POSW ------------ 000000 VIA PATH_OVER_CENTER- N MAX_PRESCAN --------- 044450 PRESCAN_TO_MAP_DIST - 011430 CURVE_MOVE_MAP ------ N RQ MAP MODE MAP MODE: SINGLE RQ STN 01 A MAP FB STN 0001 000000

Copyright © 2023, Brooks Automation, Inc.

Request Station Map	
	RQ STN 01 A MAP FB THICKNESS STN 0001 000000 RQ STN 01 A MAP RVALMAP STN 0001 0000000 RQ STN 01 A MAP TVALMAP STN 0001 000000

Store Station Map	
Command Syntax	STORE STN *station* <<(ARM)>[(A|B)]> MAP {[RVALMAP] [TVALMAP] [WVALMAP] [WAVALMAP|WBVALMAP]} [FB|FB THICKNESS] [FB1|FB2|FB THICKNESS1|FB THICKNESS2] [ANGLE (DISTANCE|DISTANCE+|OFFSET)]
Response Syntax	**SINGLE Mode Response:** RVALMAP ---------- rValMap TVALMAP ---------- tValMap WVALMAP ---------- wValMap WAVALMAP ---------- waValMap WBVALMAP ---------- wbValMap FB ---------- findBias FB THICKNESS ------ fbThick **SINGLE+, DUAL, DUAL+, TRIPLE Mode Response:** RVALMAP -------- rValMap TVALMAP --------- tValMap WVALMAP --------- wValMap WAVALMAP -------- waValMap WBVALMAP -------- wbValMap FB1 ------------ findBias1 FB THICKNESS1 --- fbThick1 FB2 ------------ findBias2 FB THICKNESS2 --- fbThick2 ANGLE DISTANCE -- angleDist ANGLE DISTANCE+ - angleDistPlus ANGLE OFFSET ---- angleOffset
Arguments	See Table 13-1.
Description	Stores the absolute coordinate values of the various wafer mapping features.
Example	STORE STN 1 MAP ALL

Copyright © 2023, Brooks Automation, Inc.

Remove Station Map		
Command Syntax	`REMOVE STN station <<(ARM)>[(A|B)]> MAP`	

Response Syntax	**SINGLE Mode Response:**
	`RVALMAP ---------- rValMap`
	`TVALMAP ---------- tValMap`
	`WVALMAP ---------- wValMap`
	`WAVALMAP ---------- waValMap`
	`WBVALMAP ---------- wbValMap`
	`FB ---------- findBias`
	`FB THICKNESS ------ fbThick`
	
	SINGLE+, DUAL, DUAL+, TRIPLE Mode Response:
	`RVALMAP -------- rValMap`
	`TVALMAP --------- tValMap`
	`WVALMAP --------- wValMap`
	`WAVALMAP -------- waValMap`
	`WBVALMAP -------- wbValMap`
	`FB1 ------------- findBias1`
	`FB THICKNESS1 --- fbThick1`
	`FB2 ------------- findBias2`
	`FB THICKNESS2 --- fbThick2`
	`ANGLE DISTANCE -- angleDist`
	`ANGLE DISTANCE+ - angleDistPlus`
	`ANGLE OFFSET ---- angleOffset`	

Arguments	station:	Station number. If station is not entered, current station will be default.
Description	Removes the absolute coordinate values of the various wafer mapping features.	
Example	`REMOVE STN 1 MAP`	

Set/Request/Store Station Map Curve Move Map

Sets, requests, and stores the mode for using the MAP VIA point in Mapping.

NOTE: This command is used with Through-Beam and Reflective Mapping Sensors.

The CURVE_MOVE_MAP command allows operation of the VIA point moves during Mapping. This command is used with additional station settings to perform compound moves rather that traditional pure radial moves by introducing a MAP VIA point.

Set Station Map Curve Move Map	
Command Syntax	`SET STN station <<(ARM)>[(A|B)]> MAP CURVE_MOVE_MAP (Y|N)`
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Station Map Curve Move Map		
Arguments	station:	Station number
	ARM:	Arm descriptor
	CURVE_MOVE_MAP:	VIA Point Curve Move mode Y: Sets the VIA option ON. N: Sets the VIA option OFF. VIA point moves are allowed only if a non-zero value for the point is entered and the CURVE_MOVE_MAP option is set to Y.
Description	Sets, requests, and stores the mode for using the MAP VIA point in Mapping.	
Example	`SET STN 1 MAP CURVE_MOVE_MAP Y`	

Request Station Map Curve Move Map		
Command Syntax	`RQ STN station <<(ARM)>[(A|B)]> MAP CURVE_MOVE_MAP`	
Response Syntax	`STN 0001 N`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	CURVE_MOVE_MAP:	VIA Point Curve Move mode VIA point moves are allowed only if a non-zero value for the point is entered and the CURVE_MOVE_MAP option is set to Y.
Description	Requests the mode for using the MAP VIA point in Mapping.	
Example	`RQ STN 1 MAP CURVE_MOVE_MAP` `STN 0001 N`	

Store Station Map Curve Move Map		
Command Syntax	`STORE STN station <<(ARM)>[(A|B)]> MAP CURVE_MOVE_MAP`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	CURVE_MOVE_MAP:	VIA Point Curve Move mode VIA point moves are allowed only if a non-zero value for the point is entered and the CURVE_MOVE_MAP option is set to Y.
Description	Stores the mode for using the MAP VIA point in Mapping.	
Example	`STORE STN 1 MAP CURVE_MOVE_MAP`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Station Map PreScan

Sets, requests, and stores data to prevent wafer collision while Mapping.

NOTE: This command is used with Through-Beam Mapping Sensors.

The PreScan feature allows any wafer slide out to be verified before a final Map will be performed. PreScans are performed at pan speed. Scans are alternated up and down.

The first PreScan will start either up or down so that the final PreScan ends down, ready for Map. If a wafer is detected during a PreScan, all motion will stop, the arm will retract, and all 3s will be displayed as the Map results. The first PreScan will look for a wafer at the max_prescan position (Position 3). The sensor begins the scan of the cassette at the MAX_PRESCAN distance. Then the mapper moves closer to the cassette and performs another prescan (Position 2). The mapper continues to move closer by each PRESCAN_TO_MAP_DIST step until it reaches the actual mapping position where it performs the actual scan of the cassette. To properly detect and avoid collisions, the prescanning parameters must be set correctly.

`MAX_PRESCAN`

Distance, in microns, that a wafer can protrude from a cassette that must be detected. The first PRESCAN will look for a wafer at this position. For Load Port Modules, this distance is to the Wafer Slide Out sensor. The MAX_PRESCAN distance cannot be greater than the distance between the Mapping position and the MAP VIA position

`PRESCAN_TO_MAP_DISTANCE`

The PRESCAN_TO_MAP_DISTANCE is the distance, in microns, between two scans or a “step”. If the amount of each step does not divide evenly, the difference will be made up at the first PreScan (PreScan 1). PreScan 2 and 3 are the PRESCAN_TO_MAP_ DISTANCE.

Copyright © 2023, Brooks Automation, Inc.

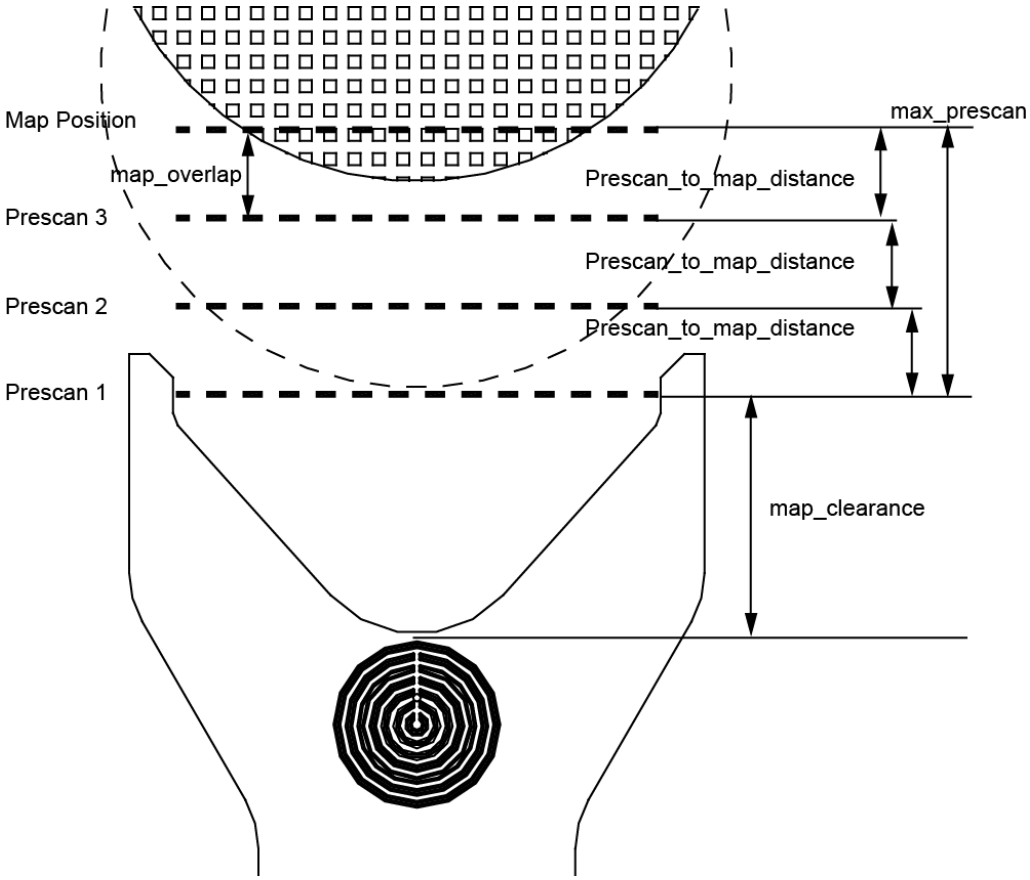


Figure 13-3: PreScan Setup

Setting the Prescan Values

Step	Action
1.	Teach the R and T values for map_overlap (Position 1 in See Figure 13-3). Map_overlap is the mapping position for the station. The R value is typically measured from the beam to the edge of the wafer. The radial value increases as the end effector moves toward the cassette. For 200mm wafer with flat fiducial=0.2 inches (5080 microns). For 200mm wafer with notch or no fiducial=0.7 inches (2000 microns). For 300mm wafer=0.7 inches (2000 microns). The map_overlap is set as RVALMAP rValMap in "Set/Request/Store/Remove Station Map" on page 291.

