



IFA-600-AUCME3

Optical Device Alignment/Bonding System

Programming Manual Revision 1.0

[FMA-OMF6-MA2609A4-Rev10]

Sep 21, 2026

FIBERPRO, Inc.
TEL +82-42-360-0030
FAX +82-42-360-0040
<http://www.fiberpro.com>
E-mail: sales@fiberpro.com

Revision History

Sep. 21, 2026: Rev. 1.0

- Official release

Table of Contents

Revision History.....	2
1. Control Commands.....	8
1.1. Common parameters.....	9
1.1.1. Name of the stages and directions.....	9
1.1.2. Name of cameras.....	11
1.1.3. Names of powermeter/currentmeter.....	11
1.2. Movement commands and position commands.....	12
1.2.1. ABSMOVE.....	12
1.2.2. RELMOVE.....	12
1.2.3. STARTPOSITION.....	12
1.2.4. RELOAD.....	13
1.2.5. INPUT_ALIGN_POS.....	13
1.2.6. OUTPUT_ALIGN_POS.....	13
1.2.7. HOMING.....	13
1.2.8. MEMORIZE_POSITION.....	14
1.2.9. MOVE_TO_POSITION.....	14
1.2.10. GOABSMULTIMODE.....	15
1.2.11. RETURNABSMULTIMODE.....	15
1.2.12. SWITCH_CAMERAPOS.....	16
1.2.13. ZBACK.....	16
1.3. Alignment commands.....	17
1.3.1. ZAPPROACH.....	17
1.3.2. ROUGH_APPROACH.....	17
1.3.3. APPROACH.....	18
1.3.4. CUSTOM_APPROACH.....	19
1.3.5. ANGLESEARCH.....	19
1.3.6. CUSTOM_ANGLESEARCH.....	20
1.3.7. XYSEARCH.....	21
1.3.8. XYBLINDSEARCH.....	22
1.3.9. STEPSEARCH.....	24
1.3.10. STEPSEARCH_DAQ.....	25
1.3.11. GROUP_STEPSEARCH.....	25
1.3.12. ZOPTIMIZE.....	26
1.3.13. CUSTOM_OPTIMIZE.....	27
1.3.14. PEAK_SEARCH.....	28
1.3.15. XYBALANCESEARCH.....	29
1.3.16. CHANNEL_OPTIMIZE.....	30
1.3.17. CUSTOM_CHANNEL_OPTIMIZE.....	31
1.3.18. ROLLSEARCH.....	32
1.3.19. CUSTOM_ROLLSEARCH.....	33
1.3.20. U_ROLLSEARCH.....	34
1.3.21. SIDECHECK.....	35
1.4. Message commands.....	36
1.4.1. MSG.....	36
1.4.2. SPLASH_MSG.....	36
1.4.3. RUNMSG / POPUP_RUN_BTN.....	37
1.4.4. TMSG.....	37
1.4.5. POWMONMSG.....	38
1.4.6. SET_VARIABLE_MSG.....	38
1.4.7. QMSG.....	39
1.4.8. UPDATE_VARIABLE_UI.....	39
1.4.9. CHECK_POWER.....	41
1.4.10. XYSCAN.....	42

1.4.11. TWOAXIS_SCAN.....	43
1.4.12. 3DSCAN.....	43
1.4.13. EPOXYMSG.....	44
1.4.14. UVMSG.....	44
1.4.15. TEMPMONMSG.....	45
1.4.16. CONDITIONAL_WAIT_TEMP.....	45
1.5. Save commands.....	47
1.5.1. SAVE_DATA.....	47
1.5.2. MAKE_CUSTOM_REPORT.....	47
1.6. Lamp commands.....	48
1.6.1. SET_LAMP.....	48
1.7. UV commands.....	48
1.7.1. UV_CURING.....	48
1.7.2. UVSOURCE.....	49
1.7.3. SETCURINGTIME.....	49
1.7.4. SET_UV_INTENSITY.....	49
1.8. Dispenser command.....	49
1.8.1. EPOXY_APPLY.....	49
1.8.2. EPOXY.....	50
1.8.3. EPOXY_DISPENSING.....	50
1.8.4. EPOXY_DISPENSING_LINE.....	50
1.8.5. SYRINGESHIFT.....	51
1.9. TEC commands.....	51
1.9.1. TEC_TEMP.....	51
1.9.2. TEC_OUTPUT.....	51
1.9.3. TEC_THERMISTOR_CONST.....	51
1.9.4. TEC_PID_CONST.....	51
1.9.5. TEC_CURRENT_LIMIT.....	52
1.9.6. TEC_AUTOTUNE (Only for Arroyo 53xx series).....	52
1.10. Variable manager.....	52
1.10.1. CREATE_VARIABLE.....	52
1.10.2. UPDATE_VARIABLE.....	53
1.10.3. DELETE_VARIABLE.....	53
1.10.4. CALC_VARIABLE.....	53
1.10.5. LINEAR_VARIABLE.....	54
1.11. Miscellaneous commands.....	54
1.11.1. LEFTFIX.....	54
1.11.2. RIGHTFIX.....	54
1.11.3. SET_VACUUM.....	54
1.11.4. REARCAMERA.....	55
1.11.5. SET_SOLENOID.....	55
1.11.6. EXECMACRO.....	55
1.11.7. LOAD_MACRO.....	55
1.11.8. PAUSE.....	55
1.11.9. STEP.....	56
1.11.10. SET_COORDINATE_MODE.....	56
1.11.11. SET_GUI_LOCK.....	56
1.11.12. CHECK_PROCESSINFO.....	57
1.11.13. CLEAR_PROCESSINFO.....	57
1.11.14. MDU_CURRENT.....	57
1.11.15. SET_OPTICALSOURCE.....	58
1.11.16. CTS_ZEROING.....	58
1.11.17. CTS_UNIT.....	58
1.11.18. SELECT_SPEEDTABLE.....	59
1.11.19. MONITOR_POWER.....	59
1.11.20. GET_GUIDE_POS.....	59
1.11.21. TERMINAL_SHOW.....	59

1.11.22. SET_LOOPCNT.....	60
1.11.23. WAIT_FOR_STOP.....	60
1.11.24. SEQUENCE_STOP.....	60
1.11.25. UPDATE_POSITION.....	60
1.11.26. SAVE_POSITION.....	60
1.11.27. IF_CONDITION / ELSE_CONDITION / ELSEIF_CONDITION / END_CONDITION.....	61
1.11.28. CHECK_CONDITION.....	62
1.11.29. MDU_CONTROL.....	64
1.11.30. SET_PM_WAVE.....	64
1.11.31. SET_PM_RANGE_UPPER.....	64
1.11.32. GET_PM_RANGE_UPPER.....	65
1.11.33. SET_PM_AUTO_RANGE.....	65
1.11.34. PM_COMM.....	66
1.11.35. SET_PM_AOUT.....	66
1.11.36. SET_PM_GAIN.....	67
1.11.37. SET_FAN.....	67
1.11.38. HANDLE_COLLISION.....	67
1.11.39. STAGE_CONTROL_MODE.....	67
1.11.40. SELECT_TAB.....	67
1.11.41. SET_SW_LIMIT.....	68
1.11.42. USB_COMM.....	68
1.11.43. GPIB_COMM.....	69
1.11.44. CREATE_RANDOM.....	69
1.11.45. SAVE_LAYOUT.....	69
1.11.46. LOAD_LAYOUT.....	70
1.12. System test commands.....	70
1.12.1. TEST_XYSEARCH.....	70
1.12.2. TEST_APPROACH.....	71
1.12.3. TEST_CUSTOM_APPROACH.....	72
1.12.4. TEST_ANGLESEARCH.....	73
1.12.5. TEST_CUSTOM_ANGLESEARCH.....	74
1.12.6. TEST_ROLLSEARCH.....	75
1.12.7. TEST_CUSTOM_ROLLSEARCH.....	76
1.12.8. TEST_U_ROLLSEARCH.....	77
1.12.9. SEARCHZEXTREMA.....	78
1.12.10. TEST_ZSCAN.....	78
1.13. Image processing commands.....	79
1.13.1. V_SET_COORDINATE.....	80
1.13.2. SHOW_ROIBOX_LIST.....	82
1.13.3. V_DRAWCROSSLINE.....	82
1.13.4. V_ADDCROSSLINE.....	83
1.13.5. V_IMAGEDISPLAYMODE.....	83
1.13.6. V_SYNC_CAMERAPOS.....	83
1.13.7. V_DECODE_BARCODE.....	84
1.13.8. SELECT_CAMERA.....	84
1.13.9. V_MOVE_TO_POINT.....	85
1.13.10. V_MOVE_TO_PATTERN.....	85
1.13.11. V_CALIB_IMAGEPIXELSIZE.....	86
1.13.12. V_ACCUMULATE_IMAGE.....	87
1.13.13. V_ACQUIRE_TEMPLATEIMAGE.....	87
1.13.14. V_SHOW_FRAMERATE.....	87
1.13.15. SET_CAMERAOPTION.....	88
1.13.16. V_CALIB_INSTERROR.....	88
1.13.17. V_SET_LINE_PARALLEL.....	88
1.13.18. V_SET_PARALLEL.....	90
1.13.19. CALIB_STAGE_TILT_THY.....	92
1.13.20. V_AUTOFOCUS.....	94

1.14. Remote control commands.....	95
1.14.1. GET_MASTER_PERMISSION.....	95
1.14.2. HANDOVER_MASTER_PERMISSION.....	95
1.14.3. CHECK_LASTERROR.....	96
1.14.4. CHECK_SEQSTATUS.....	96
1.14.5. CHECK_MSGSTATUS.....	96
1.14.6. SEQUENCE_CONTROL RUN / RESUME / STOP.....	96
1.14.7. SEQUENCE_LOAD.....	96
1.14.8. ISBUSY?.....	97
1.14.9. CONTROL_MSGBOX.....	97
1.14.10. GET_VARIABLE.....	97
1.14.11. CHECK_STATUSMASK.....	97
1.15. Return codes for remote control commands.....	98
2. Using Control Software.....	99
2.1. Using software.....	99
2.1.1. Loading a sequence.....	99
2.1.2. Run/Stop/Resume of a sequence.....	99
2.1.3. Using command execution window.....	99
2.2. Preparing image processing.....	99
2.2.1. Acquiring template images.....	99
2.2.2. Camera pixel size calibration.....	100
2.2.3. Mapping camera image position to real position.....	101
2.3. Example: Sequence for alignment/bonding.....	102
2.3.1. Example sequence.....	102
3. Remote Control of IFA-600-AUCME3.....	105
3.1. Introduction.....	105
3.2. Example code.....	106
4. Variable Manager.....	107
4.1. List of system variables.....	107
5. Appendix.....	109
5.1. Files used by software.....	109
5.1.1. Configuration file for the process information window.....	109
5.1.2. Sequence file.....	111
(a) LOOP OPERATION.....	113
(b) INCLUDE.....	114
5.1.3. Port setting “S.P”, “M.P”.....	116
5.1.4. Calibration data file.....	116
5.1.5. Stage information file.....	117
5.1.6. Macro file.....	118
5.1.7. Macro button.....	119
5.1.8. Jog controller settings.....	119
5.1.9. GUI Lock.....	121
5.1.10. Danger zone.....	122
5.2. FIBERPRO Limited Warranty.....	124

List of Figures

Fig. 1.1: Directions of the stages.....	9
Fig. 1.2: Names of stages in the left/right alignment stage assemblies.....	9
Fig. 1.3: Names of stages in the center stage assembly/tool positioner.....	10
Fig. 1.4: Flow chart of CHECK_CONDITION command.....	63
Fig. 1.5: Types of CHECK_CONDITION messages (Condition=FALSE).....	64
Fig. 1.6 The result of “TEST_XYSEARCH” command.....	75
Fig. 1.7 The result of “TEST_APPROACH” command.....	76

Fig. 1.8 The result of “TEST_CUSTOM_APPROACH” command.....	77
Fig. 1.9 The result of “TEST_ANGLESEARCH” command.....	78
Fig. 1.10 The result of “TEST_CUSTOM_ANGLESEARCH” command.....	79
Fig. 1.11: The result of “TEST_ROLLSEARCH” command.....	80
Fig. 1.12: The result of “TEST_CUSTOM_ROLLSEARCH” command.....	81
Fig. 1.13: The result of “TEST_U_ROLLSEARCH” command.....	82
Fig. 1.14: Configuration file for image processing.....	83
Fig. 2.1: Main window.....	104
Fig. 2.2: Acquiring a template image.....	105
Fig. 2.3: Obtaining scale factor of camera image.....	106
Fig. 2.4: Mapping a position on a camera image to a physical position.....	106
Fig. 3.1: Remote operation of IFA-600-AUCME3 software.....	110
Fig. 3.2: Example code for the remote control.....	111
Fig. 5.1: File: ProcessInfo.ini.....	115
Fig. 5.2: Sequence file example.....	117
Fig. 5.3: Expanding-folding command list.....	118
Fig. 5.4: Block diagram of the loop operation.....	119
Fig. 5.5: Display the number of repetition.....	120
Fig. 5.6: The format of "INCLUDE" section.....	120
Fig. 5.7: Sequence file.....	120
Fig. 5.8: Displayed message on the "Sequence Processing" window.....	121
Fig. 5.9: Structure of the sequence with the "INCLUDE" function.....	121
Fig. 5.10: Deviceinfo.ini.....	122
Fig. 5.11: Measurement_Calibration.ini.....	122
Fig. 5.12: Stage information file.....	123
Fig. 5.13: seqmacro.ini.....	124
Fig. 5.14: Macro buttons.....	125
Fig. 5.15: File: ViewInfo.ini.....	125
Fig. 5.16: Jog controller setting example.....	126
Fig. 5.17: GUI Lock setting.....	127
Fig. 5.18: GUI Lock item #1.....	127
Fig. 5.19: GUI Lock item #2.....	128
Fig. 5.20: Danger zone.....	128
Fig. 5.21: Setting “Viewinfo.ini” for the danger zone.....	129

1. Control Commands

- All commands and parameters are case insensitive except user messages and file names.
- The separator between a command and parameter(s) is one or more SPACE(" ") characters.
- The separator in parameters is a COMMA(",") character.
- All the commands return values containing the result of the execution. The return values are also displayed on the process monitoring window. The return values are shown in the table below.

Table 1.1: Return values of commands

TYPE	Return value	Description
RET_NOBUSY	2	The system is ready to receive a command. Return code only for "ISBUSY?" command
RET_BUSY	1	The system is executing a command, and not ready to take another command. Return code only for "ISBUSY?" command
RET_SUCCEEDED	0	A command is completed with success.
RET_FAILED	-1	A command is completed with failure.
RET_STOP	-100	A command is completed by user interrupt. Return code obtained by "CHECK_LASTERROR" command after the last command has been stopped by "SEQUENCE_CONTROL_STOP" command.
RET_EIO	-99	I/O error in communication with peripherals
RET_EIO_READ	-98	I/O read error in communication with peripherals
RET_EIO_WRITE	-97	I/O write error in communication with peripherals
RET_EINVALARG	-96	Invalid argument
RET_EFILE	-95	File handling error
RET_UNKNOWNCMD	-94	Unknown command
RET_VAR_ALREADY_EXISTED	-92	Failed to create a variable since a variable of the same name is existing.
RET_VAR_NOT_EXISTED	-91	The variable does not exist.
RET_VAR_NOT_PERMITTED	-90	No permission to handle the variable
RET_VAR_ISINTERNAL	-89	Tried to change a system variable.
RET_VAR_ISSTRING	-88	Failed to perform a numeric operation.
RET_VAR_INVALID_DELIMITER	-87	Invalid delimiter
RET_HOMING_REQUIRED	-56	The system needs to do the homing process.
RET_NO_PATTERN	-55	Failed to find the specified pattern from the image.