Copyright © 2023, Brooks Automation, Inc.

Step	Action

2.	Set the PRESCAN_TO_MAP_DISTANCE. This value sets the maximum amount a wafer is allowed to slide out of the cassette.
	
	`SET STN station MAP PRESCAN_TO_MAP_DIST 11430`
	`STORE STN station MAP PRESCAN_TO_MAP_DIST`
	
	The PRESCAN_TO_MAP_DIST value must be greater than the MAP_ OVERLAP value by at least 1-2 mm.

3.	Set the MAX_PRESCAN distance. This value determines how close the back of the end effector can come to the cassette without breaking a wafer that has slid out of the cassette. In the following example, this value is set at 1.25”.
	
	`SET STN station MAP MAX_PRESCAN 31750`
	`STORE STN station MAP MAX_PRESCAN`

4.	To verify the Prescan settings are correct, perform a Find Bias. Refer to "Find Bias" on page 270.

Set Map PreScan		
Command Syntax	`SET STN station <<(ARM)>[(A|B)]> MAP [MAX_PRESCAN maxPreScan] [PRESCAN_TO_MAP_DIST preScanMapDist]`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	maxPreScan:	The distance, in microns, that a wafer can protrude out of the cassette Field Size = 6 Default = 44450
	preScanMapDist:	Defines the size of the steps between scan. See description below for behavior with older versions of software. Field Size: 6 Default = 11430 microns
Description	Sets data to prevent wafer collision while Mapping.	

Example	`SET STN 1 MAP MAX_PRESCAN 44500`
	
	`SET STN 1 MAP PRESCAN_TO_MAP_DIST 11430`	

Request Map PreScan	
Command Syntax	`RQ STN station <<(ARM)>[(A|B)]> MAP [MAX_PRESCAN] [PRESCAN_TO_MAP_ DIST]`

Copyright © 2023, Brooks Automation, Inc.

Request Map PreScan		
Response Syntax	`PRESCAN_TO_MAP_DIST --- preScanMapDist`	
Arguments	station: maxPreScan: preScanMapDist:	Station number The distance, in microns, that a wafer can protrude out of the cassette Field Size = 6 Default = 44450 Defines the size of the steps between scan. See description below for behavior with older versions of software. Field Size: 6 Default = 11430 microns
Description	Requests data to prevent wafer collision while Mapping.	

Example	`RQ MAX_PRESCAN`
	`MAX_PRESCAN -------------- 044450`
	
	`RQ PRESCAN_TO_MAP_DIST PRESCAN_`
	`TO_MAP_DIST --------------- 011430`	

Store Map PreScan		
Command Syntax	`STORE STN` *`station`* `<<(ARM)>[(A|B)]> MAP [MAX_PRESCAN] [PRESCAN_TO_ MAP_DIST]`	
Response Syntax	`PRESCAN_TO_MAP_DIST --- preScanMapDist`	
Arguments	station: maxPreScan: preScanMapDist:	Station number The distance, in microns, that a wafer can protrude out of the cassette Field Size = 6 Default = 44450 Defines the size of the steps between scan See description below for behavior with older versions of software. Field Size: 6 Default = 11430 microns
Description	Stores data to prevent wafer collision while Mapping.	
Example	`STORE STN 1 MAP MAX_PRESCAN`	

Set/Request/Store Station Map Retract Position W

Sets, requests, and stores data to support trailing arm with retracted position at Map Station.

Copyright © 2023, Brooks Automation, Inc.

NOTE: This command is used with Through-Beam Mapping Sensors on the 5-Axis Armset.

Set Station Map Retract Position W		

Command Syntax	`SET STN station <<(ARM)>[(A|B)]> MAP RE_POSW_ENB (Y|N)`
	
	`SET STN station <<(ARM)>[(A|B)]> MAP RE (POSWA mapRePosWa|POSWB mapRePosWb)`	

Response Syntax	`RE_POSW_ENB -------------- mapRePosWEnable (Y|N)`	

Arguments	station:
	ARM:
	RE_POSW_ENB:
	POSWx :
		Station number
		Arm descriptor
		Retract Position W-axis mode at Map station
		Y: Mode is enabled.
		N: Mode is disabled.
		Retracted position of trailing arm coordinate in millidegrees
		`POSWA mapRePosWa: Arm A`
		`POSWB mapRePosWb: Arm B`

Description	Sets data to support trailing arm with retracted position at Map Station.	
Example	`SET STN 1 MAP RE_POSW_ENB Y`	

Request Station Map Retract Position W		
Command Syntax	`RQ STN station <<(ARM)>[(A|B)]> MAP [RE_POSW_ENB] [RE (POSWA|POSWB)]`	
Response Syntax	`RE POSWA -------------------- mapRePosWa`	

Arguments	station:
	ARM:
	RE_POSW_ENB:
	POSWx :
		Station number
		Arm descriptor
		Retract Position W-axis mode at Map station
		Y: Mode is enabled.
		N: Mode is disabled.
		Retracted position of trailing arm coordinate in millidegrees
		`POSWA mapRePosWa: Arm A`
		`POSWB mapRePosWb: Arm B`

Description	Requests data to support trailing arm with retracted position at Map Station.	
Example	`RQ STN 1 MAP RE_POSW_ENB`	

Copyright © 2023, Brooks Automation, Inc.

Store Station Map Retract Position W		
Command Syntax	`STORE STN station <<(ARM)>[(A|B)]> MAP [RE_POSW_ENB] [RE (POSWA|POSWB)]`	
Response Syntax	`RE POSWB ---------- mapRePosWb`	

Arguments	station:
	ARM:
	RE_POSW_ENB:
	POSWx :
		Station number
		Arm descriptor
		Retract Position W-axis mode at Map station
		Y: Mode is enabled.
		N: Mode is disabled.
		Retracted position of trailing arm coordinate in millidegrees
		`POSWA mapRePosWa: Arm A`
		`POSWB mapRePosWb: Arm B`

Description	Stores data to support trailing arm with retracted position at Map Station.	
Example	`STORE STN 1 MAP RE_POSW_ENB`	

Set/Request/Store Station Map VIA Point

Sets, requests, and stores the VIA point for Mapping functions for the specified station.

> ***NOTE:*** This command is used with the SCARA armset and Through-Beam Mapping Sensors.

VIA point moves are only allowed if the CURVE_MOVE_MAP option is set to Y. This command is used with additional station and mapping settings to perform compound moves rather that traditional pure radial moves by introducing a VIA point.

To perform straight, non-compound moves to a specified station, set the value of POSR to zero for that station. The value for POST will be remembered by the robot, but will not be used unless the POSR value is set to non-zero again.

If a VIA point is set, modifications to station values are not allowed if R is changed more than 10mm or T is changed more than 3°. To change these stations values, the VIA point must be cleared first. To clear the VIA point, set the station option for the VIA point to zero or enter SET STN station MAP VIA CLEAR.

Copyright © 2023, Brooks Automation, Inc.

Set Station Map VIA Point		
Command Syntax	`SET STN station <<(ARM)>[(A|B)]> MAP VIA [(DIST mapViadist)] [(POSR mapViaPosR)|(POST mapViaPosT)|(POSW mapViaPosW)] [(POSWA mapViaPosWa)|(POSWB mapViaPosWb)] [PATH_OVER_CENTER (Y|N)] [CLEAR]`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	DIST:	`mapViadist`: The distance value, in microns, for the specified mapping VIA point
	POSR:	`mapviaPosR`: The radial coordinate of the VIA point in microns A non-zero value for POSR causes Compound Moves to occur for the designated station. Setting this value to zero causes only straight moves to occur for the designated station.
	POST:	`mapviaPosT`: The theta coordinate of the VIA point in millidegrees
	POSW:	This command is used with the 4-axis armsets only. `mapviaPosW`: The rotating wrist coordinate of the VIA point in millidegrees
	POSWA:	This command is used with the 5-axis armsets only. `mapviaPosWa`: The Articulated Wrist coordinate for Arm A VIA point in millidegrees
	POSWB:	`mapviaPosWb`: The Articulated Wrist coordinate for Arm B VIA point in millidegrees
	PATH_OVER_CENTER:	This command is used with the 4-axis and 5-axis armsets only. Allows the wrist to pass over or go around Singular Point. Y: Move through the center point. N: Move around the center point.
	CLEAR:	Sets the VIA Mapping Point values to zero.
Description	Sets the VIA point for Mapping functions for the specified station.	

Example	`SET STN 3 MAP VIA PATH_OVER_CENTER Y`
	
	`SET STN 1 MAP VIA POSR 592200`
	`SET STN 1 MAP VIA POST 238000`
	`SET STN 1 MAP VIA POSW 238000`	

Copyright © 2023, Brooks Automation, Inc.

Request Station Map VIA Point		
Command Syntax	`RQ STN station <<(ARM)>[(A|B)]> MAP VIA [(DIST|POSR|POST|POSW|POSWA|POSWB|PATH_OVER_CENTER|ALL)]`	
Response Syntax	`STN 0001 000000 000000 000000 000000 N`	
Arguments	station:	Station number
	ARM:	Arm descriptor
	DIST:	`mapViadist`: The distance value, in microns, for the specified mapping VIA point
	POSR:	`mapviaPosR`: The radial coordinate of the VIA point in microns A non-zero value for POSR causes Compound Moves to occur for the designated station. Setting this value to zero causes only straight moves to occur for the designated station.
	POST:	`mapviaPosT`: The theta coordinate of the VIA point in millidegrees
	POSW:	This command is used with the 4-axis armsets only. `mapviaPosW`: The rotating wrist coordinate of the VIA point in millidegrees
	POSWA:	This command is used with the 5-axis armsets only. `mapviaPosWa`: The Articulated Wrist coordinate for Arm A VIA point in millidegrees
	POSWB:	`mapviaPosWb`: The Articulated Wrist coordinate for Arm B VIA point in millidegrees
	PATH_OVER_CENTER:	This command is used with the 4-axis and 5-axis armsets only. Allows the wrist to pass over or go around Singular Point.
Description	Requests the VIA point for Mapping functions for the specified station.	
Example	`RQ STN 1 MAP VIA ALL` `STN 0001 000000 000000 000000 000000 N`	

Copyright © 2023, Brooks Automation, Inc.

Store Station Map VIA Point		
Command Syntax	`STORE STN station <<(ARM)>[(A|B)]> MAP VIA [(DIST|POSR|POST|POSW|POSWA|POSWB|PATH_OVER_CENTER)]`	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor
	DIST:	`mapViadist`: The distance value, in microns, for the specified mapping VIA point
	POSR:	`mapviaPosR`: The radial coordinate of the VIA point in microns A non-zero value for POSR causes Compound Moves to occur for the designated station. Setting this value to zero causes only straight moves to occur for the designated station.
	POST:	`mapviaPosT`: The theta coordinate of the VIA point in millidegrees
	POSW:	This command is used with the 4-axis armsets only. `mapviaPosW`: The rotating wrist coordinate of the VIA point in millidegrees
	POSWA:	This command is used with the 5-axis armsets only. `mapviaPosWa`: The Articulated Wrist coordinate for Arm A VIA point in millidegrees
	POSWB:	`mapviaPosWb`: The Articulated Wrist coordinate for Arm B VIA point in millidegrees
	PATH_OVER_CENTER:	This command is used with the 4-axis and 5-axis armsets only. Allows the wrist to pass over or go around Singular Point.
Description	Stores the VIA point for Mapping functions for the specified station.	
Example	`STORE STN 3 MAP ALL`	

Request Thickness

Requests the thickness of each wafer in a slot.

NOTE: This command is used with Through-Beam and Reflective Mapping Sensors.

The RQ THICKNESS command will support Scan Modes SINGLE, SINGLE+, and DUAL only.

Copyright © 2023, Brooks Automation, Inc.

In a scenario where more than one wafer is present in a slot or during multiple scans of a wafer, the controller records different values of the wafer during each scan then calculate the average value from each reading.

Request Thickness	
Command Syntax	`RQ THICKNESS`

Response Syntax	**In Monitor mode:**
	
	`RQ THICKNESS`
	`STN: station`
	`TOTAL SLOTS: numSlots`
	`Slot: 1: thickness`
	
	In PKT mode:
	
	`RQ THICKNESS`
	`thickness 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00`

Arguments	None
Description	Requests the thickness of each wafer in a slot.
Example	Single Wafer1 present in a Slot 1 of width 1000. In Single Mode controller detects Wafer1 width of 1000. The RQ THICKNESS command will display “1000.” In SINGLEPLUS mode: Scan 1 reads Wafer1 width of 1000. Scan 2 reads Wafer1 width of 1050. RQ THICKNESS command will display “1025.” In DUAL mode Scan 1 reads Wafer1 width of 1000 Scan 2 reads Wafer1 width of 1100 RQ THICKNESS command will display “1050.” Two wafers present in a Slot 1 of width 1000 and 1100. In Single Mode controller detects Wafer1 width of 1000 and Second Wafer2 of width 1100. RQ THICKNESS command will display “1050.” In SINGLEPLUS mode Scan 1 reads Wafer1 width of 1000. Scan 2 reads Wafer1 width of 1040. Scan1 reads Wafer2 width of 1100. Scan2 reads Wafer2 width of 1000. RQ THICKNESS command will display average of all readings 1035. In DUAL mode Scan 1 reads Wafer1 width of 1000. Scan 2 reads Wafer1 width of 1040. Scan1 reads Wafer2 width of 1100. Scan2 reads Wafer2 width of 1000. RQ THICKNESS command will display average of all readings 1035.

Copyright © 2023, Brooks Automation, Inc.

14. Aligners

This chapter provides commands used to set up and operate various aligners. For complete information on using the aligner, see the specific Aligner User Manual.

Overview

Allows direct communication with the aligner through the robot via the secondary serial port or Linked Device.

> ***NOTE:*** Commands may be passed through the aligner using the ALIGNER prefix (shown here); however, to avoid aligner mismatch problems, it is recommended the SET ALIGNER command be used instead to SET and STORE parameters. This assures the aligner and the robot will have the same parameters.

Table 14-1: Action Commands

Command	Command String	Description
Align	ALGN	Scans a wafer and post-positions it to a user-specified position.
Cold_Start	COLD_START	Transfers all default parameters from the PROM to the EEPROM.
Home	HOME	References rotate servo.
Motor	MOTOR	Removes or returns power to the Theta axis.
Move Relative	MOVT REL	Moves the rotate axis a specified number of degrees.
Reset	RSET	Resets all parameters to those stored in the EEPROM and causes the aligner to HOME.
Scan	SCAN	Performs a single scan characteriza- tion of wafer.

Copyright © 2023, Brooks Automation, Inc.

Table 14-2: Request Commands

Command	Command String	Description
Display Environment	DENV	Returns information on the calibra- tion factor, maximum allowable eccentricity, and overtravel distance.
Edge	EDGE	Reports in mils the location of the wafer shadow.
Flash	FLSH	Reports in “pixels” the location of the wafer shadow.
Futn	FUTN	Gives a continuous reading in mils of the location of the wafer shadow.
Request CCD (Number)	RQCD	Returns the active CCD number.
Request CCD (Position)	RQCCDPOS	Returns the (virtual) positions of up to three installed CCDs.
Request Device ID	RQID	Returns the device identification number.
Request Firmware Revision	RVSN	Returns the firmware revision level.
Request Post Position	RQPS	Returns information on post posi- tions.
Request Serial I/O Mode	RQIO	Returns serial I/O settings.
Request Wafer Type	RQWF	Returns information on wafer type selected.
Result	RSLT	Returns the results of the last SCAN or ALGN.

Table 14-3: Set Commands

Command	Command String	Description
Set Calibra- tion Factor	LDCL	Sets the calibration factor desired.
Set CCD Position	LDCCDPOS	Sets the position of up to three installed CCDs.
Set Overtravel Value	LDOT	Sets the overtravel value desired.
Set Post Position (Location)	LDPS	Sets up to four post position loca- tions.
Set Post Position (Number)	SLPS	Sets post position as number 1, 2, 3, or 4.
Set Serial I/O Mode	SLIO	Sets serial I/O mode.
Set Wafer Type	SLWF	Sets wafer type.

Copyright © 2023, Brooks Automation, Inc.

Table 14-4: Store Commands

Command	Command String	Description
Store Baud Rate	SVBD	Stores the current baud rate to the EEPROM.
Store Calibration Factor	SVCL	Stores the current calibration factor to the EEPROM.
Store CCD (Number)	SVCD	Stores the index of the active CCD to the EEPROM.
Store CCD (Position)	SVCCDPOS	Stores the current settings for the virtual positions of up to three CCD arrays to the EEPROM.
Store Com- munication Mode	SVMB	Stores the current communications mode to the EEPROM.
Store Echo Mode	SVEC	Stores the current echo mode to the EEPROM.
Store Fiducial Type	SVFD	Stores the current wafer fiducial type to the EEPROM.
Store Overtravel Value	SVOT	Stores the current overtravel value to the EEPROM.
Store Post Position (Location)	SVPS	Stores the four current post position locations to the EEPROM.
Store Post Position (Number)	SVCP	Stores the index of the selected post position to the EEPROM.
Store Wafer Size	SVWS	Stores the current wafer size to the EEPROM.

Aligner	
Command Syntax	`ALIGNER` *`identifier command`*
Response Syntax	There is no response to this command.
Arguments	identifier: Identifier for the specific aligner. Range = 1 - 4 Default = 0 If the index is 0 or is omitted, the local aligner will be used. command: A valid aligner SET, STORE, or Action command. See the Aligner User Manual for commands.
Description	This command may be used to send information to the aligner using the aligner command through the robot. If multiple remote aligners are present, the index identifier must be used to command the specific aligner.

Copyright © 2023, Brooks Automation, Inc.

Aligner	

Example	The following example will home the local aligner attached to the robot controller:
	
	`ALIGNER HOME`
	`ALIGNER SCAN`
	`ALIGNER ALGN`
	
	The following example requests the identification of the local aligner attached to the robot controller:
	
	`RQ ALIGNER RQID`

Robot Setup for 1 Aligner and 1 Robot

The following commands set up a one-aligner/one-robot network where the robot has one aligner attached to it on the auxiliary serial port (COM 2). The aligner is slaved to the robot in which it is physically connected.

a. Define the communication link that will be used for Robot 1.

See "Configure/Request Link" on page 313.

```
CONFIG LINK 4 TYPE USER_CMD PORT SERIAL PARA 9600 N 8 1
```

b. Create the **Device Table** and set the device to a communication link.