1.1. Common parameters

1.1.1. Name of the stages and directions

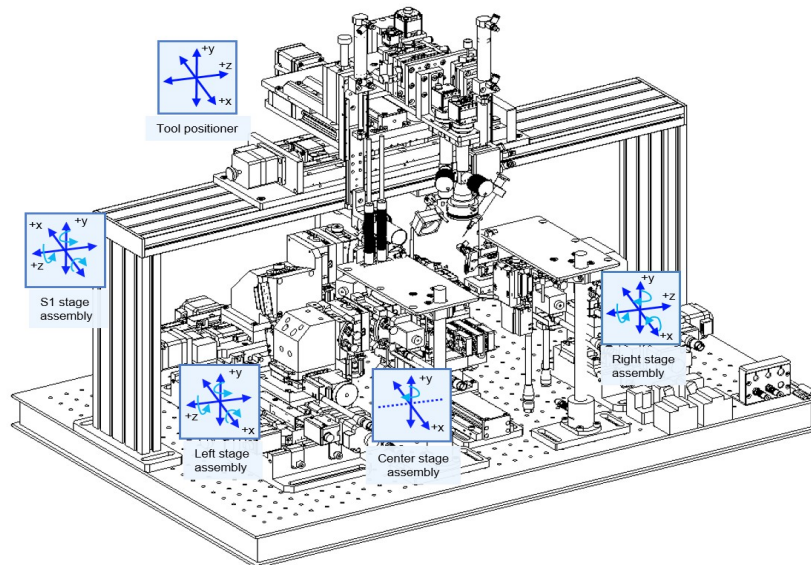


Fig. 1.1: Directions of the stages

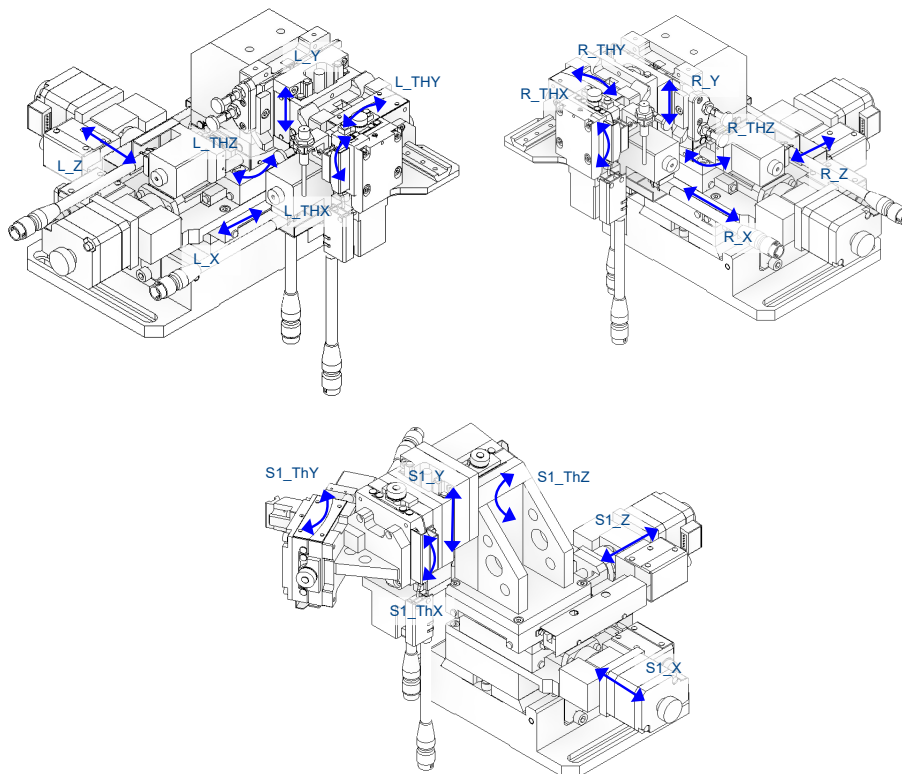


Fig. 1.2: Names of stages in the left/right alignment stage assemblies

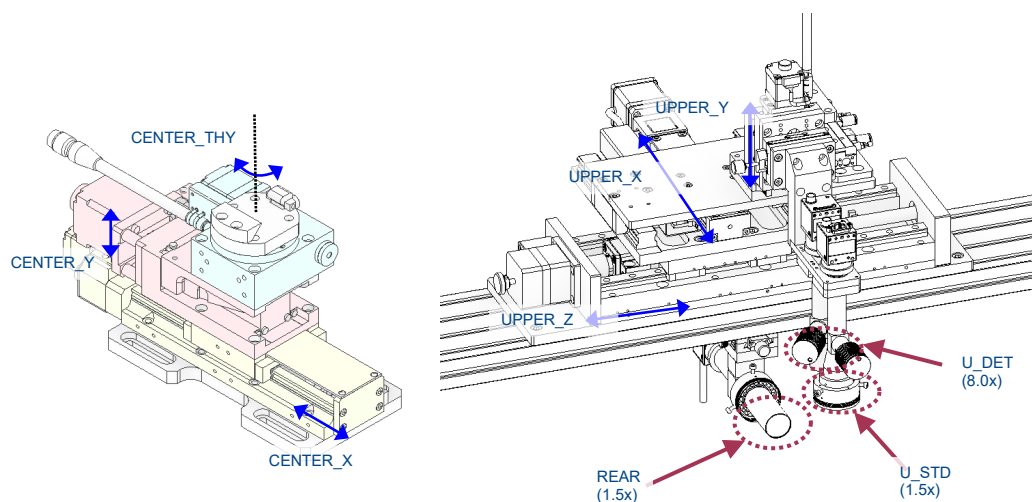


Fig. 1.3: Names of stages in the center stage assembly/tool positioner

AXIS	POSITION (UNIT)
L X	μm
L Y	μm
L Z	μm
L THX	Degree
L THY	Degree
L THZ	Degree
S1 X	μm
S1 Y	μm
S1 Z	μm
S1 THX	Degree
S1 THY	Degree
S1 THZ	Degree
R X	μm
R Y	μm
R Z	μm
R THX	Degree
R THY	Degree
R THZ	Degree
UPPER X	μm
UPPER Y	μm
UPPER Z	μm
CENTER X	μm
CENTER Y	μm
CENTER THY	Degree

Table 1.2: Name of stages

Ass'y name	Description(s)
LEFT	LEFT alignment stage assembly
S1	S1 alignment stage assembly
RIGHT	RIGHT alignment stage assembly
UPPER	Tool positioner
CENTER	CENTER stage assembly

Table 1.3: Name of stage ass'y

1.1.2. Name of cameras

Name of camera	Description(s)
U STD	Upper camera 1 (1.5 x)
U DET	Upper camera 2 (8.0 x)
REAR	Rear camera (1.5 x)

Table 1.4: Name of cameras

1.1.3. Names of powermeter/currentmeter

Name of channels	Description(s)
PMx	Powermeter(x=1...4)
CMx	Currentmeter(y=1...4)

Table 1.5: Names of powermeter/currentmeter channels

1.2. Movement commands and position commands

1.2.1. ABSMOVE

ABSMOVE	
Description :	It moves the specified stage to the specified position. ABSMOVE is an abbreviation of ABSolute MOVEMENT.
Syntax :	ABSMOVE [AXIS],[POSITION]
Parameter	[AXIS]: Axis name (See Name of stages) [POSITION]: Absolute position
Optional Parameter(s)	[BC]: Backlash compensation [SP]: Speed (1~5) [NWS]: With this option, it orders movement to the motorized stages and terminates the process immediately without waiting for motor stop signal. [TWISTBACK]: Only applicable to left/right z stage. It rotates ThX stage by 0.5 degree in negative direction before retreating the z stage, and restore the ThX stage position after the retreat.
Response :	See Table 1.1: Return values of commands.
Example :	ABSMOVE L_X,400 → It moves the X axis of the left stage ass'y moves to the coordinate of "400 (μm)".

1.2.2. RELMOVE

RELMOVE	
Description :	It moves the specified stage by a certain distance. RELMOVE is an abbreviation of RELative MOVEMENT.
Syntax :	RELMOVE [AXIS],[DISTANCE]
Parameter :	[AXIS]: Axis name (See Name of stages) [DISTANCE]: Relative distance
Optional Parameter(s)	[BC]: Backlash compensation [SP]: Speed (1~5) [NWS]: With this option, it orders movement to the motorized stages and terminates the process immediately without waiting for motor stop signal. [TWISTBACK]: Only applicable to left/right z stage. It rotates ThX stage by 0.5 degree in negative direction before retreating the z stage, and restore the ThX stage position after the retreat.
Response :	See Table 1.1: Return values of commands.
Example :	RELMOVE L_Z 500, BC, SP=3 → It moves L_Z stage by 500 um with stage speed "3".

1.2.3. STARTPOSITION

STARTPOSITION	
Description :	It moves the specified stage(s) to the starting positions, which are defined the sequence loaded for the operation. In a sequence file, there are two sets of positions, and those sets need to be defined when the sequence file is created. The stages to be controlled can be selected with parameters.
Syntax :	STARTPOSITION [AXIS = L_X, L_Y...]
Parameter(s) :	[AXIS]: Axis name (See Name of stages)
Response :	See Table 1.1: Return values of commands.
Example :	STARTPOSITION L_X, L_Y, L_Z STARTPOSITION L_X, L_Y, L_Z, L_THX, L_THY, L_THZ, R_X, R_Y, R_Z, R_THX, R_THY, R_THZ

1.2.4. RELOAD

RELOAD	
Description :	This command moves the stages to a predefined position(Reload Position) in the following orders: 1. L_Y, R_Y and S1_Y stages move upward by 5,000 μm. 2. L_Z, R_Z and S1_Z stages move to "Reload position". 3. L_Y, R_Y and S1_Y stages move down by 5,000 μm. 4. L_X, R_X, S1_X and CENTER_X stages move to "Reload position". The reload position can be set from the position window, and the position information of which is saved in the file "IFA-600-AUCME3\System\Mc\mc_stageinfo_table.ini" (See details in 5.1.5)
Syntax :	RELOAD
Parameter(s) :	NONE
Response :	See Table 1.1: Return values of commands.
Example :	RELOAD

1.2.5. INPUT_ALIGN_POS

INPUT_ALIGN_POS	
Description :	It moves tool positioner, left stages to the predefined position for monitoring /vision processes. The position is defined in the current sequence file, and editable from the "Position windows".
Syntax :	INPUT_ALIGN_POS
Parameter(s) :	NONE
Optional Parameter(s) :	[SP]: Speed (1~5)
Response :	See Table 1.1: Return values of commands.
Example :	INPUT_ALIGN_POS

1.2.6. OUTPUT_ALIGN_POS

OUTPUT_ALIGN_POS	
Description :	It moves tool positioner and right stages to the predefined position for monitoring/vision processes. The position is defined in the current sequence file, and editable from the "Position windows".
Syntax :	OUTPUT_ALIGN_POS
Parameter(s) :	NONE
Optional Parameter(s) :	[SP]: Speed (1~5)
Response :	See Table 1.1: Return values of commands.
Example :	OUTPUT_ALIGN_POS

1.2.7. HOMING

HOMING	
Description :	This command starts homing process for all the stages, and moves the stages to the predefined positions(Home positions) after the homing process. The home positions is defined in the file "mc_stageinfo_table.ini" (See details in 5.1.5). The command moves the stages following orders. 1. L_Z, R_Z and S1_Z stages move backward by 60,000 μm. 2. L_X, UPPER_X, UPPER_Y and UPPER_Z stages move by the origin offset defined by configuration file. (See details in 5.1.5)

	3. S1_X stages move backward by 60,000 μm. 4. The software executes the homing process for UPPER_X, UPPER_Y and UPPER_Z stages. 5. UPPER_X, UPPER_Y and UPPER_Z stages move to the “Home position”. 6. L_Y, R_Y and S1_Y stages move upward by 30,000 μm. 7. CENTER_Y stage moves down by 5,000 μm. 8. CENTER_THY stage moves by origin offset. 9. The software executes homing process for CENTER_THY stage. 10. The stages except for the L_X, CENTER_THY, UPPER_X, UPPER_Y and UPPER_Z move by origin offset. 11. The software executes the homing process except for UPPER_X, UPPER_Y, UPPER_Z and CENTER_THY stages. 12. The stages except for the UPPER_X, UPPER_Y and UPPER_Z move to the “Home position”.
Syntax :	HOMING
Parameter(s) :	NONE
Response :	See Table 1.1: Return values of commands.
Example :	HOMING

1.2.8. MEMORIZE_POSITION

MEMORIZE_POSITION	
Description :	This command saves the positions of the specified stages temporarily with a user-defined name. The lifetime of this position information ends when the software terminates.
Syntax :	MEMORIZE_POSITION [NAME],[AXIS = R_X, R_Y...]
Parameter(s) :	[NAME]: A user defined name related with the positions to be saved temporarily [AXIS]: The names of axes whose position are to be saved (See Name of stages)
Response :	See Table 1.1: Return values of commands.
Example :	MEMORIZE_POSITION RIGHTPOSITION,R_X,R_Y,R_THY

1.2.9. MOVE_TO_POSITION

MOVE_TO_POSITION	
Description :	This command moves the specified stages to the position predefined by MEMORIZE_POSITION command. With optional offset values, it moves the stages to the position with the offset distance relative to the predefined position.
Syntax :	MOVE_TO_POSITION [NAME]
Parameter(s) :	[NAME]: • A name of A name of the position set defined by “MEMORIZE_POSITION” command • A name of predefined positions that can be set from the “Position window” * There are predefined positions that can be set from the “Position window”. With the names of the predefined positions, the user can move the corresponding stages to the positions. The names of the predefined positions are as following <NAME>: <Description> “VIEW_IN”: The position of the tool positioner to monitor the input of chip. “VIEW_OUT”: The position of the tool positioner to monitor the output of chip. “VIEW_DUT”: The position of the tool positioner to monitor the center of chip. “POLARIZER_POSITION_01”: The position of the right stage assembly. “POLARIZER_POSITION_02”: The position of the right stage assembly. “UV_CURING_IN”: The position of the tool positioner for the UV curing of the input of

	chip. “UV_CURING_OUT”: The position of the tool positioner for the UV curing of the output of chip. “EPOXY_IN”: The position of the tool positioner for the epoxy dispensing of the input of chip. “EPOXY_OUT”: The position of the tool positioner for the epoxy dispensing of the output of chip. “BARCODE_READING”: The position of the tool positioner for the barcode reading of chip. “ALIGN_CENTER_POS”: The position of the center stage assembly. “TOOL_01”: User-defined positions of the tool positioner. “TOOL_02”: User-defined positions of the tool positioner. “TOOL_03”: User-defined positions of the tool positioner.
Optional Parameter(s) :	[OFFSET_X]: X offset relative to predefined position (Unit: μm) [OFFSET_Y]: Y offset relative to predefined position (Unit: μm) [OFFSET_Z]: Z offset relative to predefined position (Unit: μm) [SP]: Stage speed used in the step move (Range: 1~5)
Response :	See Table 1.1: Return values of commands.
Example :	MOVE_TO_POSITION MyPosition,OFFSET_X=30,OFFSET_Y=10,OFFSET_Z=5 → It moves the specific stages to 30 μm in x direction, 10 μm in y direction, 5 μm in z direction from the registered position, “MyPosition”. MOVE_TO_POSITION VIEW_IN → It moves the tool positioner to the predefined position in the “Position Window” with the name of “VIEW_IN”.

1.2.10. GOABSMULTIMODE

GOABSMULTIMODE	
Description :	This command moves one of the alignment stage assy to the position predefined for the multimode blind search process. The position can be updated by pressing “Update Multi” button in the “Search panel windows”.
Syntax :	GOABSMULTIMODE
Parameter(s) :	NONE
Optional Parameter(s) :	[SP]: Stage speed used in the step move. (Range:1~5)
Response :	See Table 1.1: Return values of commands.
Example :	GOABSMULTIMODE → It moves right assy to the multimode blind search position.

1.2.11. RETURNABSMULTIMODE

RETURNABSMULTIMODE	
Description :	Return one of the stage assy to the position where the stages located right before the GOABSMULTIMODE command was executed.
Syntax :	RETURNABSMULTIMODE
Parameter(s) :	NONE
Optional Parameter(s) :	[SP]: Stage speed used in the step move (Range:1~5)
Response :	See Table 1.1: Return values of commands.
Example :	RETURNABSMULTIMODE → It moves right stage assy. to the original position.

1.2.12. SWITCH_CAMERAPOS

SWITCH_CAMERAPOS	
Description :	This command moves the upper stage and switch the camera view for another camera. The locations displayed in the camera view before and after using this command are the same. This command is useful to monitor a certain location with a difference magnification.
Syntax :	SWITCH_CAMERAPOS [CAM]
Parameter :	[CAM]: Upper camera name (See Name of cameras) or index(1~2)
Response :	See Table 1.1: Return values of commands.
Example :	SWITCH_CAMERAPOS U_DET → The camera monitor window will display the image of the current location taken with the specified camera.