See "Set/Request/Store Device ID Link" on page 317.

```
SET DEVICE 1 LINK 4
```

Store the device identifier.

```
STORE DEVICE 1
```

c. Reboot the robot for the changes to take effect.

```
RESET
```

d. Go to "Aligner Setup" on page 319 for setting up the aligner.

Copyright © 2023, Brooks Automation, Inc.

Configure/Request Link

Configures and requests the link identifier for each port type and link type.

The LINK feature allows generic links to be established using Ethernet or RS-232 Serial connection between the host and other controllers and slave devices. This command configures external links to the LEAP Controller, which include Ethernet (LAN), DeviceNet, and Serial Ports. It is often used in conjunction with DeviceNet setup, but links may be set up without a *Device* association.

Internally, the *Link Number* is the same as the port number, such that the configuration of links is somewhat defined by the port. If the defined link is used by any other internal process, it will be automatically redefined. The settings for any *Link* or *Device* becomes active after power-up. Consequently after changing any link settings, the robot must be rebooted for the changes to take effect.

To remove a Link, set the TYPE to NONE.

Configure Link	

Command Syntax	`CONFIG LINK link_number TYPE (RAW_TEXT|USER_CMD|USER_DEV|HOST_DEV|FIXLOAD)|(NONE)]`
	`[PORT SERIAL PARA` *`baud parity dataBits stopBits`*`]`
	`[PORT ENET IP` *`ipAddress`* `RMTLNK` *`rmtLinkNo`*`]`
	`[PORT DNETSGW4` *`deviceName`* `CHANNEL channel PARA` *`baud parity dataBits stopBits`*

Copyright © 2023, Brooks Automation, Inc.

Configure Link	

Response Syntax	In Monitor Mode for Link TYPE SERIAL
	`LINK link_number`
	`TYPE ----- (RAW_TEXT|USER_CMD|USER_DEV|HOST_DEV|FIXLOAD|UNKNOWN) PORT ---- SERIAL`
	`BAUD ----- baud`
	`PARITY --- parity`
	`DATA ----- dataBits`
	`STOP ----- stopBits`
	
	In Packet Mode for Link TYPE SERIAL
	`LINK link_number TYPE (RAW_TEXT|USER_CMD|USER_DEV|HOST_DEV|FIXLOAD|UNKNOWN) PORT SERIAL PARA baud parity dataBits stopBits`
	
	In Monitor Mode for Link TYPE ENET
	`LINK link_number`
	`TYPE ----- (RAW_TEXT|USER_CMD|USER_DEV|HOST_DEV|FIXLOAD|UNKNOWN)`
	`PORT ----- ENET`
	`IP ------- ipAddress`
	`RMTLNK --- rmtLinkNo`
	
	In Packet Mode for Link TYPE ENET
	`LINK link_number`
	`TYPE(RAW_TEXT|USER_CMD|USER_DEV|HOST_DEV|FIXLOAD|UNKNOWN)`
	`PORT ENET IP ipAddress RMTLNK rmtLinkNo`
	
	In Monitor Mode for Link TYPE FIXLOAD
	`LINK link_number`
	`TYPE ------ (RAW_TEXT|USER_CMD|USER_DEV|HOST_DEV|FIXLOAD|UNKNOWN) PORT ------ DNETSGW4`
	`NAME ------ name`
	`CHANNEL -- channel`
	`BAUD ------ baud`
	`PARITY ---- parity`
	`DATA ------- dataBits`
	`STOP ------- stopBits`
	
	In Packet Mode for Link TYPE FIXLOAD
	`LINK link_number TYPE (RAW_TEXT|USER_CMD|USER_DEV|HOST_DEV|FIXLOAD|UNKNOWN) PORT DNETSGW4 name CHANNEL channel PARA baud parity dataBits stopBits`

Copyright © 2023, Brooks Automation, Inc.

Configure Link		
Arguments	link_number:	Link number Range = 1-19 ***NOTE:*** *Configuring links 1-5 may cause the robot to stop responding. Attempting to configure links 0, 2, and 3 will report an error. Attempting to configure a serial link with a number greater than 5 will report an error.*
	TYPE:	`RAW_TEXT`: Serial messages pass-through to the auxiliary RS-232 port using COMn prefix `USER_CMD`: Allows communication with Aligner or another robot in Packet Mode `USER_DEV`: Robot link to execute command on other device `HOST_DEV`: Robot link to carry out request from other device `FIXLOAD`: Robot link to FixLoad 6M Load Port Module
	PORT:	`SERIAL`: RS-232 connection `ENET`: Ethernet connection *parameters: IP ppp.ppp.ppp.ppp RMTLNK LnkNo* `DNETSGW4`: FixLoad 6M Connection (SI30 to SI39) `deviceName`: device name for serial gateway device with 4 serial ports
	PARA:	`baud`: (9600|19200) `parity`: (N|E|O) `N`: none `E`: even `O`: odd `dataBits`:(8|7) `stopBits`: (1|2)
	CHANNEL:	`channelID`: channel number used in 4 port device
Description	Configures the link identifier for each port type and link type.	

Example	`CONFIG LINK 4 TYPE USER_CMD PORT SERIAL PARA 9600 N 8 1`
	Link 4 is set at the factory for communication with a single, integrated aligner.
	`CONFIG LINK 6 TYPE USER_CMD PORT SERIAL PARA 9600 N 8 1`
	`CONFIG LINK 11 TYPE HOST_DEV PORT ENET IP 20.20.249.2 RMTLNK 0`
	`CONFIG LINK 12 TYPE USER_DEV PORT ENET IP 20.20.249.3 RMTLNK 11`	

Request Configure Link	
Command Syntax	`RQ CONFIG LINK (ALL|`*`link_number`*`) <CURRENT>`

Response Syntax	In Monitor Mode:
	`RQ CONFIG LINK ALL`
	`PORT ----- SERIAL`
	`BAUD ------ baud`
	`PARITY ---- parity`
	`DATA ------ dataBits`
	`STOP ------ stopBits`

Arguments	ALL | link_number

Copyright © 2023, Brooks Automation, Inc.

Request Configure Link	
Description	Requests the link identifier for each port type and link type.
Example	`RQ CONFIG LINK ALL`

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Device ID Link

Sets, requests, and stores the Device Table for Multiple Device mode.

This command sets the Device ID for each device used in the Multiple Device mode. This identifier will be used to command the specific device.

The link defines the device that the commands must be forwarded to from this robot. The device that the link attaches to is not necessarily the final destination for the message. The recipient on the other end of the link may have another link in its table for the identified device. The message will be continuously forwarded until the appropriate device is reached. For the current device, the link number is 0.

Set Device ID Link		
Command Syntax	`SET DEVICE deviceId LINK linkId`	
Response Syntax	There is no response to this command.	
Arguments	deviceId:	Two-digit device identifier Default = 00
	linkId:	Two-digit link identifier Default = 00
Description	Sets the Device Table for Multiple Device mode.	
Example	`SET DEVICE 01 LINK 00`	

Request Device ID Link		
Command Syntax	`RQ DEVICE deviceId (linkId <LINK>|ALL`	
Response Syntax	`DEVICE deviceId LINK linkId`	
Arguments	deviceId:	Two-digit device identifier Default = 00
	linkId:	Two-digit link identifier Default = 00
Description	Requests the Device Table for Multiple Device mode.	
Example	`RQ DEVICE 01 LINK 00` `DEVICE 01 LINK --`	

Copyright © 2023, Brooks Automation, Inc.

Store Device ID Link		
Command Syntax	`STORE DEVICE deviceId (linkId <LINK>|ALL)`	
Response Syntax	There is no response to this command.	
Arguments	deviceId:	Two-digit device identifier. Default = 00
	linkId:	Two-digit link identifier. Default = 00
Description	Stores the Device Table for Multiple Device mode.	
Example	`STORE DEVICE 01 ALL`	

Copyright © 2023, Brooks Automation, Inc.

Aligner Setup

Depending on whether the aligner is directly connected to the robot (local aligner) or if the aligner is connected through the Linked Device (remote aligner), will determine the setup procedure for the aligner.

Local Aligner Setup

This feature allows serial commands to pass through the robot to control the aligner through the serial communications SIO2 port located on the controller. The robot is connected directly to the aligner through the serial port. The robot checks the aligner configuration at start-up and has the ability to align material with integrated robot/aligner commands. The following section demonstrates the setup of an aligner as a peripheral device. The communication link must be established before beginning this procedure.

a. **Set Basic Aligner Parameters**

First, set up the aligner for communication and configuration.

```
SET ALIGNER 1 STN 1 TYPE type DEV 01
```

When setting up the aligner for the first time, the command must include the station number, aligner type, and the device identifier.

```
STORE ALIGNER 1 STN TYPE DEV
```

Verify the aligner parameters have been set correctly:

```
RQ ALIGNER 1 STN
```

Set Information Aligner Parameters

In this example, the aligner is set for round, 300 mm wafers, notched fiducial, opaque, variety type 1. See Table 14-5: "Wafer Identifier" on page 325.

```
SET ALIGNER 1 WAFERID RO-300-NT-O-01
STORE ALIGNER 1 STN DEV WAFERID
```

Verify the aligner information is correct:

```
RQ ALIGNER 1 ALL
```

Copyright © 2023, Brooks Automation, Inc.

b. **Determining the CCD and Post Position**

Establish the coordinate system of the aligner with the robot. This is done by setting a CCD position where a center-line can be drawn through the robot, the aligner's chuck, and the fiducial of the wafer as shown in Figure 14-1.

The CCD position that aligns the robot and aligner coordinate system will be stored on the robot.

The aligner will have the CCD position set to 0°. **The Post Position will be relative to the physical CCD** and is the origin of the aligner coordinate system.

In the following example, the CCD and PostPositions are set using the calculated values shown:

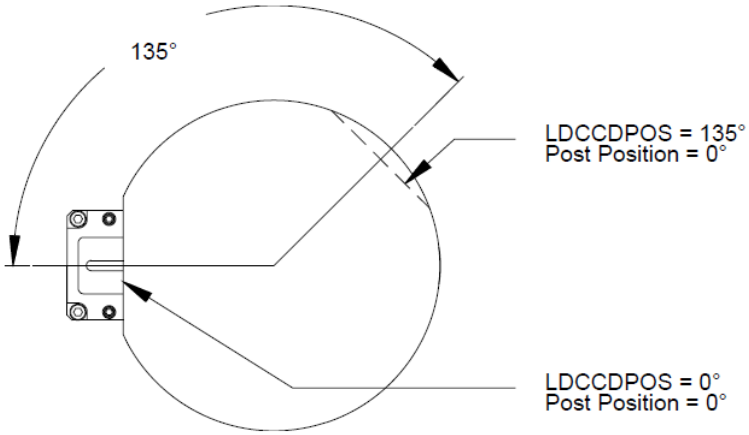


Figure 14-1: Determining CCD Position

```
SET ALIGNER 1 CCDPOS 98000
STORE ALIGNER 1 CCDPOS

SET ALIGNER 1 POSTPOS 1 POS 82000
STORE ALIGNER 1 POSTPOS
```

To verify the settings, enter the following: `RQ ALIGNER 1 ALL`

Copyright © 2023, Brooks Automation, Inc.

Operate the Aligner using the Integrated Commands

Integrated robot/aligner commands perform coordinated sequences by selecting the appropriate options as described below:

ALGN:	Command may be added to the end of the PICK|PLACE|XFER commands with the following results:
	
	XFER: after PICK and prior to PLACE, material is aligned with the aligner.
	example:
	`XFER 1 2 ALGN`
	(aligner is at station 3 and is not specified)
	
	PICK: wafer is aligned before PICK from aligner.
	example:
	`PICK 3 ALGN`
	(aligner is at station 3)
	
	PLACE: after PLACE to aligner, wafer is aligned.
	example:
	`PLACE 3 ALGN`
	(aligner is at station 3)
	
	NOTE: *With PLACE, the robot indicates completion prior to alignment completion. Additional moves to aligner station are prevented until the alignment is complete.*

PALGN:	May be added to the PICK command with the following results: **PICK:** wafer is Picked using the last available alignment information from the aligner. `PICK 3 PALGN` (aligner is at station 3) In this example, the wafer is not aligned. Information from the previous alignment is used to PICK from the aligner. This command is used only if a successful ALGN took place prior to using this command. The PICK PALGN command can be performed from aligner station. The PICK PALGN command will use the data of the last align command to adjust the PICK position.
ADJ_STN:	May be added to the PICK command with the following results: **PICK:**wafer alignment is used to adjust the stored position information of a station. `PICK 1 ADJ_STN` (aligner station is not specified) In this example, the wafer is Picked from Station 1 and placed on the aligner. The aligner performs a SCAN, examines the results and applies them to adjust the Station 1 Stored position information.

Copyright © 2023, Brooks Automation, Inc.

Integrated Error Reporting from the Aligner

Aligner errors are reported directly from the aligner. The robot displays the aligner errors in a special format. The errors reported from the aligner will begin with the following prefix: ALGN_ERR. For example, the aligner reports error code 41; the robot will report this error as ALGN_ERR 0041. Aligner errors codes are located in the Aligner User Manual.

Wafer Presence Detection on the Aligner

The aligner will check for the presence of a wafer and if a wafer is detected, will not allow the robot to place a wafer to the aligner during PLACE, PLACE ALGN, PICK ADJ_STN, XFER, and XFER ALGN commands.

Set/Request/Store Aligner

Sets, requests, and stores various station-related parameters for the aligner station.

Commands may be passed through the aligner using the ALIGNER prefix; however, to avoid aligner mismatch problems, it is recommended the SET ALIGNER command be used instead. This assures the aligner and the robot will have the same parameters.

The station parameter assigns the station number for the aligner. The TYPE option must be entered before the Device. If the correct order is not followed, an Unknown Command or a Communications error will be reported. The TYPE cannot be changed unless the aligner is set to 0 and SET ALIGNER STN TYPE DEV re-entered.

The XFER option allows the user to select whether the robot retracts away from the aligner while it is performing the align function during the XFER station station ALGN command. Using the EX parameter, the robot remains extended at the station during alignment, thereby saving throughput time. However, some end effector designs interfere with the align process. In this case, the RE parameter moves the robot away from the aligner during alignment.

The Wafer ID is a code used to identify the wafers. The format is as follows:

```
SHAPE-SIZE-FIDUCIAL-TYPE-OTHER
```

For example, `RO-200-NT-0-01` indicates a Round, 200mm wafer with a notch, opaque, and variety.

Copyright © 2023, Brooks Automation, Inc.

Set Aligner	
Command Syntax	SET ALIGNER identifier STN station [TYPE (DEFAULT|EDGE_ALGN)] [DEV deviceId] [[XFER (EX|RE)] [CCDNUM ccdNum] [CCDPOS ccdAngle] [WAFERID waferId] [POSTPOS (1|2|3|4) POS ppAngle] [POSTPOS INDEX (1|2|3|4)] [MAXANGLE maxAngle] [HANDOFF INDEX handOffIndex] [HANDOFF POS (1|2|3|4)] [CCD_CORR (ON|OFF)] [COORD (AWAY FROM ROBOT|TOWARD ROBOT)] [FID_CORR (ON|OFF)]
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Set Aligner		
Arguments	identifier:	Identifier for the specific aligner Range = 1 - 4 Default = 0 If the index is 0 or is omitted, the local aligner will be used.
	station:	Station number for the aligner
	TYPE:	Aligner model `DEFAULT`: all other types of aligners
	DEV:	`deviceId`: Device identifier from the Device Table Default = 0
	XFER:	*This command is used with the default aligner.* Sets where the robot will stay during the align. `EX`: Robot will stay extended during alignment. `RE`: Robot will retract during alignment.
	CCDNUM:	*This command can be used with any aligner.* `ccdNum`: CCD number `1`: 200mm `2`: 300mm
	CCDPOS:	*This command can be used with any aligner.* `ccdAngle`: Two-digit device identifier Default = 00
	WAFERID:	This command can be used with any aligner. `wafer_id`: Code used to set the wafer identifier
	POSTPOS:	*This command can be used with any aligner.* The Post Position number at the aligner: 1, 2, 3, or 4 `POS ppAngle`: The angle of the post position relative to the CCD Range = 0 - 360000 millidegrees
	CCD_CORR:	*This command is for legacy WAVE I compatibility.* Enabling this feature will cause the CCD Correction on the local aligner to behave in a manner consistent with WAVE I while running from within a WAVE II environment. `ON`: Correction will work in WAVE I mode. `OFF`: Correction is not on.
	COORD:	*This command can be used with any aligner.* **AWAY FROM ROBOT:** When the Post Position is set to 0° in relation to the robot, the fiducial points away from the robot (default). **TOWARD THE ROBOT:** When the Post Position is set to 0°. the fiducial points toward the robot.

Set Aligner	
Description	Sets various station-related parameters for the aligner station.

Copyright © 2023, Brooks Automation, Inc.

Set Aligner	
Example	SET ALIGNER 1 STN 3 DEV 04 SET ALIGNER 1 WAFERID RO-200-FL-O-01 XFER EX SET ALIGNER CCDPOS 98000 POSTPOS 1 POS 82000

Table 14-5: Wafer Identifier

Shape	Size	Fiducial	Type	Other
RO = Round RT = Rectangle or Square	300 200 150 size in mm	NT = Notch SF = Single Flat FS = Featureless UN = Unknown FL = Flat	O = Opaque	Number indicating a variety of wafers if more than one type falls in the same category.

NOTE: Not all aligners support all wafer types.

Request Aligner	
Command Syntax	RQ ALIGNER identifier [ALL|{(STN|TYPE||DEV| XFER|CCDNUM|CCDPOS|WAFERID|POSTPOS|MAXANGLE| HANDOFF INDEX|CCD_CORR|COORD|VERSION|FID_CORR)}] RQ ALIGNER identifier DEBUG ***NOTE:*** *The DEBUG option reports the aligner values along with the robot values for the aligner. This command is used to diagnose robot-aligner integration mismatch issues.*
Response Syntax	ID: *id* MODE: *mode*
Arguments	
Description	Requests various station-related parameters for the aligner station.
Example	RQ ALIGNER 1 DEBUG ID: 1 MODE: MULTI_ROBOT

Store Aligner	
Command Syntax	STORE ALIGNER *identifier* [ALL|{(STN|TYPE|DEV| XFER|CCDNUM|CCDPOS|WAFERID|POSTPOS|MAXANGLE| HANDOFF INDEX|CCD_CORR|COORD|VERSION|FID_CORR)}]
Resonse Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Store Aligner	
Arguments	
Description	Stores various station-related parameters for the aligner station.
Example	`STORE ALIGNER 1 ALL`

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Aligner Mode

Sets, requests, and stores the robot to support WAVE mode (single aligner in system) or WAVE II (multiple aligners in system).

The data storage and usage location of aligner information is based on the mode set. Upgrading or switching between SINGLE_ROBOT and MULTI_ROBOT modes will keep all aligner information and support most commands for each mode.

The SINGLE_ROBOT mode allows only one aligner in the system and the aligner must be a local aligner. In SINGLE_ROBOT mode, parameters can be set directly to the aligner; however, it is recommended that you use the SET and STORE ALIGNER commands to prevent mismatch errors.

The data storage and usage location of aligner information is based on the mode set. Upgrading or switching between SINGLE_ROBOT and MULTI_ROBOT modes will keep all aligner information and support most commands for each mode.