1.2.13. ZBACK

ZBACK	
Description :	This command moves all stage assy. backward in Z direction.
Syntax :	ZBACK [DISTANCE]
Parameter(s) :	[DISTANCE]: distance to move in μm unit
Response :	See Table 1.1: Return values of commands.
Example :	ZBACK 100 → It moves L_Z and R_Z backward by 100 μm .

1.3. Alignment commands

1.3.1. ZAPPROACH

ZAPPROACH	
Description :	This command is to make a FAB contact a chip with fast speed when the distance between the FAB and the chip is relatively large(e.g. >100 μm). It is not for precise gap control. Firstly it pulls back the Z stage by a certain distance(B), then moves the stage(FAB) in the continuous mode toward the chip until the changes the displacement sensor value becomes large than a certain value(Threshold_1). Secondly it pull back the stage, then moves towards the chip step by step with the step of 10 μm until the change in the sensor value is larger than a certain value(Threshold_2). Thirdly it repeats the second process with the step size of 2 μm and the threshold value of "Threshold_3". Finally it pulls back the stages by a certain distance(A) The threshold values are defined in the section "SENSOR" of the file "DeviceInfo.ini".
Syntax :	ZAPPROACH [Stage Ass'y],[Backward after approach A],[Back Distance (B)]
Parameter(s) :	[Stage Ass'y]: Name of stage ass'y [Backward after approach A]: See the description above. (Unit: μm) [Back Distance(B)]: See the description above. (Unit: μm)
Optional Parameter(s) :	[STEPSP]: Stage speed used in the step move (Range:1~5) [CONTSP]: Stage speed used the first contact (Range:1~5) [TWISTBACK]: It rotates ThX stage by 0.5 degree in negative direction before retreating the left/right z stage, and restores the ThX stage position after the retreat. <i>* The units of the distances, and positions are "μm"</i>
Response :	See Table 1.1: Return values of commands.
Example :	ZAPPROACH LEFT,0,200 → It makes a FAB contact a chip with fast speed in the z direction. Firstly it pulls back the z stage by 200 μm, then moves the stage in the continuous mode toward the chip until the changes the displacement sensor value becomes larger than a certain value. Secondly it pulls back the stage, then moves towards the chip step by step with the step of 10 μm until the change in the sensor value is larger than a certain value. Thirdly it repeats the second process with the step size of 2 μm and the threshold value. Finally it pulls back the stages by 0.

1.3.2. ROUGH_APPROACH

ROUGH_APPROACH	
Description :	This command is to make a FAB contact a chip with fast speed when the distance between the FAB and the chip is relatively large(e.g. >100 μm). It is not for precise gap control. Firstly it pulls back the Z stage by a certain distance(B), then moves the stage(FAB) in the continuous mode toward the chip until the changes the displacement sensor value becomes large than a certain value(Threshold_1). Secondly it pull back the stage, then moves towards the chip step by step with the step of 10 μm until the change in the sensor value is larger than a certain value(Threshold_2). Thirdly it repeats the second process with the step size of 2 μm and the threshold value of "Threshold_3". Finally it pulls back the stages by a certain distance(A) The threshold values are defined in the section "SENSOR" of the file "DeviceInfo.ini".
Syntax :	ROUGH_APPROACH [AXIS],[Backward after approach A],[Back Distance (B)]
Parameter(s) :	[AXIS]: See Name of stages [Backward after approach A]: See the description above. (Unit: μm)

	[Back Distance(B)]: See the description above. (Unit: μm)
Optional Parameter(s) :	[STEPSP]: Stage speed used in the step move (Range:1~5) [CONTSP]: Stage speed used the first contact (Range:1~5) [TWISTBACK]: It rotates ThX stage by 0.5 degree in negative direction before retreating the left/right z stage, and restores the ThX stage position after the retreat. * The units of the distances, and positions are “ μm ” * The units of the angle, and positions are “degree”
Response :	See Table 1.1: Return values of commands.
Example :	ROUGH_APPROACH L_Z,0,200 → It makes a FAB contact a chip with fast speed in the z direction. Firstly it pulls back the z stage by 200 μm , then moves the stage in the continuous mode toward the chip until the changes the displacement sensor value becomes larger than a certain value. Secondly it pulls back the stage, then moves towards the chip step by step with the step of 10 μm until the change in the sensor value is larger than a certain value. Thirdly it repeats the second process with the step size of 2 μm and the threshold value. Finally it pulls back the stages by 0.

1.3.3. APPROACH

APPROACH	
Description :	This command is to control the gap between a FAB and a chip precisely. First it pulls back the Z stage by a certain distance(B). Secondly, it moves the stage toward the chip by a certain step(C) while the changes in the displacement sensor value is within a certain value(Threshold_2), then pull the stage backward with a step size(D) until the sensor value is smaller than (the initial value +Threshold_2). Then it repeats the previous process with a step size(E) and a threshold value(Threshold_3). Finally it pulls back the Z stage by a certain distance(A). [NOTE]: The threshold values are defined in the section “SENSOR” of the file “DeviceInfo.ini”.
Syntax :	APPROACH [STAGE],[BackdistanceAfterApproach (A)],[Back Distance(B)], [Forward_step1(C)],[Backward_step(D)],[Forward_step2(E)]
Parameter(s) :	[STAGE_ASSY]: Name of stage ass'y [BackdistanceAfterApproach (A)]: see description above [Back Distance(B)]: see the description above. [Forward_step1(C)]: see the description above [Backward_step(D)]: see the description above [Forward_step2(E)]: see the description above * The units of the distances, and positions are “ μm ”
Optional Parameter(s) :	[SP]: Speed (1~5) [TWISTBACK]: Only applicable to left/right z stage. The software rotates ThX stage by 0.5 degree in negative direction before retreating the left/right z stage, and restores the ThX stage position after the retreat.
Response :	See Table 1.1: Return values of commands.
Example :	APPROACH RIGHT,50,0,10,10,0.5,SP=3 → It makes the gap between an FAB and a chip to be 0. Firstly it moves the stage toward the chip by 10 μm while the changes in the displacement sensor value is within a certain value(Threshold_2), then pull the stage backward with 10 μm until the sensor value is smaller than (The initial value +Threshold_2). Then it repeats the previous process with 0.5 μm and a threshold value(Threshold_3). Finally it pulls back the stage by 50 μm .

1.3.4. CUSTOM_APPROACH

CUSTOM_APPROACH	
Description :	This command is to control the gap between a FAB and a chip precisely. First it pulls back the Z stage by a certain distance(B). Secondly, it moves the stage toward the chip by a certain step(C) while the changes in the displacement sensor value is within a certain value(Threshold_2), then pull the stage backward with a step size(D) until the sensor value is smaller than (the initial value +Threshold_2). Then it repeats the previous process with a step size(E) and a threshold value(Threshold_3). Finally it pulls back the Z stage by a certain distance(A). [NOTE]: The threshold values are defined in the section “SENSOR” of the file “DeviceInfo.ini”.
Syntax :	CUSTOM_APPROACH [AXIS],[BackdistanceAfterApproach (A)],[Back Distance(B)], [Forward_step1(C)],[Backward_step(D)],[Forward_step2(E)]
Parameter(s) :	[AXIS]: See Name of stages [BackdistanceAfterApproach (A)]: see description above [Back Distance(B)]: see the description above. [Forward_step1(C)]: see the description above [Backward_step(D)]: see the description above [Forward_step2(E)]: see the description above * The units of the distances, and positions are “μm” * The units of the angle, and positions are “degree”
Optional Parameter(s) :	[SP]: Speed (1~5) [TWISTBACK]: Only applicable to left/right z stage. The software rotates ThX stage by 0.5 degree in negative direction before retreating the left/right z stage, and restores the ThX stage position after the retreat.
Response :	See Table 1.1: Return values of commands.
Example :	CUSTOM_APPROACH L_Z,50,0,10,10,0.5,SP=3 → It makes the gap between a FAB and a chip to be 0. Firstly it moves the stage toward the chip by 10 um while the changes in the displacement sensor value is within a certain value(Threshold_2), then pull the stage backward with 10 um until the sensor value is smaller than (The initial value +Threshold_2). Then it repeats the previous process with 0.5 um and a threshold value(Threshold_3). Finally it pulls back the stage by 50 um.

1.3.5. ANGLESEARCH

ANGLESEARCH	
Description :	This command is to make the end faces of a FAB parallel to that of a chip. First it pushes the FAB toward the chip by certain distance, then it performs the angle search with ThY stage first and then with ThX stage by default. For the angle search with a stage, it rotates a goniometer(ThX or ThY) both in the positive direction and the negative direction with the specified angle span while monitoring the displacement sensor, then moves the stage to the angle calculated from the measurement.
Syntax :	ANGLESEARCH [STAGE_ASSY],[Push Distance],[Angle_thy],[Angle_thx]
Parameter(s) :	[STAGE_ASSY]: LEFT/RIGHT [Push Distance]: distance to push FAB [Angle_ThX]: Total rotation angle of ThY stage [Angle_ThY]: Total rotation angle of ThX stage
Optional Parameter(s)	[RETRY]: The maximum number of retry in case of the angle search failure. [THXFIRST]: The angle search starts in ThX direction first with this option. otherwise it starts with ThY direction by default. [SINGLEAXIS]: Do the angle search with only the first axis.

	[SP]: Stage speed used for the ANGLESEARCH. (Range: 1~5) [LOGGING]: Save in a text file the displayed curves of ANGLESEARCH - File path: .\IFA-600-AUCME3\REPORT\ANGLESEARCH_SCAN\ANGLESEARCH_SCAN_yyMMdd_hhmmss.txt [COMP_XOFF]: With this option, it Find peak position with the algorithm which compensate X-axis offset of the rotation center. [VIEW_PEAK]: Display peak position which the algorithm finds on the graph. * Cautions: When the [COMP_XOFF] option is used, the followings need to be considered - The rotation center of a jig should be located outside of the FAB in x direction. - The recommended push distance is about 50 μm. If the push distance is not long enough, the accuracy may decrease. With “VIEW_PEAK” option, the user can check the accuracy of the angle search command with the given parameters - If the push distance is too big, this action may damage the FAB or the chip under process. - If the scan range in ThX direction is too big, it may create “stage stop” situation since the FAB holder is pushed back too much. Therefore keep the scan range parameter small as far as the angle accuracy does not deteriorate.
Response :	See Table 1.1: Return values of commands.
Example :	ANGLESEARCH LEFT,50,3,5

1.3.6. CUSTOM_ANGLESEARCH

CUSTOM_ANGLESEARCH	
Description :	This command is to make the end faces of a FAB parallel to that of a chip. First it pushes the FAB toward the chip by certain distance, then it performs the angle search with selected first stage first and then with selected second stage by default. For the angle search with a stage, it rotates a goniometer both in the positive direction and the negative direction with the specified angle span while monitoring the displacement sensor, then moves the stage to the angle calculated from the measurement.
Syntax :	CUSTOM_ANGLESEARCH [Axis1],[Axis2],[Push axis],[Angle_axis1],[Angle_axis2],[Push Distance],[Sensor channel]
Parameter(s) :	[Axis1]: Name of stages [Axis2]: Name of stages [Push axis]: Name of stages [Angle_axis1]: Total rotation angle of axis1 [Angle_axis2]: Total rotation angle of axis2 [Push distance]: push distance [Sensor channel]: Displacement sensor channel 1: Left 2: Right
Optional Parameter(s)	[RETRY]: The maximum number of retry in case of the angle search failure. [SINGLEAXIS]: Do the angle search with only the first axis. [SP]: Stage speed used for the ANGLESEARCH. (Range: 1~5) [LOGGING]: Save in a text file the displayed curves of ANGLESEARCH File path: .\IFA-600-AUCME3\REPORT\ANGLESEARCH_SCAN\CUSTOM_ANGLESEARCH_SCAN_yyMMdd_hhmmss.txt [COMP_XOFF]: With this option, it Find peak position with the algorithm which compensate X-axis offset of the rotation center. [VIEW_PEAK]: Display peak position which the algorithm finds on the graph.

- * Cautions: When the [COMP_XOFF] option is used, the followings need to be considered
- The rotation center of a jig should be located outside of the FAB in x direction.
 - The recommended push distance is about 50 μm . If the push distance is not long enough, the accuracy may decrease. With “VIEW_PEAK” option, the user can check the accuracy of the angle search command with the given parameters
 - If the push distance is too big, this action may damage the FAB or the chip under process.
 - If the scan range in ThX direction is too big, it may create “stage stop” situation since the FAB holder is pushed back too much. Therefore keep the scan range parameter small as far as the angle accuracy does not deteriorate.

Response : See Table 1.1: Return values of commands.

Example : CUSTOM_ANGLESEARCH L THX,L THZ,L Y,2,2,50,1

1.3.7. XYSEARCH

XYSEARCH	
Description :	This command is for the alignment with a specified channel in X and Y directions. The alignment command consists of two alignment methods, “scan search” and “step search”, and provides many options to optimize the process for a certain situation. The “scan search” measures the power levels at a certain interval while the stage moves over a specified distance (SPAN). This method is a fast process but less accurate. The “step search” repeats the power measurement after moving the stage by a certain step (STEP_MIN, default value 0.4 μm) over the maximum distance of “STEPDIST” (default: 6 μm). This method is relatively slow, but accurate. “XYSEARCH” performs the “scan search” in both X and Y directions to find the peak position roughly, and then performs the “step search” for precise alignment. By default, the scan search in the X direction is followed by that in Y direction, and the order can be swapped with the option “YFIRST”. The SPAN size is STEP_SIZE*MAX_STEP. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	XYSEARCH [STAGE_ASSY],[CHANNEL],[STEP SIZE],[MAX STEP]
Parameter(s) :	[STAGE_ASSY]: LEFT/RIGHT/S1 [CHANNEL]: See Names of powermeter/currentmeter channels [STEP_SIZE]: Step size of the scan search in μm unit [MAX_STEP]: Maximum step allowed during alignment. (NOTE: Only the value “STEP_SIZE*MAX_STEP” is meaningful in the command.)
Optional Parameter(s)	[TRGSP]: Stage speed used for the scan search (Range: 1~5) [STEPSP]: Stage speed used for the step search (Range: 1~5) [STEPMIN]: Step size used for the step search in μm unit (Default: 0.4 μm) [STEPDIST]: Maximum moving distance during the step search in μm unit (Default: 6 μm) [STEPSKIP]: With this option, it skips the step search to increase the processing speed. [MEDSRCH]: With this option, it determines the peak position using the median value. [YFIRST]: With this option, it performs the scan search in the Y direction first.

[NOFASTSCAN]: With this option, it omits the scan search.

[SLEEP=time(ms)]: Waiting time in millisecond unit between steps during the “Step search” phase. (Default: 0 ms)

[SINGLESEARCH]: With this option, it performs XYSEARCH only for the first axis.

[TILT_THX=“tilt_angle”] or [TILT_THXTAN=“tangential”]: With this option, it moves the Z stage and Y stage during “step search” phase in the direction defined by the ThX angle(“tilt_angle”) or by the tilt slope in the ThX direction.

[LOGGING]: With this option, the displayed XYSEARCH scan curves are saved to a text file.

- File path: \IFA-600-AUCME3\REPORT\XYSEARCH_SCAN\XYSEARCH_SCAN_yyMMdd_hhmmss.txt

[BEYONDMAX=“cnt”]: With this option, stage movement is stopped after “cnt” consecutive decreases in power during the “step search”.

[REFRESH_AT_END]: With this option, the program refreshes the power readings in the data monitoring window between the internal alignment steps.

[FINDPEAK]: With this option, the position with the highest measured power is selected during the “step search”. ([Warning]: FINDPEAK and POLY5 cannot be used simultaneously.)

[POLY5]: With this option, the peak position is calculated by polynomial fitting using a 2nd-order polynomial with five points near the highest power point during the “step search”. ([Warning]: FINDPEAK and POLY5 cannot be used simultaneously.)

[SHOW_TARGET]: With this option, it displays the peak position calculated from the measurement data on the graph after the “step search”.

[INTV_BY_STEP]: With this option, the graph display and data logging intervals are determined based on the STEP_SIZE. STEP_SIZE is a parameter used for the scan search. If the specified value is not a multiple of the stage resolution, the value is adjusted to the nearest multiple of the stage resolution. As the STEP_SIZE is adjusted, the search distance may be reduced accordingly.

[PM_RANGE_UPPER=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W.

[PM_RANGE_UPPER_DBM=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.

Response : See Table 1.1: Return values of commands.