The SINGLE_ROBOT mode allows only one aligner in the system and the aligner must be a local aligner. In SINGLE_ROBOT mode, parameters can be set directly to the aligner; however, it is recommended that you use the SET and STORE ALIGNER commands to prevent mismatch errors.

The data storage and usage location of aligner information is based on the mode set. Upgrading or switching between SINGLE_ROBOT and MULTI_ROBOT modes will keep all aligner information and support most commands for each mode.

The SINGLE_ROBOT mode allows only one aligner in the system and the aligner must be a local aligner. In SINGLE_ROBOT mode, parameters can be set directly to the aligner; however, it is recommended that you use the SET and STORE ALIGNER commands to prevent mismatch errors.

Set Aligner Mode	
Command Syntax	`SET ALIGNER` *`identifier`* `MODE` *`mode`*

Response Syntax	In Monitor Mode:
	`MODE .....(SINGLE_ROBOT|MULTI_ROBOT)`
	
	In Packet Mode:
	`ALIGNER aligner (SINGLE_ROBOT|MULTI_ROBOT)`

Copyright © 2023, Brooks Automation, Inc.

Set Aligner Mode		
Arguments	identifier:	Identifier for the specific aligner Range = 1 - 4 Default = 0 If the index is 0 or is omitted, the local aligner will be used.
	mode:	Operating mode of Aligner/Robot interface **SINGLE_ROBOT:** Single, local aligner in system **MULTI_ROBOT:** Multiple, remote aligners in system
Description	Sets the robot to support WAVE mode (single aligner in system) or WAVE II (multiple aligners in system).	
Example	`SET ALIGNER 2 MODE SINGLE_ROBOT`	

Request Aligner Mode		
Command Syntax	`RQ ALIGNER` *identifier* `MODE`	

Response Syntax	In Monitor Mode:
	`MODE .....(SINGLE_ROBOT|MULTI_ROBOT)`
	
	In Packet Mode:
	`ALIGNER aligner (SINGLE_ROBOT|MULTI_ROBOT)`	

Arguments	identifier:	Identifier for the specific aligner Range = 1 - 4 Default = 0 If the index is 0 or is omitted, the local aligner will be used.
	mode:	Operating mode of Aligner/Robot interface **SINGLE_ROBOT:** Single, local aligner in system **MULTI_ROBOT:** Multiple, remote aligners in system
Description	Requests the robot to support WAVE mode (single aligner in system) or WAVE II (multiple aligners in system).	
Example	`RQ ALIGNER 2 MODE`	

Store Aligner Mode	
Command Syntax	`STORE ALIGNER` *identifier* `MODE`

Copyright © 2023, Brooks Automation, Inc.

Store Aligner Mode		

Response Syntax	In Packet Mode:
	`ALIGNER aligner (SINGLE_ROBOT|MULTI_ROBOT)`
	
	In Monitor Mode:
	`MODE .....(SINGLE_ROBOT|MULTI_ROBOT)`	

Arguments	identifier:	Identifier for the specific aligner Range = 1 - 4 Default = 0 If the index is 0 or is omitted, the local aligner will be used.
	mode:	Operating mode of Aligner/Robot interface **`SINGLE_ROBOT:`** Single, local aligner in system **`MULTI_ROBOT:`** Multiple, remote aligners in system
Description	Stores the robot to support WAVE mode (single aligner in system) or WAVE II (multiple aligners in system).	
Example	`STORE ALIGNER 1 MODE`	

Copyright © 2023, Brooks Automation, Inc.

Request Align Offsets

Requests the R and T offsets after an ALGN command.

Reports the radial and theta offsets calculated after a wafer alignment command to the aligner. The robot calculates the offsets and converts them to robot coordinates. These values may be used when Multi-Point Paths are implemented.

Request Align Offsets		
Command Syntax	`RQ ALGN alignerId OFFSETS [<ARM> (A|B)]`	

Response Syntax	Monitor Mode:
	`RQ ALGN alignerID OFFSETS`
	`RO Offset: offset_value`
	`TO Offset: offset_value`
	
	Packet Mode:
	`RO offset_value TO offset_value`	

Arguments	alignerID:
	
	ARM:
	
	offset_value:
		Two-digit device identifier
		Default = 00
		
		Arm descriptor
		
		Radial and Theta Offsets
		`RO OFFSET:`
		`TO OFFSET:`

Description	Requests the R and T offsets after an ALGN command.	
Example	`RQ ALGN 1 OFFSETS` `RO -325 TO -43`	

Copyright © 2023, Brooks Automation, Inc.

Halt Aligner

NOTICE **NOTICE**

While no damage to the equipment will occur due to a HALT command, the HALT command may result in abrupt motions, and may cause the misalignment of a wafer.

Immediately halts all motion operations.

A controlled stop to the aligner is applied to halt all motion while minimizing wafer movement. Referencing is maintained after a HALT; all axes that were referenced before the HALT will still be referenced after the HALT. A “ready” response will be returned when the halt action is complete.

This command can also be activated using `<CTRL-C>` on the keyboard.

Halt Aligner		
Command Syntax	`HALT ALIGNER (`*`deviceId`*`)`	
Response Syntax	There is no response to this command.	
Arguments	deviceID:	Two-digit device identifier Default=00
Description	Immediately halts all motion operations.	
Example	The following example stops all current movement of the aligner: `HALT ALIGNER 1`	

Copyright © 2023, Brooks Automation, Inc.

Set/Request/Store Interlock

Sets, requests, and stores interlocks to avoid wafer collisions when using an aligner.

`ALIGNER WFR_PRS`: This Wafer Present Interlock is used to check if a wafer is on the aligner before attempting to place a wafer at the aligner station.

Set Interlock		
Command Syntax	`SET INTLCK [ALIGNER WFR_PRS (Y|N)]`	

Response Syntax	In Monitor Mode:
	`Interlock`
	`Aligner Wfr_Prs ------ (Y|N)`
	
	In Packet Mode:
	`INTLCK ALGN_WFR_PRS (Y|N)`	

Arguments	ALIGNER WFR_POS:	Checks for wafer presence on the aligner using CCD. Y: Robot will check for wafer presence on the aligner. N: Robot will not check for wafer presence.
Description	Sets interlocks to avoid wafer collisions when using an aligner.	
Example	`SET INTLCK ALGN 300_WFR_PRS Y` `SET INTLCK HOME WFR_PRS Y`	

Request Interlock		
Command Syntax	`RQ INTLCK [ALIGNER WFR_PRS]`	

Response Syntax	In Monitor Mode:
	`Interlock`
	`Aligner Wfr_Prs ------ (Y|N)`
	
	In Packet Mode:
	`INTLCK ALGN_WFR_PRS (Y|N)`	

Arguments	ALIGNER WFR_POS:	Check for wafer presence on the aligner using CCD. Y: Robot will check for wafer presence on the aligner. N: Robot will not check for wafer presence.
Description	Requests interlocks to avoid wafer collisions when using an aligner.	
Example	`RQ INTLCK ALIGNER WFR_PRS` `INTLCK ALGN_WFR_PRS N`	

Copyright © 2023, Brooks Automation, Inc.

Store Interlock		
Command Syntax	`STORE INTLCK [ALIGNER WFR_PRS]`	

Response Syntax	In Monitor Mode:
	`Interlock`
	`Aligner Wfr_Prs ------ (Y|N)`
	
	In Packet Mode:
	`INTLCK`
	`ALGN_WFR_PRS (Y|N)`	

Arguments	ALIGNER WFR_POS:	Check for wafer presence on the aligner using CCD. Y: Robot will check for wafer presence on the aligner. N: Robot will not check for wafer presence.
Description	Stores interlocks to avoid wafer collisions when using an aligner.	
Example	`STORE INTLCK ALIGNER WFR_PRS`	

Copyright © 2023, Brooks Automation, Inc.

Pick Aligner

Causes the robot arm to pick a wafer from a specified station and slot number.

Pick Aligner		
Command Syntax:	`PICK station [<ARM> (A|B|AB)] [ALGN <palgnId> <SLPS postPosNum>] [PALGN <palgnId>] [ADJ_STN]`	
Response	There is no response to this command.	
Arguments	STATION:	Station number
	ARM:	Arm descriptor *Arm AB is used with Radius armsets only.*
	ALGN:	Wafer is aligned before PICK from aligner.
	PALGN:	Wafer is PICKed using the last available alignment information from the aligner.
	postPosNum:	Post Position number
	ADJ_STN:	Wafer alignment is used to adjust the stored position information of a station.
Description	Causes the robot arm to pick a wafer from a specified station and slot number.	
Example	`PICK 1 PALGN`	

Copyright © 2023, Brooks Automation, Inc.

Place Aligner

Causes the robot arm to place a wafer at a specified aligner station and then perform an align.

Place Aligner		
Command Syntax	PLACE *station* [<ARM> (A|B|AB)] [ALGN <*palgnId*> <SLPS *postPosNum*>]	
Response Syntax	There is no response to this command.	
Arguments	station:	Station number
	ARM:	Arm descriptor *Arm AB is used with Radius armsets only.*
	ALGN:	Place the wafer at the aligner station and perform alignment.
	postPosNum:	Post Position number
Description	Causes the robot arm to pick a wafer from a specified station and slot number.	
Example	PLACE 1 ALGN PLACE 2 B ALGN	

Transfer Aligner (XFER)

Transfers a wafer from one specified station to another specified station with alignment in between transfer.

For a normal transfer, this function picks the wafer from one station, retracts, rotates, and places it to the aligner. After the alignment, the robot picks the wafer from the aligner centered on the end effector based on aligner data, and places the wafer at the next station.

Transfer Aligner (XFER)	
Command Syntax	XFER [<ARM> (A|B|AB)] pickStnNum [SLOT pkSlotNum] placeStnNum [SLOT plSlotNum] [ALGN *alignerId* (SLPS *postPosNum*)]
Response Syntax	There is no response to this command.