Example : XYSEARCH LEFT,PM1,0.08,500,STEPSKIP,INTV_BY_STEP,LOGGING
→ It performs XYSEARCH on the LEFT using channel PM1. The scan search is executed with a step size of 0.08 μm and a scan range of 400 μm, and the step search is skipped. The graph display and data logging intervals are determined based on the step size, and the scan data is saved to a file.

XYSEARCH LEFT,PM1,1,30,YFIRST,MEDSRCH,SLEEP=200
→ It performs XYSEARCH on the LEFT using channel PM1. The scan search is executed in the Y direction first with a step size of 1 μm and a scan range of 30 μm. The peak position is determined based on the median value, and a waiting time of 200 ms is applied between steps to stabilize the measurement.

1.3.8. XYBLINDSEARCH

XYBLINDSEARCH

Description : This command is for the optical alignment in X & Y directions between a FAB and a chip when the waveguides in both sides are not aligned at all thereby the coupled power level is not detectable. It scans the area of “XSPAN × YSPAN”, and find the position with the maximum coupling.

Firstly it moves the stages by XSPAN/2 and YSPAN/2 in -X and -Y directions, respectively. Secondly, it moves the stages in +X direction by XSPAN while measuring the

power, then moves the stages by a certain distance(Step) in +Y direction. Thirdly, it moves the stages in -X direction by XSPAN, then moves the stages by a certain step(Step) in +Y direction.

It repeats the second and the third process until the total moving distance in Y direction reaches YSPAN. Finally it finds the maximum coupling position and move the X and Y stages to the position.

* When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option.

An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy.

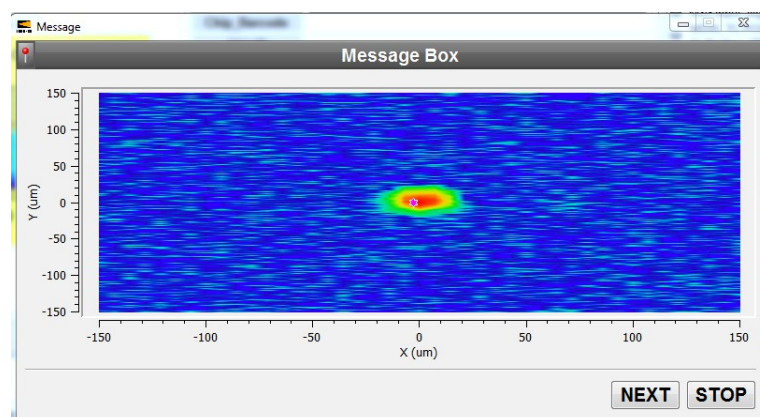
If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.

NOTE:

Despite of the name “XYBLINDSERACH”, this command can use any stages with the options of “TRG_STAGE” and “STEP_STAGE”.

Syntax :	XYBLINDSEARCH [STAGE_ASSY],[CHANNEL],[XSPAN],[YSPAN],[STEP]
Parameter(s) :	[STAGE_ASSY]: LEFT/RIGHT/S1 [CHANNEL]: See Names of powermeter/currentmeter channels [XSPAN]: See the description above (Unit: μm) [YSPAN]: See the description above (Unit: μm) [STEP]: See the description above (Unit: μm)
Optional Parameter(s) :	[LINEAR]: With this option, it draws Z values of the graph in linear scale (Default: dB scale) [YTRIG]: With this option, it swaps the XY axes to used for the blind-search [COARSESCAN]: With this option, it performs coarse search. [MANUALFIND]: The program displays a contour graph after finishing 2D scan, with which a user can pick a position to move the stage assy under control to the corresponding position. This option is useful when the 2d profile is too complex to be processed automatically. [AUTONEXT=Time(s)]: With this option, it automatically presses “NEXT” button when the time expires. [LOGGING]: With this option, it saves in a text file the displayed curves of XYBLIND-SEARCH File path: \IFA-600-AUCME3\REPORT\BLIND_FULLSCAN\BLIND_FULLSCAN_yyMMdd_hhm mss.txt [USE_DAQ]: With this option, it performs alignment process with the analog input channel of SC7000. (DAQ3 or DAQ4) [INTV_BY_STEP]: With this option, the graph display and 2d data logging intervals are determined based on the STEP. If the specified value is not a multiple of the stage resolution, the value is adjusted to the nearest multiple of the stage resolution. In this command, the graph display interval is internally set to four times the calculated STEP-based interval. This behavior is defined by internal processing policy and does not effect the actual search operation. [PM_RANGE_UPPER=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm. ----- [TRG_STAGE]: Set the stage to be used for “trigger scan”(Default stage: X axis) [STEP_STAGE]: Set the stage to be used for “step move”(Default stage: Y axis)

- Stage Names:
See Name of stages



Response : See Table 1.1: Return values of commands.

Example :
 XYBLINDSEARCH LEFT,PM1,100,100,1 → Channel 1 powermeter X, Y 100 μ m scan.
 XYBLINDSEARCH LEFT,PM1,100,100,1,TRG_STAGE=L_THX,STEP_STAGE=L_THY → Channel 1 powermeter L_THX, L_THY 100 μ m scan.
 XYBLINDSEARCH LEFT,DAQ3,100,100,1,USE_DAQ → X, Y 100 μ m scan with analog input channel #3.

1.3.9. STEPSEARCH

STEPSEARCH	
Description :	This command is for the alignment with a specified channel in a single direction. The search repeats the power measurement after moving a stage by a certain step over the scan range. This method is relatively slow, but accurate. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	STEPSEARCH [STAGE],[CHANNEL],[STEP_SIZE],[RANGE]
Parameter(s) :	[STAGE_ASSY]: See Name of stages [CHANNEL]: See Names of powermeter/currentmeter channels [STEP_SIZE]: Step size of the scan search in μm unit or degree. The maximum step size for the goniometer (Gonio stage) is 3 degree. [RANGE]: Total scan length. The maximum value for the goniometer is 3 degree.
Optional Parameter(s) :	[STEPSP]: Stage speed used for the step search (Range: 1~5) [SLEEP=time(s)]: Waiting time in ms unit between steps in “Step search” phase (Default: 0 ms) [BEYONDMAX=“cnt”]: With this option, the moving stops after “cnt” times of consecutive decreases in power on processing “step search”. [REFRESH_AT_END]: With this option, the program refreshes the power readings in the data monitoring window between the internal alignment steps. [FINDPEAK]: With this option, it finds the position which get highest power on processing

	“step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [POLY5]: With this option, it finds the peak position in fit data with 2nd polynomial using the 5 points near the highest point on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [SHOW_TARGET]: With this option, it displays the peak position calculated by measurement data on the graph. [TILT_THX=“tilt_angle”] or [TILT_THXTAN=“tangential”]: With this option, it moves the z stage and y stages in the direction with the ThX of “tilt_angle” or the tilt slope in ThX direction of “tangential”. [PM_RANGE_UPPER=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response :	See Table 1.1: Return values of commands.
Example :	STEPSEARCH L_Z,PM1,1,30 STEPSEARCH L_Z,PM1,1,30,SLEEP=200

1.3.10. STEPSEARCH_DAQ

STEPSEARCH	
Description :	This command is for the alignment with a specified analog input channel in a single direction. The search repeats the power measurement after moving a stage by a certain step over the scan range. This method is relatively slow, but accurate.
Syntax :	STEPSEARCH [STAGE],[CHANNEL],[STEP_SIZE],[RANGE]
Parameter(s) :	[STAGE_ASSY]: See Name of stages [CHANNEL]: DAQ3 or DAQ4 [STEP_SIZE]: Step size of the scan search in μm unit or degree. The maximum step size for the goniometer (Gonio stage) is 3 degree. [RANGE]: Total scan length. The maximum value for the goniometer is 3 degree.
Optional Parameter(s) :	[STEPSP]: Stage speed used for the step search (Range: 1~5) [SLEEP=time(s)]: Waiting time in ms unit between steps in “Step search” phase (Default: 0 ms) [BEYONDMAX=“cnt”]: With this option, the moving stops after “cnt” times of consecutive decreases in power on processing “step search”. [REFRESH_AT_END]: With this option, the program refreshes the power readings in the data monitoring window between the internal alignment steps. [FINDPEAK]: With this option, it finds the position which get highest power on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [POLY5]: With this option, it finds the peak position in fit data with 2nd polynomial using the 5 points near the highest point on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [SHOW_TARGET]: With this option, it displays the peak position calculated by measurement data on the graph. [TILT_THX=“tilt_angle”] or [TILT_THXTAN=“tangential”]: With this option, it moves the z stage and y stages in the direction with the ThX of “tilt_angle” or the tilt slope in ThX direction of “tangential”.
Response :	See Table 1.1: Return values of commands.
Example :	STEPSEARCH_DAQ L_Z,DAQ3,1,30 STEPSEARCH_DAQ L_Z,DAQ3,1,30,SLEEP=200

1.3.11. GROUP_STEPSEARCH

GROUP_STEPSEARCH

Description :	This command performs the “STEPSEARCH” moving two stages simultaneously with the same step condition.
Syntax :	GROUP_STEPSEARCH [STAGE1],[STAGE2],[CHANNEL],[STEP_SIZE],[RANGE]
Parameter(s) :	[STAGE1]: See Name of stages [STAGE2]: See Name of stages [CHANNEL]: See Names of powermeter/currentmeter channels [STEP_SIZE]: Step size of the scan search in μm unit or degree. The maximum step size for the goniometer (Gonio stage) is 3 degree. [RANGE]: Total scan length. The maximum value for the goniometer is 3 degree.
Optional Parameter(s) :	[STEPSP]: Stage speed used for the step search (Range: 1~5) [SLEEP=time(s)]: Waiting time in ms unit between steps in “Step search” phase (Default: 0 ms) [BEYONDMAX=“cnt”]: With this option, the moving stop after “cnt” times of consecutive decreases in power on processing “step search”. [REFRESH_AT_END]: With this option, the program refresh the power readings in the data monitoring window between the internal alignment steps. [FINDPEAK]: With this option, it finds the position which get highest power on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [POLY5]: With this option, it finds the peak position in fit data with 2 nd polynomial using the 5 points near the highest point on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [TILT_THX=“tilt_angle”] or [TILT_THXTAN=“tangential”]: With this option, step search moves the z stage and y stages in the direction with the ThX of “tilt_angle” or the tilt slope in ThX direction of “tangential” [SHOW_TARGET]: With this option, it displays the peak position calculated by measurement data on the graph.
Response :	See Table 1.1: Return values of commands.
Example :	GROUP_STEPSEARCH L_X,R_X,PM1,1,30 GROUP_STEPSEARCH L_X,R_X,PM1,1,30,SLEEP=200

1.3.12. ZOPTIMIZE

ZOPTIMIZE	
Description :	This command is prepared to maximize the coupling in z-direction. It performs “XYSEARCH” for every step with the step size of “STEP_SIZE” until step count in z direction reaches “MAX_STEP”. Then it moves the z stage to the maximum coupling position. By default, the command moves the z stage in the negative direction in order to avoid possible collision. “FWD” option reverses the moving direction, which need to be used with great caution to avoid collisions. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	ZOPTIMIZE [STAGE_ASSY],[STEP_SIZE],[MAX_STEP],[CHANNEL]
Parameter(s) :	[Stage Ass’y]: LEFT/RIGHT/S1 [STEP_SIZE]: Step size in Z-direction (Unit: μm) [MAX_STEP]: Maximum number of step count [CHANNEL]: See Names of powermeter/currentmeter channels

Optional Parameter(s) :	[STEP="step"]: With this option, it sets the alignment step in μm unit. (Default: 1.0 μm) [MAXSTEP="max step"]: Maximum step allowed during alignment. (Default: 50 μm) [FWD]: With this option, it moves the z stage in positive direction. [SP]: Speed (Range: 1~5) [SKIPSEARCH or SKIPXY]: With this option, it skips XY alignment except for first and last XY alignment. [TILT_THX="tilt_angle"] or [TILT_THXTAN="tangential"]: With this option, step search moves the z stage and y stages in the direction with the ThX of "tilt_angle" or the tilt slope in ThX direction of "tangential". [ZBEYONDMAX="zcnt"]: With option, the command stop after "zcnt" times of consecutive decreases in power. [GROUP_SEARCH="2ndAssy"] With this option, this command controls two stage assemblies("STAGE ASSY" and "2ndAssy") for the optimization. This option is useful for the U-waveguide alignment. [BEYONDMAX="cnt"]: With this option, the moving stop after "cnt" times of consecutive decreases in power on processing "step search". [REFRESH_AT_END]: With this option, the program refresh the power readings in the data monitoring window between the internal alignment steps. [FINDPEAK]: Find the position which get highest power on processing "step search". ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [POLY5]: With this option, it finds the peak position in fit data with 2nd polynomial using the 5 points near the highest point on processing "step search". ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [SHOW_TARGET]: With this option, it displays the peak position calculated by measurement data on the graph after the "step search". [PM_RANGE_UPPER="power"]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM="power"]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response:	See Table 1.1: Return values of commands.
Example:	ZOPTIMIZE LEFT,1,50,PM1 ZOPTIMIZE LEFT,1,50,PM1,GROUP_SEARCH=RIGHT ZOPTIMIZE LEFT,1,50,PM1,ZBEYONDMAX=2 → It performs "z-optimization" with the LEFT stage assembly with the step size of 1 μm in z direction and the maximum steps of 50. If the measured power after the alignment drops two consecutive steps, it stops taking additional steps and moves the stage to the position with the maximum measured power value.

1.3.13. CUSTOM_OPTIMIZE

CUSTOM_OPTIMIZE	
Description :	This command is prepared to maximize the coupling in specific direction. It performs "PEAK_SEARCH" for every step with the step size of "STEP_SIZE" in [Axis1] and [Axis2] until step count in [Axis3] direction reaches "MAX_STEP". Then it moves the [Axis3] to the maximum coupling position. By default, the command moves the [Axis3] stage in the negative direction in order to avoid possible collision. "FWD" option reverses the moving direction, which need to be used with great caution to avoid collisions. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is

	automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	CUSTOM_OPTIMIZE [Axis1],[Axis2],[Axis3],[STEP_SIZE],[MAX_STEP],[CHANNEL]
Parameter(s) :	[Axis1]: Name of stages [Axis2]: Name of stages [Axis3]: Name of stages [STEP_SIZE]: Step size in [Axis3] (Unit: μm) [MAX_STEP]: Maximum number of step count [CHANNEL]: See Names of powermeter/currentmeter channels
Optional Parameter(s) :	[STEP="step"]: With this option, it sets the alignment step in μm unit. (Default: 1.0 μm) [MAXSTEP="max step"]: Maximum step allowed during alignment. (Default: 50 μm) [FWD]: With this option, it moves the z stage in positive direction. [SP]: Speed (Range: 1~5) [SKIPSEARCH or SKIPXY]: With this option, it skips XY alignment except for first and last XY alignment. [TILT_THX="tilt_angle"] or [TILT_THXTAN="tangential"]: With this option, step search moves the z stage and y stages in the direction with the ThX of "tilt_angle" or the tilt slope in ThX direction of "tangential". [OPTIMIZE_BEYONDMAX="zcnt"]: With option, the command stop after "zcnt" times of consecutive decreases in power. [BEYONDMAX="cnt"]: With this option, the moving stop after "cnt" times of consecutive decreases in power on processing "step search". [REFRESH_AT_END]: With this option, the program refresh the power readings in the data monitoring window between the internal alignment steps. [FINDPEAK]: Find the position which get highest power on processing "step search". ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [POLY5]: With this option, it finds the peak position in fit data with 2 nd polynomial using the 5 points near the highest point on processing "step search". ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [SHOW_TARGET]: With this option, it displays the peak position calculated by measurement data on the graph after the "step search". [PM_RANGE_UPPER="power"]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM="power"]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response:	See Table 1.1: Return values of commands.
Example:	CUSTOM_OPTIMIZE L_X,L_Y,L_Z,1,50,PM1 CUSTOM_OPTIMIZE L_X,L_Y,L_Z,1,50,PM1 CUSTOM_OPTIMIZE L_X,L_Z,L_Y,1,50,PM2,ZBEYONDMAX=2 → It performs "optimization" with the step size of 1 μm in L_Y direction and the maximum steps of 50. If the measured power after the alignment drops two consecutive steps, it stops taking additional steps and moves the stage to the position with the maximum measured power value.

1.3.14. PEAK_SEARCH

PEAK_SEARCH	
Description :	This command is an integrated version of "XYSEARCH" and "XYSEARCH_Daq". Like "XYSEARCH", this command consists of two alignment methods, "scan search" and "step search", and provides many options to optimize the process for a certain situation. The difference is this command takes the stages to be used as parameters, while XYSEARCH only used X axis and Y axis.

	* When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	PEAK_SEARCH [STAGE1],[STAGE2],[CHANNEL],[STEP_SIZE],[MAX_STEP]
Parameter(s) :	[STAGE1]: See Name of stages [STAGE2]: See Name of stages [CHANNEL]: See Names of powermeter/currentmeter channels [STEP_SIZE]: Step size of the scan search in μm unit. [MAX_STEP]: Maximum step allowed during alignment. [NOTE]: Only the value “STEP_SIZE*MAX_STEP” is meaningful in the command.
Optional Parameter(s) :	[TRGSP]: Stage speed used for the scan search. (Range: 1~5) [STEPSP]: Stage speed used for the step search. (Range: 1~5) [STEPMIN]: The step size used for the step search in μm unit. (Default: 0.4 μm) [STEPDIST]: The maximum distance to move during the step search in μm unit. (Default: 6 μm) [STEPSKIP]: With this option, it skips the step search for speed. [MEDSRCH]: With this option, it finds the peak position using median value. [YFIRST]: With this option, it performs the scan search with in Y direction first. [NOFASTSCAN]: With this option, it omits the scan search. [SLEEP=time(s)]: Waiting time in ms unit between steps in “Step search” phase. (Default: 0 ms) [USE_DAQ]: With this option, it performs alignment process with the analog input channel of SC7000. [SINGLESEARCH]: With this option, it performs XYSEARCH only for first axis. [BEYONDMAX=“cnt”]: With this option, the moving stops after “cnt” times of consecutive decreases in power on processing “step search”. [REFRESH_AT_END]: With this option, the program refreshes the power readings in the data monitoring window between the internal alignment steps. [FINDPEAK]: With this option, it finds the position which get highest power on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [POLY5]: With this option, it finds the peak position in fit data with 2nd polynomial using the 5 points near the highest point on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [SHOW_TARGET]: With this option, it displays the peak position calculated by measurement data on the graph after the “step search”. [PM_RANGE_UPPER=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response :	See Table 1.1: Return values of commands.
Example :	PEAK_SEARCH L_X,L_Y, PM1,1,50 PEAK_SEARCH L_X,L_Y,DAQ3,1,50,USE_DAQ

1.3.15. XYBALANCESEARCH

XYBALANCESEARCH	
Description :	It moves the selected stage assembly by a certain step to find the position where the

	power product of the selected channels becomes the maximum value. The command performs the maximum finding both in X axis and Y axis by default. The option “SELSTAGE” can limits the stages for this command. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	XYBALANCESEARCH [ASSY],[Channel1],[Channel2],[StepSize],[MaxStep]
Parameter(s) :	[Stage Ass’y]: LEFT/RIGHT/S1 [Channel1]: See Names of powermeter/currentmeter channels [Channel2]: See Names of powermeter/currentmeter channels [StepSize]: Step size in μm unit [Maxstep]: Maximum number of steps
Optional Parameter(s) :	[SP]: Speed (1~5) [SELSTAGE]: “X”: X axis only, “Y”: Y axis only [PM_RANGE_UPPER=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response :	See Table 1.1: Return values of commands.
Example :	XYBALANCESEARCH LEFT,PM1,PM2,1,10 → It searches for the maximum average power in increments of 1um in the left stage in each direction.

1.3.16. CHANNEL_OPTIMIZE

CHANNEL_OPTIMIZE	
Description :	The command performs alignment in the way that the optimization criterion value of a given algorithm becomes maximized. Default method first scans in X direction, and move the position where the multiplications of all selected channels is the maximum value, then do the same thing in Y direction. For debug purpose only, an optional parameter “FULLSCAN” is prepared. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	CHANNEL_OPTIMIZE [STAGE_ASSY],[CHANNEL_RANGE],[STEP],[MAXSTEP]
Parameter(s) :	[Stage Ass’y]: LEFT/RIGHT/S1 [CHANNEL_RANGE]: See Names of powermeter/currentmeter channels (PM startChannel – PM endChannel, or PM channel PM channel) [STEP]: The alignment step size in μm unit [MAXSTEP]: Maximum number of the step allowed during alignment
Optional Parameter(s) :	[FULLSCAN]: With this option, it scans XY plane with a given step size(‘step’) and span of “step*max_step” in each direction, and save the scan data into a file. The file name in the format of

	“./REPORT/SAVEDATA/CHANNEL_OPTIMIZE_FULLSCAN_YYYYMMDDhhmmss” unless explicitly specified through the option “NAME”. [NAME = “File Name”]: The name of data file [METHOD=POS]: With this option, it finds peak positions of channels, and move to the position that the distance from peak positions is minimized. [SP]: Stage speed used for the CHANNEL_OPTIMIZE. Range: 1~5 [BEYONDMAX]: The number of steps to move the stage after the average value passed the maximum value. (Default: 2) [PM_RANGE_UPPER=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response :	See Table 1.1: Return values of commands.
Example :	CHANNEL_OPTIMIZE LEFT, PM1-PM4, 0.1, 10 → It performs the default method using PM1 ~ PM4. CHANNEL_OPTIMIZE LEFT, PM1 PM2 PM4, 0.1, 10 → It performs the default method using PM1, PM2, PM4. CHANNEL_OPTIMIZE LEFT, PM1-PM4, 0.1, 10, FULLSCAN → It scans 10 μm x 10 μm area with 0.5 μm step, and save the power readings of PM1 ~ PM4 in a file. The center of the scan area is set by the stage position when the command issued. CHANNEL_OPTIMIZE LEFT, PM1-PM4, 2, 20, BEYONDMAX=5 → It scans +/- 40 μm from the current position with 2 μm step in X/Y directions, then find and move the peak position of the average value of PM1, PM2, PM3, and PM4. If the average power value is lower than previous maximum value 5 times continuously, then it is stopped and moves the stage to the peak position.

1.3.17. CUSTOM_CHANNEL_OPTIMIZE

CUSTOM_CHANNEL_OPTIMIZE	
Description :	The command performs alignment in the way that the optimization criterion value of a given algorithm becomes maximized. Default method first scans in X direction, and moves the position where the multiplications of all selected channels is the maximum value, then do the same thing in Y direction. For debug purpose only, an optional parameter “FULLSCAN” is prepared. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	CUSTOM_CHANNEL_OPTIMIZE [STAGE_AXIS1],[STAGE_AXIS2],[CHANNEL_RANGE],[STEP],[MAXSTEP]
Parameter(s) :	[STAGE_AXIS1]: Name of stages [STAGE_AXIS2]: Name of stages [CHANNEL_RANGE]: See Names of powermeter/currentmeter channels (PM startChannel – PM endChannel, or PM channel PM channel) [STEP]: The alignment step size in μm unit [MAXSTEP]: Maximum number of the step allowed during alignment
Optional Parameter(s) :	[FULLSCAN]: With this option, it scans XY plane with a given step size(‘step’) and span of “step*max_step” in each direction, and save the scan data into a file. The file name in

	the format of <code>“./REPORT/SAVEDATA/CHANNEL_OPTIMIZE_FULLSCAN_YYYYMMDDhhmmss”</code> unless explicitly specified through the option “NAME”. [NAME = “File Name”]: The name of data file [METHOD=POS]: With this option, it finds peak positions of channels, and move to the position that the distance from peak positions is minimized. [SP]: Stage speed used for the CHANNEL_OPTIMIZE. Range: 1~5 [BEYONDMAX]: The number of steps to move the stage after the average value passed the maximum value. (Default: 2) [PM_RANGE_UPPER=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response :	See Table 1.1: Return values of commands.
Example :	<code>CUSTOM_CHANNEL_OPTIMIZE L_Y,L_Z, PM1-PM4, 0.1, 10</code> → It performs the default method using PM1 ~ PM4. <code>CUSTOM_CHANNEL_OPTIMIZE L_X,L_Y,PM1 PM2 PM4, 0.1, 10</code> → It performs the default method using PM1, PM2, PM4. <code>CUSTOM_CHANNEL_OPTIMIZE L_Z,L_X,PM1-PM4, 0.1, 10, FULLSCAN</code> → It scans 10 μm x 10 μm area with 0.5 μm step, and save the power readings of PM1 ~ PM4 in a file. The center of the scan area is set by the stage position when the command issued. <code>CUSTOM_CHANNLE_OPTIMIZE L_X,L_Y,PM1-PM4,2,20,BEYONDMAX=5</code> → It scans +/- 40 μm from the current position with 2 μm step in X/Y directions, then find and move the peak position of the average value of PM1, PM2, PM3, and PM4. If the average power value is lower than previous maximum value 5 times continuously, then it is stopped and moves the stage to the peak position.

1.3.18. ROLLSEARCH

ROLLSEARCH	
Description :	This command is for the roll alignment. First it finds the position that gives the maximum power at the first channel(CH_1), then does the same thing with the second channel(CH_N). From the two sets of positions obtained, it calculates the angle in the roll direction(ThZ), and move the ThZ stage to the angle. Finally it performs XY alignment with the first channel. It uses “XYSEARCH” internally for the alignment in XY plane. Refer to “XYSEARCH” section for more information on the parameters. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	<code>ROLLSEARCH [STAGE],[CH_1],[CH_N],[CHAN_SEP],[SCAN_DIST]</code>
Parameter(s) :	[Stage Ass’y]: Name of stage ass’y [CH_1]: See Names of powermeter/currentmeter channels [CH_N]: See Names of powermeter/currentmeter channels [CHAN_SEP]: Distance in μm unit between channel 1 and channel N. [SCAN_DIST]: Distance used for “XYSEARCH” command in μm unit (WARNING: This value must be an INTEGER value)

Optional Parameter(s) :	[TRGSP]: Stage speed used for the scan search. (Range: 1~5) [STEPSP]: Stage speed used for the step search. (Range: 1~5) [DEGSTEP]: Maximum step size that ThZ axis moves at a time. Default value depends on the channel separation, 1.0 degree for <1000 μm, 0.4 degree for 1000 μm~2000 μm, 0.2 degree for 2000~3000 μm, and 0.1 degree for >3000 μm. [STEPSKIP]: With this option, it skips the step search for fast alignment. [MEDSRCH]: With this option, it finds the peak position using median value. [YFIRST]: With this option, it performs the scan search in Y-axis first. [NOFASTSCAN]: With this option, it does not use SCAN method for searching. [SHIFT_RC_X]: With this option, it shifts the x coordinate of the rotation center by the given values. [SHIFT_RC_Y]: With this option, it shifts the y coordinate of the rotation center by the given values. [BEYONDMAX="cnt"]: With this option, the moving stops after "cnt" times of consecutive decreases in power on processing "step search". [REFRESH_AT_END]: With this option, the program refreshes the power readings in the data monitoring window between the internal alignment steps. [FINDPEAK]: Find the position which get highest power on processing "step search". ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [POLY5]: Find the peak position in fit data with 2nd polynomial using the 5 points near the highest point on processing "step search". ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [PM_RANGE_UPPER="power"]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM="power"]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response :	See Table 1.1: Return values of commands.
Example :	ROLLSEARCH RIGHT,PM1,PM2,1750,2,YFIRST

1.3.19. CUSTOM_ROLLSEARCH

CUSTOM_ROLLSEARCH	
Description :	This command is for the ROLL alignment. First, it finds the position that gives the maximum power at the first channel(CH_1), then does the same thing with the second channel(CH_N). From the two sets of positions obtained, it calculates the angle in the roll direction([Rotation Axis]), and move the ThZ stage to the angle. Finally it performs PEAK_SEARCH in [Axis1] and [Axis2] with the first channel. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed. It uses "PEAK_SEARCH" internally for the alignment in [Axis1] and [Axis2] plane. Refer to "PEAK_SEARCH" section for more information on the parameters.
Syntax :	CUSTOM_ROLLSEARCH [Axis1],[Axis2],[Rotation Axis],[CH_1],[CH_N],[CHAN_SEP],[SCAN_DIST]
Parameter :	[Axis1]: Name of stages [Axis2]: Name of stages [Rotation Axis]: Roll axis direction (Name of stages)

	[CH_1]: Channel (See Names of powermeter/currentmeter channels) [CH_N]: Channel (See Names of powermeter/currentmeter channels) [CHAN_SEP]: Distance in μm unit between channel 1 and channel N. [SCAN_DIST]: Distance used for “PEAK_SEARCH” command in μm unit (WARNING: This value must be an INTEGER value)
Optional Parameter(s)	[TRGSP]: Stage speed used for the scan search. (Range: 1~5) [STEPSP]: Stage speed used for the step search. (Range: 1~5) [DEGSTEP]: Maximum step size that rotation axis moves at a time. Default value depends on the channel separation, 1.0 degree for <1000 μm , 0.4 degree for 1000 μm ~2000 μm , 0.2 degree for 2000~3000 μm , and 0.1 degree for >3000 μm . [STEPSKIP]: With this option, it skips the step search for fast alignment. [MEDSRCH]: With this option, it finds the peak position using median value. [NOFASTSCAN]: With this option, it does not use SCAN method for searching. [SHIFT_RC_FIRST]: With this option, it shifts the x coordinate of the rotation center by the given values. [SHIFT_RC_SECOND]: With this option, it shifts the y coordinate of the rotation center by the given values. [BEYONDMAX=“cnt”]: With this option, the moving stops after “cnt” times of consecutive decreases in power on processing “step search”. [REFRESH_AT_END]: With this option, the program refreshes the power readings in the data monitoring window between the internal alignment steps. [FINDPEAK]: Find the position which get highest power on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [POLY5]: Find the peak position in fit data with 2 nd polynomial using the 5 points near the highest point on processing “step search”. ([Warning]: FINDPEAK and POLY5 can not be used simultaneously.) [PM_RANGE_UPPER=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in W. [PM_RANGE_UPPER_DBM=“power”]: Sets the upper measurement range of the Power Meter for the alignment operation. The value is specified in dBm.
Response :	See Table 1.1: Return values of commands.
Example :	CUSTOM_ROLLSEARCH L_X,L_Z,L_THY,PM1,PM2,1750,2

1.3.20. U_ROLLSEARCH

U_ROLLSEARCH	
Description :	This command is for roll alignment of the u_waveguide. First, it moves the stage of selected stage ass’y of “Assy” to half of “ScanRange” in negative direction. Then, it performs “XYSEARCH” for every step with the step size of “StepDegree” in the direction over the scan range of “ScanRange”. Then it moves the stage ass’y to the maximum coupling. * When using a Thorlabs PM5020 Power Meter, it is recommended to specify the measurement range using either the [PM_RANGE_UPPER] or [PM_RANGE_UPPER_DBM] option. An inappropriate measurement range may cause saturation or inaccurate power readings, which may affect the alignment accuracy. If neither option is specified, the measurement range configured before the command is automatically applied during alignment and restored after the alignment operation is completed.
Syntax :	U_ROLLSEARCH [Stage Ass’y],[Channel],[StepDegree],[ScanRange],[XYSearch_Step],[XYSearch_MaxStep]
Parameter(s) :	[Stage Ass’y]: Name of stage ass’y [CHANNEL]: See Names of powermeter/currentmeter channels [StepDegree]: Step size of the scan search in degree unit.