Copyright © 2023, Brooks Automation, Inc.

Transfer Aligner (XFER)		
Arguments	ARM:	Arm descriptor Arm AB is used with Radius armsets only.
	pickStnNum:	Station number for PICK operation
	placeStnNum:	Station number for PLACE operation
	pkSlotNum:	Slot number for PICK operation
	ALGN:	Slot number for PLACE operation
	SLPS:	After picking the wafer from the station, align the wafer and place the wafer at the station.
Description	Causes the robot arm to pick a wafer from a specified station and slot number.	
Example	`XFER 1 6 ALGN 3` In this example, the robot picks the wafer from Station 1 and places it to the aligner. The aligner examines the wafer and reports the data to the robot. The robot performs a delta pick from the aligner and places the wafer at Station 6.	

Copyright © 2023, Brooks Automation, Inc.

Appendix A. Wave to LEAP Migration Issues

a. Communication FLOW Options on LEAP: Wave supported five modes of interaction: Sequential, Background (BKG), Background Plus (BKG+), Multi-mode, and Multi-Device. However, LEAP only supports the first four modes and does not support Multi-Device because Multi-Device mode is not widely used for vacuum robots.

b. Early versions of LEAP software would automatically recover after a hard track error upon the next valid motion command. From Version 3.0.1.6 and higher, LEAP mimics the behavior of Wave after an error. After a hard-track or other motion error, any motion command other than REF, SET SERVO ON, or HOME will report an error. Only after a successful REF, SET SERVO ON, or HOME command will other motion commands resume working without an error.

c. If the robot gets a servo controller error during motion that causes the robot to servo off, the robot will require a HOME or REF command to turn servos back on again. This prevents host software from continuing robot motion until the error is addressed.

d. The LEAP SET STN command has been expanded to allow the user to define station values and options for up to 99 stations. In WAVE, the number of stations remains at 50.

e. In WAVE, a new station has the parameter NSLOT set to 0. In LEAP, a new station has NSLOT set to 1. Both mean the same thing – the station has a single slot.

f. Behavior of robot when VIA is defined within retract circle:

   In WAVE, the VIA is ignored and the station is treated as a radial station.

   In LEAP, when the user attempts to access the station, an error is reported and the station is ignored when homing.

g. Behavior when station VIA configured but CURVE_MOVE not enabled.
   a. If VIA positions (R, T, DIST) are configured for a station but CURVE_MOVE is not enabled, the following events occur when the user attempts to access the station (GOTO, PICK, PLACE):
      i. In WAVE, an error is reported.
      ii. In LEAP, the robot will follow the radial path into the station, which allows users to more easily enable/disable the VIA path if desired, without having to CLEAR the VIA to access the station using the radial path and then restore the VIA settings (R, T, DIST) to access it using the VIA path.

Copyright © 2023, Brooks Automation, Inc.

h. Behavior of REMOVE IO when removing a non-existing I/O map item.
    a. In WAVE, an error is reported (error code 1002) if the I/O map item provided for REMOVE IO command was not currently mapped.
    b. In LEAP, the I/O item is removed if it exists and nothing happens if it does not exist.

i. ONE_Z_MOVE and 3_ZPOS_STN settings.
    a. In WAVE, once either the ONE_Z_MOVE option or the 3_ZPOS_STN option are assigned 'Y' for a station, the other option cannot be assigned at all. The software reports an error if an assignment is attempted.
    b. In LEAP, once either the ONE_Z_MOVE option or the 3_ZPOS_STN option are assigned to 'Y' for a station, the other option cannot be assigned to 'Y'. The software reports an error if this is attempted, but it can be assigned to 'N' without an error.

j. Workspace feature MTN_Intlck is not supported on LEAP.

Copyright © 2023, Brooks Automation, Inc.

Appendix B. I/O Input/Output Examples

ECHO - Example of a single discrete input signal

```
MAP TEST_IN_1 DISCRETE_IN LOW TO DIGITAL_IN 0x1
_RDY

RQ IO MAP ALL
IO MAP TEST_IN_1 DISCRETE_IN LOW DIGITAL_IN 0x00000001
IO MAP dio_in NUMERIC_IN HI DIGITAL_IN 0x203fffff
IO MAP dio_out NUMERIC_OUT HI DIGITAL_OUT 0x000fffff
IO MAP hw_emer_stop EMER_STOP LOW DIGITAL_IN 0x20000000
_RDY

RQ IO STATE TEST_IN_1
IO STATE TEST_IN_1 INACTIVE
_RDY
```

ECHO - Example of a single discrete output signal

```
MAP TEST_OUT_1 DISCRETE_OUT LOW TO DIGITAL_OUT 0x2
_RDY

RQ IO MAP ALL
IO MAP TEST_IN_1 DISCRETE_IN LOW DIGITAL_IN 0x00000002
IO MAP dio_in NUMERIC_IN HI DIGITAL_IN 0x203fffff
IO MAP dio_out NUMERIC_OUT HI DIGITAL_OUT 0x000fffff
IO MAP hw_emer_stop EMER_STOP LOW DIGITAL_IN 0x20000000
_RDY

SET IO STATE TEST_OUT_1 ACTIVE
_RDY
```

Copyright © 2023, Brooks Automation, Inc.

ECHO - Example of RETRACT_SEN output signal

```
MAP TEST_OUT RETRACT_SEN LOW TO DIGITAL_OUT 0x01
_RDY

SET STN 99 ARM A OPTION RETRACT_SEN TEST_OUT
_RDY

ECHO - Verify that the option is configured on the station

RQ STN 99 OPTION RETRACT_SEN
STN 0099 TEST_OUT
_RDY

ECHO - Put the arm in a known position

HOME ALL
_RDY

ECHO - Ask the robot if the arm is retracted

RQ IO STATE TEST_OUT
IO STATE TEST_OUT RETRACTED
_RDY
```


Copyright © 2023, Brooks Automation, Inc.

Appendix C. Dual Independent Robot

All functionality is the same except for the following conditions:

- Before Home
  - AT_STN is 0
  - STN_BITS are (0,0,...,0)
  - NOT_MOVING is 0
  - ARM_IN_USE is 1
  - R BITS are 01
  - Z BITS are 01
- After Home
  - AT_STN is 0
  - STN_BITS are (0,0,...,0)
  - NOT_MOVING is 0
  - ARM_IN_USE is 1
  - R BITS are 01
  - Z BITS are 01
- Single Axis Extend (1 arm) Only 1 arm extended sets extended bit
  - AT_STN is 0
  - STN_BITS are (0,0,...,0)
  - NOT_MOVING is 0
  - ARM_IN_USE is 0 or 1
  - R BITS are 10
  - Z BITS are 01
- Single Axis Retract (1 arm) Both arms must be retracted ti set retracted bit
  - AT_STN is 0
  - STN_BITS are (0,0,...,0)
  - NOT_MOVING is 0
  - ARM_IN_USE is 0 or 1
  - R BITS are 10
  - Z BITS are 01
- Both arms must be set at Station for At Station bit to set
- Command received to move ARM A to a station
  - AT_STN is 0
  - STN_BITS are (0,0,...,0)
  - NOT_MOVING is 0
  - ARM_IN_USE is 0
- Command received to move ARM B to a station
  - AT_STN is 0
  - STN_BITS are (0,0,...,0)

Copyright © 2023, Brooks Automation, Inc.

    - NOT_MOVING is 0
    - ARM_IN_USE is 1
- ARM B arrives at its station first
    - AT_STN is 0 (ARM A is still moving)
    - STN_BITS continues to be based on the station # for Arm B
    - NOT_MOVING is 0
    - ARM_IN_USE is 1
- ARM A arrives at its station last
    - AT_STN is 1
    - STN_BITS continues to be based on the station # for Arm B
    - NOT_MOVING is 1
    - ARM_IN_USE is 1

Copyright © 2023, Brooks Automation, Inc.

Appendix D. Error Codes

Table 18-1: LEAP Error Codes

Error #	Description
1	Internal Tcl error
210	Robot not at a station.
220	Robot not at retract position.
221	Invalid Arm.
233	Extend IO Interlock not enabled.
255	Global STN_OFFSET not enabled.
256	Over STN_OFFSET MAX_R limit.
257	Over STN_OFFSET MAX_T limit.
301	Duplicate command argument.
305	Unknown command or argument.
306	Command argument is out of range or invalid.
309	Command not supported due to configuration.
390	Command checksum is invalid.
392	Function not enabled.
402	Bad slot number.
405	Bad Station Lower Value.
406	Bad Station Pitch Value.
407	Bad Station Theta-T Value.
408	Bad Station Radial-R Value.
409	Bad Station Vertical-Z Value.
416	Station not initialized.
417	Robot offset too big.

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
419	Station VIA not cleared.
420	Invalid Station parameter.
422	Via Point Required, but not set.
423	Via Point Set, but not CURVE_MOTION Option.
426	Bad Station Via R Value.
427	Bad Station Via T Value.
451	Over Max Deviation Limit.
454	Bad Z_WAF: value is greater than the Lower value.
508	wafer sensor not found.
509	No Z robot.
510	No W robot.
511	No R robot.
512	No articulated wrist.
513	No A axis.
514	Not 5-axis robot.
515	No T3 axis.
516	No T robot.
601	Command Canceled.
603	Command halted.
604	CDM In Control.
611	Warning, CDM has been turned on.
612	Warning, CDM has been turned off.
615	Device Not In Control.
616	Safety Module lost CDM In Control.
658	This command is not allowed at this time.
659	Invalid command identifier.
660	Command identifier already in use.
710	Slot valve closed prior to PICK, PLACE, GOTO, or XFER.
715	Unexpected material.
730	Missing material.
734	No wafer during arm extend.