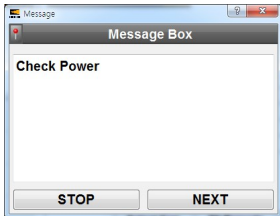
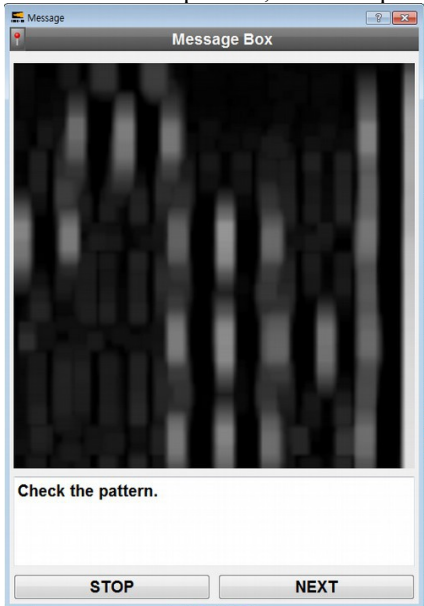
	[ScanRange]: Total scan length in degree unit. [XYSearch_Step]: Step size of the scan search in μm unit for “XYSearch”. [XYSearch_MaxStep]: Maximum step allowed during alignment for “XYSearch”. [NOTE]: Only the value STEP_SIZE*MAX_STEP” is meaningful in the command.
Optional Parameter(s)	[SP]: Speed (1~5) [SKIP_MOVE_HALFSPAN]: With this option, it skip the first movement of thz stage. [SHIFT_RC_X]: With this option, it shifts the x coordinate of the rotation center by the given values. [SHIFT_RC_Y]: With this option, it shifts the y coordinate of the rotation center by the given values. [THZ_FINDPEAK]: With this option, it finds the position which get highest power. ([Warning]: THZ_FINDPEAK and THZ_POLY5 can not be used simultaneously.) [THZ_POLY5]: With this option, it finds the peak position in fit data with 2nd polynomial using the 5 points near the highest point. ([Warning]: THZ_FINDPEAK and THZ_POLY5 can not be used simultaneously.) [ROLL_AXIS]: The name of the stage to be used for the u_roll search [XYSEARCH_FIRST_AXIS]: The name of the stage to be used for the scan search [XYSEARCH_SECOND_AXIS]: The name of the stage to be used for the scan search [TRGSP]: see “XYSEARCH” section [STEPSP]: see “XYSEARCH” section [STEPMIN]: see “XYSEARCH” section [STEPDIST]: see “XYSEARCH” section [MEDSRCH]: see “XYSEARCH” section [YFIRST]: see “XYSEARCH” section [NOFASTSCAN]: see “XYSEARCH” section [STEPSKIP]: see “XYSEARCH” section [SINGLESEARCH]: see “XYSEARCH” section [PM_RANGE_UPPER=“power”]: see “XYSEARCH” section [PM_RANGE_UPPER_DBM=“power”]: see “XYSEARCH” section
Response :	See Table 1.1: Return values of commands.
Example :	U_ROLLSEARCH LEFT,PM1,0.1,1,1,50 U_ROLLSEARCH LEFT,CM1,0.1,1,1,30,ROLL_AXIS=ThY, XYSEARCH_FIRST_AXIS=Z, XYSEARCH_SECOND_AXIS=X

1.3.21. SIDECHECK

SIDECHECK	
Description :	This command is to move the specified FAB channel to the outmost channel of the chip It moves X axis by a specified step until the reading of the specified channel is below the threshold value, and then return 1 step back. If the first power reading of the specified channel is below the threshold, the command stops immediately and returns an error code
Syntax :	SIDECHECK [STAGE_ASSY],[CHANNEL],[DISTANCE],[THRESHOLD],[MAXSTEP]
Parameter :	[STAGE_ASSY]: LEFT/RIGHT/S1 [CHANNEL]: See Names of powermeter/currentmeter channels [DISTANCE]: step size in μm unit [THRESHOLD]: Threshold value in power reading [MAXSTEP]: Maximum number of step to try
Response :	See Table 1.1: Return values of commands.
Example :	SIDECHECK RIGHT,PM1,-250,-30,10

1.4. Message commands

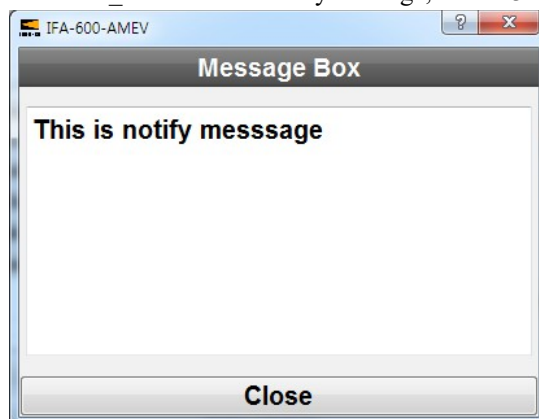
1.4.1. MSG

MSG	
Description :	This command is used to load the message window to display a text message in the message box. It holds the progress of the sequence while message window is open.
Syntax :	MSG ["MESSAGE"]
Parameter(s) :	["MESSAGE"]: A text message to show in the window
Optional Parameter(s) :	[AUTONEXT=Time(s)]: With this option, it automatically presses "NEXT" button when the time expires. [IMAGE="file name"]: With this option, it shows the image with the "file name" in the message window. The file path of the image is "./Image" folder under the IFA-600-AUCME3 folder, and the "file name" should include the extension name too. For example "MSG message string, IMAGE=test.png" displays ./Image/test.png file.
Response :	See Table 1.1: Return values of commands.
Example :	MSG Check Power  MSG Check the pattern., IMAGE=pattern.png 

1.4.2. SPLASH_MSG

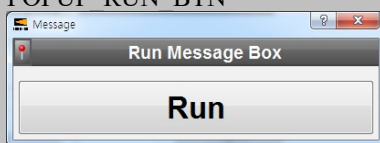
SPLASH_MSG	
Description :	This command displays a message window for a certain period of time.
Syntax :	SPLASH_MSG ["MESSAGE"]
Parameter(s) :	["MESSAGE"]: Text message to show in the window
Optional Parameter(s) :	[TIME=time(s)]: Window display time in second. (Default: 5 seconds)

Response :	See Table 1.1: Return values of commands.
Example :	SPLASH_MSG This is notify message,TIME=3 → It displays a message for 3 seconds.



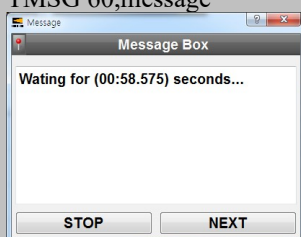
1.4.3. RUNMSG / POPUP_RUN_BTN

RUNMSG / POPUP_RUN_BTN	
Description :	This command creates a message window with a button as shown below. The user can start operation by pressing the button.
Syntax :	RUNMSG
Parameter(s) :	NONE
Response :	See Table 1.1: Return values of commands.
Example :	RUNMSG POPUP_RUN_BTN

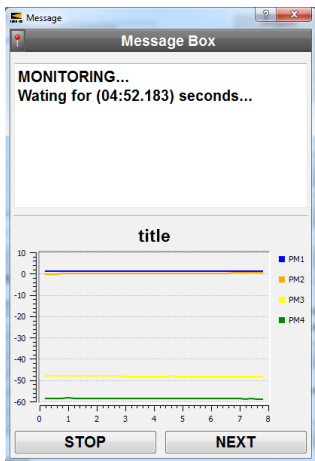


1.4.4. TMSG

TMSG	
Description :	This command loads the timer window and keeps the window open for the set time. This command can be used to suspend a sequence for a certain period of time. After timeout, the following command in the current sequence file starts.
Syntax :	TMSG [TIME (Sec)],["MESSAGE"]
Parameter(s) :	[TIME (Sec)]: Time. ["MESSAGE"]: A text message
Response :	See Table 1.1: Return values of commands.
Example :	TMSG 60,message

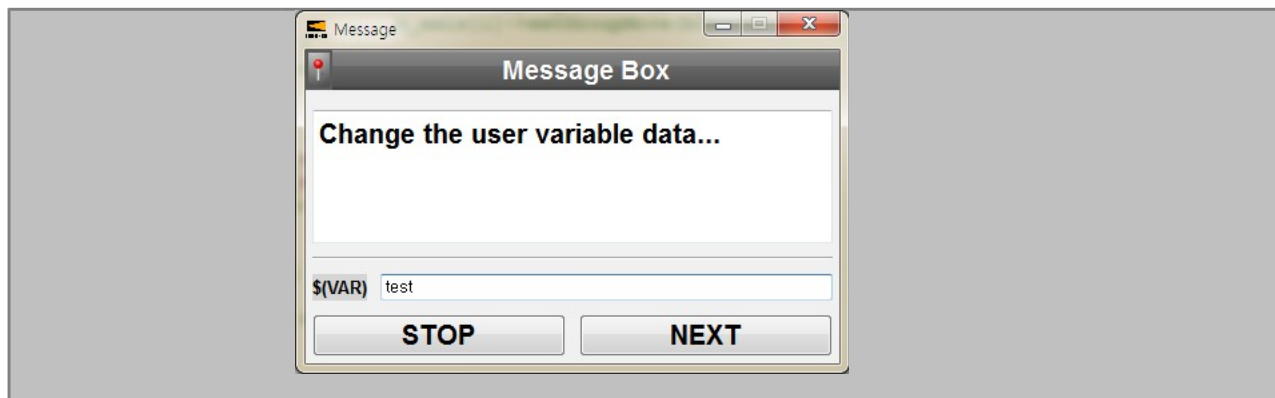


1.4.5. POWMONMSG

POWMONMSG	
Description :	It records the power/analog signal readings of the selected channels with a specified interval(INTV) for a certain period of time(TIME) into a file. It can be used for the stability test of the system. The file is saved in the folder “IFA-600-AUCME3/REPORT/POWMON”, and the format of the file name is “powmon_[Year]_[Month]_[Day]_[hh]_[mm]_[ss].txt”
Syntax :	POWMONMSG [CHANNEL_RANGE],[TIME],[“MESSAGE”]
Parameter :	[CHANNEL_RANGE]: Input channel - Powermeter channel: See Names of powermeter/currentmeter channels - Analog input channel of SC7000 (DAQ3-DAQ4): Analog input channel of SC7000(DAQ3-DAQ4) can be used only with [USE_DAQ] option parameter. [TIME]: With this option, it records time in seconds. [“MESSAGE”]: A text message
Optional Parameter(s)	[INTV]: Interval time(s) [USE_DAQ]: With this option, it allows to use the analog signal input channel of SC7000
Response :	See Table 1.1: Return values of commands.
Example :	POWMONMSG PM1-PM4,70,Monitoring... → It monitors the powermeter channel from PM1 to PM4 during 70 seconds. POWMONMSG DAQ3,60,Monitoring...,USE_DAQ → It monitors the channel #3 of the DAQ during 60 seconds.
	

1.4.6. SET_VARIABLE_MSG

SET_VARIABLE_MSG	
Description :	This command creates a pop window where a user can set a user variable value. The value of the variable is updated when “NEXT” button is clicked.
Syntax :	SET_VARIABLE_MSG [VARIABLE]
Parameter(s) :	[VARIABLE]: the name(s) of variable [MSG=“text”]: A text message on the dialog. (Default: Change the user variable data.)
Response :	See Table 1.1: Return values of commands.
Example :	SET_VARIABLE_MSG VAR SET_VARIABLE_MSG \$(VAR)



1.4.7. QMSG

QMSG	
Description :	This commands to get an input from the user through a dialog. The dialog shows a message given as a parameter and two buttons, “Yes” & “No”. If the user clicks “Yes”, then the given user variable is set to be “1”. If the user clicks “No”, then the given user variable is set to be “0”. The user variable must be defined with “CREATE_VARIABLE” before it is used. Otherwise this command terminates with “ERROR” code.
Syntax :	QMSG [USER_VAR],[“MESSAGE”]
Parameter :	[USER_VAR]: The name of the user variable to keep the user’s selection [“MESSAGE”]: A text message to be shown in the window
Optional Parameter(s)	[AUTONEXT=Time(s)]: With this option, the command ends with the user variable set with the default value given with [DEFAULT_VALUE=] option. When used without [DEFAULT_VALUE=] option, the command terminates with error in the beginning. [DEFAULT_VALUE=1/0 or YES/NO]: The default value used by [AUTONEXT] option.
Response :	See Table 1.1: Return values of commands.
Example :	QMSG RESPONSE, Check Power → A dialog with “Check Power” message appears. Then the user selection(Yes or No) is updated to a user variable “RESPONSE”(1 or 0). QMSG RESPONSE, Check Power, AUTONEXT=3, DEFAULT_VALUE=YES → The dialog windows closes in 3 seconds when no user input is given, and the user variable RESPONSE is set as “YES(1)”.

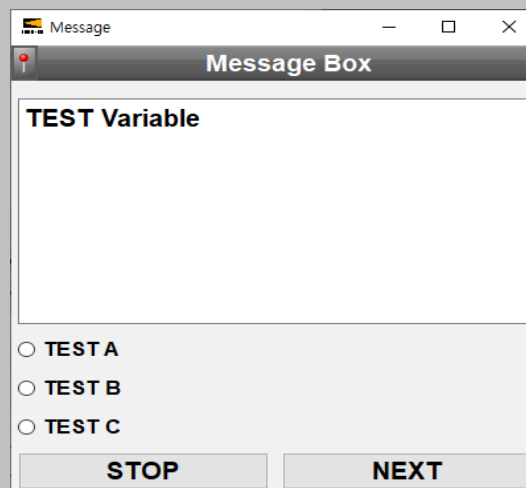
1.4.8. UPDATE_VARIABLE_UI

UPDATE_VARIABLE_UI	
Description :	This command creates two types of pop window where a user can set a user variable value. The value of the variable is updated when “NEXT” button is clicked. (Note: A user variable should be created by “CREATE_VARIABLE” command)
	- Type-A - Type-A dialog provides text filed that can put the value by the user. The input value is recorded in the user variable by clicking “NEXT” button.



2. Type-B

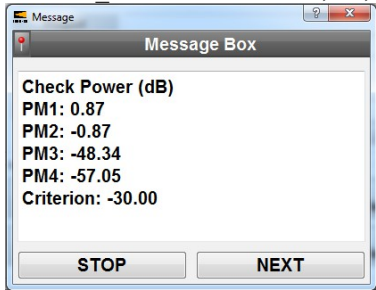
1. Type-B dialog provides multiple buttons that shows predefined values by the user. The selected value is recorded in the user variable by clicking “NEXT” button. The values are defined by the parameter [SELECTABLE_ITEM]. It returns ERROR if the user clicks “NEXT” button without choosing any buttons.

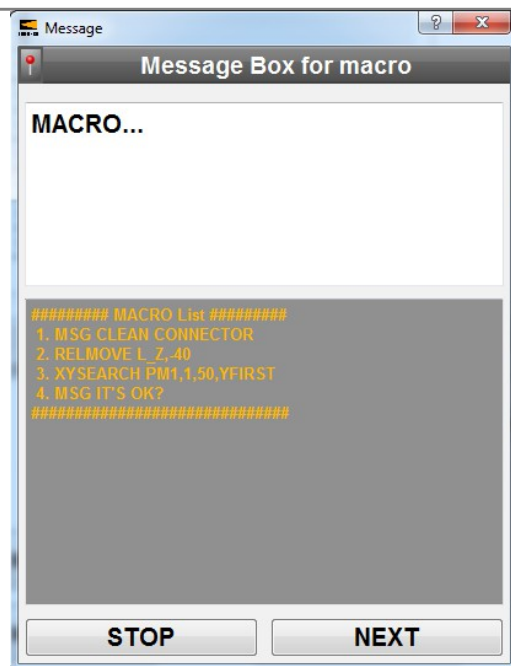


Syntax :	UPDATE_VARIABLE_UI [A], [VARIABLE], ["MESSAGE"] UPDATE_VARIABLE_UI [B], [VARIABLE], ["MESSAGE"], [SELECTABLE_ITEM]
Parameter :	[Dialog type= A/B]: Dialog type (Type-B should be used with [SELECTABLE_ITEM]). [VARIABLE]: the name(s) of variable ["MESSAGE"]: A text message to be shown in the window [SELECTABLE_ITEM]: Title of the items. (Format: Item #1 Item #2 Item #3, item separator: " "). default value of each item is item order (Item #1=1, Item #2=2, ...)
Optional Parameter(s)	[AUTONEXT=Time(s)]: With this option, the command ends with the user variable set with the default value given with [DEFAULT_VALUE=] option. When used without [DEFAULT_VALUE=] option, the command terminates with error in the beginning. [DEFAULT_VALUE=value(type-A), item title(type-B)]: The default value used by [AUTONEXT] option. [ITEM_VALUE=value]: Item values in dialog type-B. Number of values should be same number of items of the dialog.
Response :	See Table 1.1: Return values of commands.
Example :	UPDATE_VARIABLE_UI A, TEST, "Enter the test variable", AUTONEXT = 3, DEFAULT_VALUE = 10 → It shows the type-A dialog with text filed and text “Enter the test variable”. It will close the dialog and update 10 to the user variable “TEXT” in 3 seconds

UPDATE VARIABLE_UI B, TEST, "Enter the test variable", A|B|C
 → It shows the type-B dialog with 3 radio buttons(A, B, C) and the text "Enter the test variable.". If the user click "NEXT" button with choosing A, then it updates 1 to the variable "TEST".

1.4.9. CHECK_POWER

CHECK_POWER	
Description :	It creates a message box when the power reading of the selected channels are under a certain level(Threshold). Optionally, it can run a "MACRO" dialog instead of the simple message box. This option can be used to run a series of commands(MACRO) when the alignment result does not meet a certain condition.
Syntax :	CHECK_POWER [CHANNEL_RANGE],[VALUE],[MSG]
Parameter(s) :	[CHANNEL_RANGE]: startChannel – endChannel (Names of powermeter/currentmeter channels) [VALUE]: With this option, it enters the value of designated power [MSG]: The message that users want to have in window
Optional Parameter(s) :	[MACRO="Macro Name"]: The name of the "MACRO function" to be run by this command. The "Macro Name" should be one of the Macros defined in the configuration file. Read "Macro Buttons" in the chapter 6. [ELSE_MSG= "the other message"]: The message to be displayed the "else" case. If this option is not given, [MSG] is shown on the message box. [ELSE_MACRO="Macro Name"]: The name of macro to be run for the "else" case. If this option is not given, this command ends without any action.
Response :	See Table 1.1: Return values of commands.
Example :	CHECK_POWER PM1-PM4,-30, Checking power  CHECK_POWER PM1-PM4,-30, MACRO... ,MACRO=MACRO1

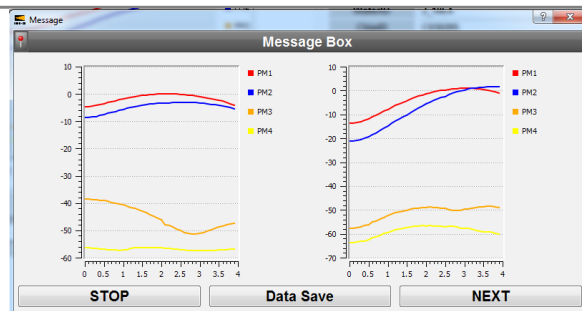


CHECK_POWER PM1-PM4, -30, low power, MACRO=MACRO1, ELSE_MSG=high power, ELSE_MACRO=MACRO2

→ It checks powers of the selected channels. If any of the values are under the “-30(dB)”, it creates a message box with “low power” text to run the macro “MACRO1”. If not, it creates a message box with “high power” text to run the “MACRO2”.

1.4.10. XYSCAN

XYSCAN	
Description :	It records the power readings of the selected channels as it moves X or Y stage for a certain range(SPAN) with a certain step size(STEP), and shows the data in forms of graph. The scan in the X direction is followed by that in the Y direction. The data can be saved into a file by using the “Data Save” button, and the format of the file name is “[WaferID] [Barcode] FAB XYSCAN.txt”.
Syntax :	XYSCAN [STAGE ASSY],[CHANNEL RANGE],[STEP],[SPAN]
Parameter(s) :	[STAGE_ASSY]: See Name of stage ass'y [CHANNEL_RANGE]: Channels(See Names of powermeter/currentmeter channels) [STEP]: Step size in μm unit [SPAN]: Scan range in μm unit
Optional Parameter(s)	[FORMAT]:1/2 1: Save data into one file 2: Save data into two separated files [NAME]: The name of data file(s) [SKIPDIALOG]: save all data acquired without showing the graph windows when the command completed
Response :	See Table 1.1: Return values of commands.
Example :	XYSCAN LEFT,PM1-PM4, 1,150 XYSCAN LEFT,PM1-PM4,1,150,FORMAT=2 → It saves data into two separated files. XYSCAN LEFT,PM1-PM4,1,150,NAME=Mydata_in_one_file.txt XYSCAN LEFT,PM1-PM4,1,150,NAME=MydataXscan.txt MydataYscan.txt



1.4.11. TWOAXIS_SCAN

TWOAXIS_SCAN	
Description :	It records the power readings of the selected channels as it moves X or Y stage for a certain range(SPAN) with a certain step size(STEP), and shows the data in forms of graph. The scan in the X direction is followed by that in the Y direction. The data can be saved into a file by using the “Data Save” button, and the format of the file name is “[WaferID] [Barcode] FAB [STAGE_AXIS1] [STAGE_AXIS2] SCAN.txt”.
Syntax :	TWOAXIS_SCAN [STAGE_AXIS1],[STAGE_AXIS2],[CHANNEL_RANGE],[STEP],[SPAN]
Parameter(s) :	[STAGE_AXIS1]: Name of stage ass'y [STAGE_AXIS2]: Name of stage ass'y [CHANNEL_RANGE]: Channels(See Names of powermeter/currentmeter channels) [STEP]: Step size in μm unit [SPAN]: Scan range in μm unit
Optional Parameter(s)	[FORMAT]:1/2 1: Save data into one file 2: Save data into two separated files [NAME]: The name of data file(s) [SKIPDIALOG]: save all data acquired without showing the graph windows when the command completed
Response :	See Table 1.1: Return values of commands.
Example :	TWOAXIS_SCAN L_Y,L_Z,PM1-PM4, 1,150 TWOAXIS_SCAN L_Y,L_Z,PM1-PM4,1,150,FORMAT=2 → It saves data into two separated files. TWOAXIS_SCAN L_Y,L_Z,PM1-PM4,1,150,NAME=Mydata_in_one_file.txt TWOAXIS_SCAN L_Y,L_Z,PM1-PM4,1,150,NAME=MydataXscan.txt MydataYscan.txt

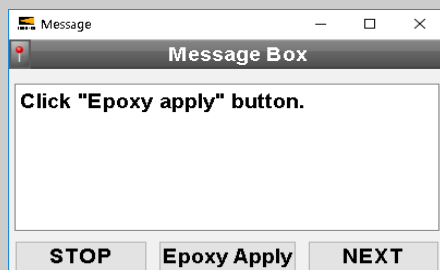
1.4.12. 3DSCAN

3DSCAN	
Description :	It displays X/Y/Z direction profile(Scanning range: [Step size]*[N Step]) on the a layer.
Syntax :	3DSCAN [Stage Assy.],[Channel],[Step Size],[N Step]
Parameter(s) :	[Stage Assy.]: Set the stage Assy. for the scanning.(See Name of stage ass'y) [Channel]: Set the scanning channel (See Names of powermeter/currentmeter channels) [Step Size]: Set the step size. [N Step]: Set the number of step.
Optional Parameter(s) :	[Name=“File name”]: With this option, it sets the data file name. (Default: ./IFA-600-AUCME3/REPORT/XYZSCAN/XYZSCAN_\$(YYYY)-\$(MM)-\$(DD)-\$(HH)\$(MI)\$

	(SS).txt) [Xstep/Ystep/Zstep]: With this option, it sets the step sizes of the each direction. [TILT_THX="Angle"]: With this option, it executes tilted scanning to ThX direction. [SP="Speed"]: Set the scanning speed (Speed= 1~5) [SLEEP=time(s)]: Waiting time in ms unit between steps (Default: 0 ms) [SELECT_STAGE = "STAGECOMBINATION"]: This option specifies the stages to be used in this command. The option value can be one of "X", "Y", "Z", "XY", "XZ", "YZ". With out this option, the command uses X, Y, and Z stages.
Response :	See Table 1.1: Return values of commands.
Example :	3DSCAN LEFT,PM1,0.02,50,SP=2 → It performs scan with L_X, L_Y, and L_Z stages with the step size of 0.02 μ m in total 50 steps(i.e. total scan in each axis is 1 μ m). 3DSCAN LEFT,PM1,0.02,50,SP=2,XSTEP=0.3,YSTEP=0.5,ZSTEP=0.4,TILT_THX=30 → It performs scans with step sizes of 0.3, 0.5, 0.4 for L_X, L_Y, L_Z stages respectively. When it scans with the L_Y stage, it also moves the L_Z stage according to the TILT_THX value(30). When it scans with the L_Z stage, it also moves the L_X stage according to the TILT_THX value(30). 3DSCAN LEFT,PM1,0.02,50,SELECT_STAGE=XZ → It performs scans for L_X, L_Y stages with 0.02 μ m step size in total 50 steps

1.4.13. EPOXYMSG

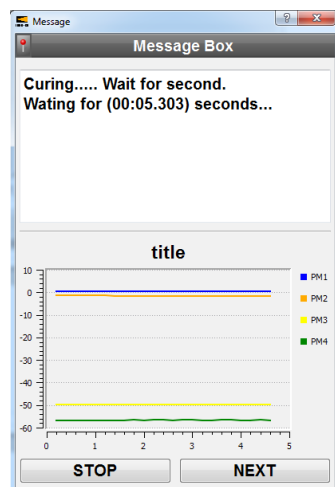
EPOXYMSG	
Description :	It displays a message box which has a button to apply epoxy.
Syntax :	EPOXYMSG ["MESSAGE"]
Parameter(s) :	["MESSAGE"]: A text message.
Response :	See Table 1.1: Return values of commands.
Example :	EPOXYMSG Click "Epoxy apply" button.



1.4.14. UVMSG

UVMSG	
Description :	Command for monitoring the measurement data during the UV curing process. It displays UV emitting time, monitored power/analog signal readings as well as the user message.
Syntax :	UVMSG [CHANNEL_RANGE],[TIME (Sec)],["MESSAGE"]
Parameter(s) :	[CHANNEL_RANGE]: startChannel – endChannel ex) PM1-CM8 (See Names of powermeter/currentmeter channels) [TIME]: Record time in seconds. ["MESSAGE"]: A text message.
Optional Parameter(s) :	[INTV=time(s)]: Interval time [INTERLOCK]: With this option, it checks the interlock signal status, and stop the countdown while it is active
Response :	See Table 1.1: Return values of commands.

Example : UVMSG PM1-PM4, 60, Curing....



1.4.15. TEMPMONMSG

TEMPMONMSG	
Description :	It records the temperature readings from the TEC controller with a specified interval(INTV) for a certain period of time(TIME) into a file. It can be used for the stability test of the system. The file is saved in the folder “IFA-600-AUCME3/REPORT/TEMPMON”, and the format of the file name is “tempmon_[Year]_[Month]_[Day]_[hh]_[mm]_[ss].txt”
Syntax :	TEMPMONMSG [TIME],["MESSAGE"]
Parameter(s) :	[TIME]: With this option, it records time in seconds. ["MESSAGE"]: A text message
Optional Parameter(s) :	[INTV]: Interval time(s) [MONINTV]: With this option, it changes the monitoring interval time of the TEC controller in the temperature monitoring. (Default: 0.5 s, unit: seconds)
Response :	See Table 1.1: Return values of commands.
Example :	TEMPMONMSG 60,monitoring... → It monitor the temperature for 60 s

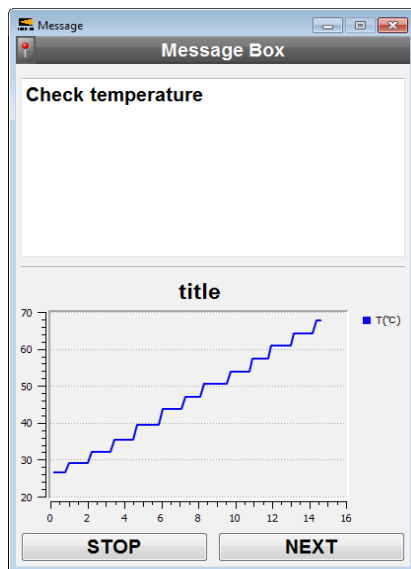
1.4.16. CONDITIONAL_WAIT_TEMP

CONDITIONAL_WAIT_TEMP	
Description :	This command displays a message window, then monitors the center jig temperature from the temperature controller, and wait until the temperature reaches the specified value or a time-out occurs. By default or with “HIGH” optional parameter, this command assumes heating and wait until the temperature become higher that the specified value. With “LOW” optional parameter, this command wait until the temperature becomes lower than the specified value.
Syntax :	CONDITIONAL_WAIT_TEMP [TEMPERATURE],[MSG]
Parameter(s) :	[TEMPERATURE]: Criteria temperature [MSG]: A message to be displayed on the dialog
Optional Parameter(s) :	[LOW/HIGH]: With this option, it checks if the temperature becomes lower/higher than the specified value. (Default: HIGH) [INTV=time(s)]: Interval time [TIMEOUT=value(s)]: With this option, it sets timeout values in second. (Default: 60 s)
Response :	See Table 1.1: Return values of commands.

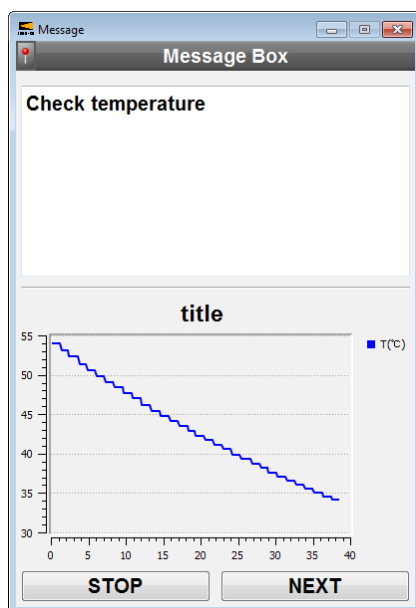
Example :

CONDITIONAL_WAIT_TEMP 70,Waiting for heating up → It displays a window with “Waiting for heating up” message, then wait until the temperature becomes higher than 70 °C or a time-out of 60 seconds occurs.

CONDITIONAL_WAIT_TEMP 25,Waiting for cooling down,TIMEOUT=120,LOW → It displays a window with “Wait for cooling down” message, then wait until the temperature becomes lower than 25 °C or a time-out of 120 seconds occurs.



CHECK_TEMP 25,Check temperature,LOW



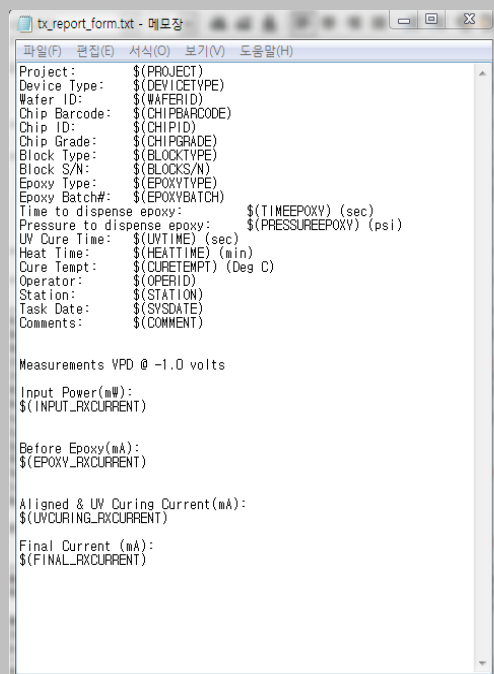
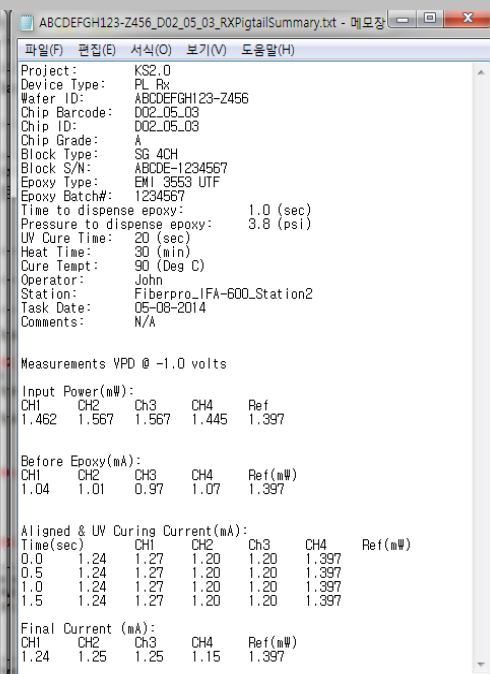
1.5. Save commands

1.5.1. SAVE_DATA

SAVE_DATA	
Description :	It saves all power readings and the positions of all stages into a file. The location of the file is under “./REPORT/SAVEDATA/”, and the default format of the file name is “[Year].[Month][Day][HH][MM][SS].txt” unless the option “NAME” is given.
Syntax :	SAVE_DATA [POW/POS/ALL]
Parameter :	POW/POS/ALL (POW: power/current readings, POS: position of all stages, ALL: POW and POS)
Optional Parameter(s)	[NAME = “File Name”]: file name
Response :	See Table 1.1: Return values of commands.
Example :	SAVE_DATA POW, NAME=SAVE_POWER_DATA.TXT

1.5.2. MAKE_CUSTOM_REPORT

MAKE_CUSTOM_REPORT	
Description :	This command is prepared to provide flexibility in the report file generation. In order to use this command, first the user needs to create a file that decides the format of the report to be generated, “form file”. The program supports various system variables and user variables that the user uses to create the form file. The variables in the form file are written in form of \$(name). For more information, check the chapter on “Variable Manager”. The example of the form file is shown in the image below. When this command is executed, it creates a report file based the specified form. The example of the output file is shown in the image below. The form file should be located in “C:\IFA-600-AUCME3\FORM” directory.