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
735	Wafer sensor failed.
736	Wafer detected during arm retract.
743	Wafer Sensor not configured.
744	Wafer detected during arm extend.
745	No wafer during arm retract.
750	No stn with wafer sensor found for ARM A.
751	No stn with wafer sensor found for ARM B.
770	Wafer Sensor not found.
771	Invalid number of sensor data.
772	Wafer Broken Error.
773	Wafer Sensor Not Enabled.
774	Bad Wafer Position Data.
775	Invalid sensor index range.
777	AWC Calculation error.
778	Sensor not found.
790	Invalid sensor mode setting.
791	No capture data.
792	Invalid position interpolation.
793	AWC sensing is terminated.
794	AWC timeout on wait for motion.
800	Robot application not found.
802	Robot application not found.
805	Robot illegal configuration.
810	Unable to open master configuration file.
811	Error reading from master configuration file
812	Unable to open object data file.
813	Error reading from object data file.
816	Unable to open current configuration file.
817	Error reading from current configuration file.
818	Error writing current configuration file
820	Unknown target for robot application property.

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
821	Robot application property not found.
825	Robot application group not found.
828	Error validating or applying robot application properties.
830	Robot application not loaded.
856	Robot application bad or unknown.
863	Unable to load robot application due to robot state.
1001	Invalid IO data type.
1002	Invalid IO name.
1003	IO name used.
1004	IO out of memory.
1005	IO name reserved.
1006	IO bad bit size.
1007	IO bad block.
1008	IO bad bit mask.
1009	IO bad IO type.
1010	IO type mismatch.
1011	IO bad channel.
1012	IO bad handle.
1013	IO bad state name.
1017	IO op. state OFFLINE.
1304	Hard tracking error.
1305	Torque limit exceeded.
1316	Home, No Safe Path found
2051	Forwarding to Tcl error
2300	Error OK
2301	A checksum error was detected
2302	An attempt to write to a wrong address was detected
2303	Underlying system not able to respond in time
2304	An attempt to read from a register that was read-protected
2305	An attempt to write to a register that was write-protected
2306	An error was detected in the format of the message

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
2307	A message was received, where the actual message length did not match the length of the message
2308	A request was received for a transaction, which is not implemented on the connected node
2309	Reserved
2310	Reserved
2311	Reserved
2312	A write to an outputs register fails because outputs are disabled on the connected node
2313	Reserved
2314	Reserved
2315	Reserved
2316	Received an input line longer longer than max allowed
2317	Writing to a serial device failed
2318	A message with an ivalid COP was received
2319	An empty message (or just extra termination) was received
2400	Take control requests not supported for this interface
2401	Take control request from an unknown interface
2402	Timeout when enabling motor power
2403	Timeout when releasing control
2404	Timeout when taking control
2405	Timeout when transitioning to new system mode
2406	Inconsistent safety inputs: TEACH, but NOT MANUAL
2407	Inconsistent safety inputs: TEACH, but NOT PENDANT
2408	Persistent communication errors with IO board
2409	No safety input handler is subscribed
2700	LED color value is out of range
2701	Unknown LED blink type
4001	Drive serial number required to create birth certificate.
4002	Manufacturing robot application required to create birth certificate.
4003	Birth certificate is already created.
4016	Unable to store birth certificate.
4110	Bad input value.
4111	Invalid Event ID.

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
6301	License violation - Version not supported.
6302	License violation - Application not supported.
6303	License violation - Current User Level not supported.
6304	License violation - Product feature not supported.
6305	License violation - Service Key is Required.
6306	License violation - Service Key is Required B.
6400	License Service Error.
6401	License Service - File Not Found.
6402	License Service - File Format Not Valid.
6403	License Service - Storage Tamper Error.
6404	License Service - Wrong GUID/HostID Error.
6405	License Service - Invalid License Parameter.
8101	Toggle bit not changed
8102	SDO timeout
8103	Command specifier unknown
8104	Out of memory
8105	Unsupported Access
8106	Write only entry
8107	Read only entry
8108	Object not existing
8109	Object can not be mapped to PDO
8110	Mapped Object exceeds PDO
8111	Parameter is incompatible
8112	Device incompatibility
8113	Hardware error
8114	Parameter length error
8115	Parameter is too long
8116	Parameter is too short
8117	Subindex (Entry) not exists
8118	Value exceeds
8119	Value is too great

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
8120	Value is too small
8121	Unequal Module Id list
8122	Value is less than minimum value
8123	General error
8124	Data can not be read or written
8125	Data can not be accessed because of local control
8126	Data can not be read or written in the current state
8127	Object is not in the object dictionary
8128	Entry can not be written because Subindex0 is not 0
8129	The object cannot be accessed via complete access
8130	Requested PDI register range is too large
8131	CANOpen Object does not exist
8132	CANOpen Object sub-index does not exist
8133	CANOpen Object data too large
8134	CANOpen Object description is missing
8135	CANOpen Object description is too big
8136	Sync Manager index is out of range (0-3)
8137	Unknown module type
8138	IO Block aready mapped. Remove map first.
8139	IO Blocks aready exist
8140	IO Blocks aready removed
8141	Not in operational (OP) state
8142	EtherCAT DIO is simulated
8143	PDI is not in control of MII bus
8144	MII bus is busy
8145	Phy reg. read error
8146	Phy reg. write error
8147	Mapping of FPGA memory failed
8148	Getting QNX IO privileges failed
8149	Wrong PHY configuration data
10000	Default debug message from the MC.

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
10001	Sync error, motor moving or encoder noisy.
10002	MC board memory allocation error.
10003	MC board unexpected event error.
10004	MC board, bad command state.
10005	MC board sync error, can't move motor.
10006	MC encoder VABS adjusted (small).
10007	MC board position settle error.
10008	MC unable to hold position.
10009	MC hard tracking error.
10010	MC soft tracking error.
10011	Motor is already moving.
10012	Motor is not configured.
10013	Motor is not referenced.
10014	Motor is already referencing.
10015	Motor is currently moving.
10016	Unable to calculate trajectory.
10017	Illegal number of polls calculated.
10018	Unable to calculate absolute position.
10019	Encoder off by many sectors.
10020	Encoder failed multiple times.
10021	Board power failure (blown fuse?).
10022	Axis overtravel limit reached.
10023	Arm actual location impossible, check sync zero.
10024	MC watchdog timed out.
10025	Defective R_MT type wafer sensor.
10026	Arm load not what expected.
10027	Specific to VT6 robot issue.
10028	Obstruction encountered for axis.
10029	Emergency stop circuit is active.
10030	Excessive current detected.
10031	Home sensor position moved.

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
10032	Map data invalid.
10033	Map command received while mapping.
10034	Enoder min/max value out of range.
10035	Bad Sync Phase Offset Value.
10036	robot links are not yet defined.
10037	MC comm port trans/recv overflowed.
10038	MC comm port trans/recv is empty.
10039	MC comm port is not open.
10040	MC comm port received a bad message.
10041	MC comm port message checksum failed.
10042	MC comm port initialization failed.
10043	MC comm port FIFO overrun.
10044	MC comm port framing error.
10045	MC comm port is not responding.
10046	MC bad hall sensor data value.
10047	MC code date/time stamp mismatch.
10048	MC stored code checksum incorrect.
10049	MC code size too large to store.
10050	I2C bus is busy.
10051	No ack on I2C send data.
10052	No ack on I2C send address.
10053	I2C bus error.
10054	DPRAM bad MC head/tail pointer.
10055	DPRAM bad SUP head/tail pointer.
10056	DPRAM bad SUP message size.
10057	Home sensor position moved too far.
10058	Set loop mode failed.
10059	Motor step move timed out.
10060	Reference failed.
10061	position impossible to reach.
10062	MC cannot send ready message.

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
10063	MC cannot send GO message.
10064	MC cannot send remote trajectory.
10065	MC did not receive remote trajectory.
10066	Bad sine/cosine values for encoder.
10067	Bad remote SYNC State.
10068	Illegal argument.
10069	Coordinate not supported.
10070	Servo disabled.
10071	Halt in progress.
10072	Move buffer is full.
10073	Singularity.
10074	Servo failed.
10075	Brake not engaged.
10076	Servo enabled.
10077	No end-effector B.
10078	Axis does not exist.
10079	Capture invalid time stamp.
10080	Wait move done time out
10081	Invalid move (blend option or singularity)
10082	Unable to clear SC axis errors
10083	Unable to clear SC mimo errors
10084	Op mode mismatch
10085	SC not ready to servo
10086	SC servo time out
10087	Blend option, cascaded wait next not allowed
10088	Center crossing
11001	Exceed velocity error limit.
11002	Commanded motor current rise too fast.
11003	Commanded motor current saturated.
11004	HW motor over current detected.
11005	HW brake failure detected.

Copyright © 2023, Brooks Automation, Inc.

Error #	Description
11006	Illegal operation mode.
11007	PWM motor power disabled.
11008	Motor torque limit reached.
11009	Encoder CRC error.
11010	Motor phase current imbalance.
11011	State estimator Error too high.
11012	Servo background task cleared.
11013	Unbalanced motor phase current.
11014	HW motor bus over current detected.
11015	One or more axis stopped while MIMO is active.
11016	Motor temperature too high.
11017	Motor bus voltage too high.
11018	Motor bus voltage too low.
11019	Encoder internal warning flag set.
11020	Encoder internal error flag set.
11021	Un-handled exception detected in Servo Controller.
11022	Robot dynamical model exception.
11023	Motion trajectory exception.
11024	Coordinate update exception from Motion Controller.
11025	Incorrect spin mode configuration.
16000	Resume Failed, Motion Not Paused.
16001	Pause Failed, No Active Command.
16002	Command Not Allowed While Paused
16003	Invalid Event
24042	ATEC feature not enabled.
24043	ATEC not calibrated.
24044	ATEC calculation failed.

Copyright © 2023, Brooks Automation, Inc.