 <Report form file> Filepath - C:\WIFA600W\FORMW	 <Generated report file> Filepath - C:\WIFA600W\REPORTW\REPORTW
---	---

Syntax :	MAKE_CUSTOM_REPORT [FORM_NAME],[REPORT_NAME]
Parameter(s) :	[FORM_NAME]: The name of form file [REPORT_NAME]: The name of report file to be generated
Optional Parameter(s) :	[CH]: startChannel – endChannel, or Channel Channel (Channel: See Names of powermeter/currentmeter channels) [APPEND]: With this option, it appends new contents to the specified file. The command overwrite the file if this option is not given. [CHECKINFO]: If this option is given, it checks whether all the items set as “INFO_CHECK_CLEAR=y” in the “ProcessInfo.ini” are not blank. Also it checks whether the number of character in items set as “CHECK_LETTERCOUNT='n'” in the “ProcessInfo.ini” is same as 'n'. If the item(s) does not satisfy conditions, this command fails and display mistaken items on the sequence processing window. For more information, check the appendix section. See Table 1.1: Return values of commands.
Response :	See Table 1.1: Return values of commands.
Example :	MAKE_CUSTOM_REPORT tx_report_form.txt, \$(WAFERID) \$(CHIPID), CH=PM1-PM4 → Create a report file with the name format of ([WAFER ID]_[CHIP_ID].txt) based on the form of “ tx_report_from.txt”.

1.6. Lamp commands

1.6.1. SET_LAMP

SET_LAMP	
Description :	It controls the brightness of the illumination level for a camera. The brightness ranges 0~100. LAMP ID 1: Upper Camera 1(1.5x) 2: Upper Camera 2(8.0x) 3: Rear Camera(1.5x) 4: Upper Camera 2 LED(8.0x) Extra: Upper Camera 1 Coaxial type(1.5x) ALL: 1 2 3 4
Syntax :	SET_LAMP [LAMP_ID],[BRIGHTNESS]
Parameter(s) :	[LAMP_ID]: Lamp ID(1~4, ALL) [BRIGHTNESS]: Brightness(0~100)
Response :	See Table 1.1: Return values of commands.
Example :	SET_LAMP1,30

1.7. UV commands

1.7.1. UV_CURING

UV_CURING	
Description :	It starts/stops UV curing process. <i>* This command is only supported on the ANUJ3500 UV controller.</i>
Syntax :	UV_CURING [START/STOP]
Parameter(s) :	[START]: UV curing start [STOP]: UV curing stop
Optional Parameter(s) :	[CH=Channel(1~4) or All]:The channel number to be set. Without this option, channel 1 is set
Response :	see table 1.1.

Example :	UV_CURING START → Turn on UV source channel 1
	UV_CURING STOP → Turn on UV source channel 1
	UV_CURING START,CH=2 → Turn on UV source channel 2
	UV_CURING START,CH=3 → Turn on UV source channel 3
	UV_CURING STOP,CH=3 → Turn off UV source channel 3

1.7.2. UVSOURCE

UVSOURCE	
Description :	It moves the UV LED head to the upper/lower position.
Syntax :	UVSOURCE [UP/DOWN]
Parameter(s) :	[UP]: It moves a needle of UV heads up position [DOWN]: It moves a needle of UV heads down position
Optional Parameter(s) :	[Delay=x(s)]: It terminates the process after x seconds
Response :	See Table 1.1: Return values of commands.
Example :	UVSOURCE DOWN UVSOURCE UP,DELAY=3 → It moves uv heads up after 3 seconds.

1.7.3. SETCURINGTIME

SETCURINGTIME	
Description :	It sets the UV curing time. * This command is only supported on the ANUJ3500 UV controller.
Syntax :	SETCURINGTIME [CURING TIME]
Parameter(s) :	[CURING TIME]: Seconds [0 ~ 9990]
Optional Parameter(s) :	[CH=Channel(1~4) or All]:The channel number to be set. Without this option, channel 1 is set
Response :	see table 1.1.
Example :	SETCURINGTIME 10 → It sets the curing time of the channel 1. SETCURINGTIME 10,CH=2 → It sets the curing time of the channel 2.

1.7.4. SET_UV_INTENSITY

SET_UV_INTENSITY	
Description :	It sets the UV irradiation intensity. * This command is only supported on the ANUJ3500 UV controller.
Syntax :	SET_UV_INTENSITY [Intensity]
Parameter(s) :	[Intensity]: Irradiation intensity (0-100%)
Optional Parameter(s) :	[CH=Channel(1~4) or All]:The channel number to be set. Without this option, channel 1 is set
Response :	see table 1.1.
Example :	SET_UV_INTENSTIY 50

1.8. Dispenser command

1.8.1. EPOXY_APPLY

EPOXY_APPLY	
Description :	It controls the epoxy dispenser to dispense the epoxy once.
Syntax :	EPOXY_APPLY
Parameter(s) :	NONE
Optional Parameter(s) :	NONE

Response :	See Table 1.1: Return values of commands.
Example :	EPOXY_APPLY

1.8.2. EPOXY

EPOXY	
Description :	It controls the position of the epoxy syringe of epoxy to UP/DOWN by using a pneumatic cylinder [WARNING]: Epoxy needle must be in the upper position to avoid collision.
Syntax :	EPOXY [UP/DOWN]
Parameter(s) :	[UP]: It moves a needle of epoxy dispenser up position [DOWN]: It moves a needle of epoxy dispenser down position
Optional Parameter(s) :	[Delay=x(s)]: It terminates the process after x seconds
Response :	See Table 1.1: Return values of commands.
Example :	EPOXY UP EPOXY DOWN,DELAY=3 → It moves the epoxy syringe down after 3 seconds.

1.8.3. EPOXY_DISPENSING

EPOXY_DISPENSING	
Description:	This command is for automatic epoxy dispensing process, and performs the following steps: - A: First it displays a dialog window with “Epoxy apply”. The user may use the button to dispense epoxy to control the epoxy volume at the tip of the needle. Then the user clicks “Next” button to start the automatic process. - B: The IFA moves the epoxy needle by a certain step size(“STEP_SIZE”), then dispense epoxy for preset time, and waits for a certain time period(“SLEEP”). - C: Then IFA repeats the step B for “MAX_STEP” times.
Syntax:	EPOXY_DISPENSING [AXIS],[STEP_SIZE],[MAX_STEP],[Message]
Parameter(s):	[AXIS]: The name of the stage to move [STEP_SIZE]: Step size in um unit. [MAX_STEP]: The number of steps to move [Message]: The message to be displayed on the dialog.
Optional Parameter(s):	[SLEEP]: The waiting time in ms unit after epoxy dispensing. Default: 3000ms
Response:	See Table 1.1: Return values of commands.
Example:	EPOXY_DISPENSING L_X,1,10, Apply epoxy!!! EPOXY_DISPENSING L_X,1,10, Apply epoxy!!! ,SLEEP=5000

1.8.4. EPOXY_DISPENSING_LINE

EPOXY_DISPENSING_LINE	
Description:	This command is a command to dispense the epoxy in a straight line for a specific distance.
Syntax:	EPOXY_DISPENSING [AXIS],[RANGE]
Parameter(s):	[AXIS]: The name of the stage to move [RANGE]: Range of the dispensing epoxy
Optional Parameter(s):	[SP]: Stage speed used in the step move (Range:1~5)
Response:	See Table 1.1: Return values of commands.
Example:	EPOXY_DISPENSING_LINE L_X,10

1.8.5. SYRINGESHIFT

SYRINGESHIFT	
Description :	This command controls the position of the epoxy syringe to LEFT/RIGHT by using a pneumatic cylinder
Syntax :	SYRINGESHIFT [RIGHT(or OUT)/LEFT(or IN)]
Parameter(s) :	[LEFT or IN]: It moves to the left. the needle moves into the camera view [RIGHT or OUT]: It moves to the left. the needle moves out of the camera view
Response :	See Table 1.1: Return values of commands.
Example :	SYRINGESHIFT LEFT SYRINGESHIFT IN

1.9. TEC commands

[Warning] IFA-600 controls the temperature controller(TEC controller, Newport 350B/Arroyo 5305) through USB bus. Due to the problem in the controller, The computer does not recognize the controller if the controller is turned on before the computer gets ready.

* This commands are only supported on the TEC controller.(Newport 350B/Arroyo 5305)

1.9.1. TEC_TEMP

TEC_TEMP	
Description :	This command sets the set point of temperature controller
Syntax :	TEC_TEMP [TEMPERATURE]
Parameter(s) :	[TEMPERATURE]: set temperature in °C
Response :	see table 1.1.
Example :	TEC_TEMP 70

1.9.2. TEC_OUTPUT

TEC_OUTPUT	
Description :	This command enables/disables the control of temperature controller (TEC controller).
Syntax :	TEC_OUTPUT [ON/OFF]
Parameter(s) :	[ON/OFF]: ON/OFF
Response :	see table 1.1.
Example :	TEC_OUTPUT ON → enable temperature control.

1.9.3. TEC_THERMISTOR_CONST

TEC_CONSTANTS	
Description :	This command sets the thermistor constants of the Steinhart-Hart equation for thermistors. These values can be set from the menu item of “Menu – option – TEC Settings”.
Syntax :	TEC_THERMISTOR_CONST [C1],[C2],[C3]
Parameter(s) :	[C1,C2,C3]: TEC sensor constants.
Response :	see table 1.1.
Example :	TEC_THERMISTOR_CONST 1.176,2.265,1.126

1.9.4. TEC_PID_CONST

TEC_PID	
Description :	This command sets the proportional, integral and derivative control gain constants of the TEC control loop. These parameters can be set from the menu item of “Menu – option – TEC Settings”.

Syntax :	TEC_PID_CONST [Proportional],[Integral],[Derivative]
Parameter(s) :	[Proportional]: Proportional gain constant (1 ~ 1000) [Integral]: Integral gain constant (1 ~ 1000) [Derivative]: Derivative gain constant (1 ~ 1000)
Response :	see table 1.1.
Example :	TEC_PID_CONST 47,1,1

1.9.5. TEC_CURRENT_LIMIT

TEC_CURRENT	
Description :	This command sets the current limit of the TEC drive circuit.. This parameter can be set from the menu item of “Menu – option – TEC Settings”.
Syntax :	TEC_CURRENT_LIMIT [Current]
Parameter(s) :	[Current]: current limit in ampere unit (0~5)
Response :	see table 1.1.
Example :	TEC_CURRENT 3.5 → It sets the current limit to be 3.5A.

1.9.6. TEC_AUTOTUNE (Only for Arroyo 53xx series)

TEC_AUTOTUNE	
Description :	This command starts PID autotune process. For the detail information, please refer to Arroyo TEC controller manual.
Syntax :	TEC_AUTOTUNE
Parameter(s) :	NONE
Response :	see table 1.1.
Example :	TEC_CURRENT 3.5 → It starts PID autotune process.

1.10. Variable manager

1.10.1. CREATE_VARIABLE

CREATE_VARIABLE	
Description :	This command create a user-defined variable, and assign a value to the variable. Using this command with pre-existing variable is not allowed in normal case. To re-create the existing variable can be done either after deleting the variable or with the option “OVERWRITE”. * See the chapter on “Variable Manager”.
Syntax :	CREATE_VARIABLE [NAME]
Parameter(s) :	[NAME]: Name of a variable to be created
Optional Parameter(s) :	[DECIMAL]: This parameter determines the number of digits below the decimal point to be shown when the variable with numeric value is written/displayed in the report file. [OVERWRITE]: With this option, it deletes preexisting variable if applicable, and create new variable with the specified name. [PERMANENT]: With the option, the command saves the variable into a file permanently for later use even after restarting the software. [STRING]: With the option, the command treats numbers as string.
Response :	See Table 1.1: Return values of commands.
Example :	CREATE_VARIABLE VAR1=12.12345, DECIMAL=3 → it creates a variable with “VAR1”, and assign 12.12345 to the variable. “12.123” will be written when the variable is written in a report file. CREATE_VARIABLE VAR1=\$(PMCH1) → it creates a variable with “VAR1”, and assign the current value of the channel 1 of the power meter to the variable.

CREATE_VARIABLE VAR1=\$(PMCH1),OVERWRITE → It overwrites VAR1 to the power value of the powermeter channel #1 if VAR1 exists.
 CREATE_VARIABLE VAR3=03,STRING → It create a variable with “VAR3”, and assign “03” (not number 3) to the variable.

1.10.2. UPDATE_VARIABLE

UPDATE_VARIABLE	
Description :	Modify/Update a variable with a new value. With “ALL” parameter, this command set all the variables with a value. For details, check the chapter “Variable Manager”.
Syntax :	UPDATE_VARIABLE [NAME]
Parameter(s) :	[NAME]: The name of variable to be modified.
Optional Parameter(s)	[DECIMAL]: With this option, it sets the number of digits to be written in the report file for the numeric value.
Response :	See Table 1.1: Return values of commands.
Example :	UPDATE_VARIABLE VAR1=\$(PMCH1) → It assigns the data of channel 1 to the “VAR1”. UPDATE_VARIABLE VAR1=\$(PMCH1), DECIMAL=1 → It assigns the data of channel 1 to the “VAR1”. The value of “VAR1” will be formatted with 1 digit below the decimal point when it is printed to a file. UPDATE_VARIABLE ALL=DEFAULT → It sets all user defined variables with default values. UPDATE_VARIABLE VAR1, DECIMAL=2 → It changes the output format of “VAR1” while keeping the value.

1.10.3. DELETE_VARIABLE

DELETE_VARIABLE	
Description :	This command removes a user-defined variable corresponding to the parameter. If “ALL” is given as the parameter, this command remove all of the user defined variables. This command has no effect on the system variables. * See the chapter on “Variable Manager”
Syntax :	DELETE_VARIABLE [NAME]
Parameter :	[NAME]: The name of the variable to be removed
Response :	See Table 1.1: Return values of commands.
Example :	DELETE_VARIABLE VAR1 → It removes the user-defined variable “VAR1”. DELETE_VARIABLE ALL → It removes all of the user-defined variables.

1.10.4. CALC_VARIABLE

CALC_VARIABLE	
Description :	This command performs numerical operation with variables, and assign the result to a user-defined variable. * See the chapter on “Variable Manager”
Syntax :	CALC_VARIABLE [EXP]
Parameter :	[EXP]: Mathematical expression
Response :	See Table 1.1: Return values of commands.
Example :	CALC_VARIABLE VAR1=\$(PMCH1)+\$(PMCH2)) / 2 → It assigns the average of the power values of channel 1 and channel 2 to “VAR1”. “VAR1” should exist before used by this command.

1.10.5. LINEAR_VARIABLE

LINEAR_VARIABLE	
Description :	This command convert a dB unit variable to a linear value and load into an existing variable. For a given variable (\$VAR), the conversion is performed based on $10^{(\$VAR)*0.1}$
Syntax :	LINEAR_VARIABLE [VarRes],[VarSrc or number]
Parameter(s) :	[VarRes]: variable name to be containing the conversion result. [VarSrc]: a variable or a number to be converted
Optional Parameter(s) :	[DECIMAL=1-6]: the number of significant digits to be used in the converted number. (Default:3)
Response :	See Table 1.1: Return values of commands.
Example :	LINEAR_VARIABLE \$(DEST),\$(SRC),DECIMAL=4 → It converts the value in variable “SRC” into linear value and assign to “DEST” variable. The result will contain 4 digits below the floating point. LINEAR_VARIABLE \$(ORIG),\$(ORIG) → It converts \$(ORIG) value and assign the result to the same variable. The result will contain 4 digits below the floating point.

1.11. Miscellaneous commands

1.11.1. LEFTFIX

LEFTFIX	
Description :	Command to hold tight the left FAB jig against external force using a pneumatic device
Syntax :	LEFTFIX [0/1]
Parameter(s) :	[0/1]: 0: fix FAB jig. 1: release FAB jig.
Response :	See Table 1.1: Return values of commands.
Example :	LEFTFIX 1

1.11.2. RIGHTFIX

RIGHTFIX	
Description :	Command to hold tight the right FAB jig against external force using a pneumatic device
Syntax :	RIGHTFIX [0/1]
Parameter(s) :	[0/1]: 0: fix FAB jig. 1: release FAB jig.
Response :	See Table 1.1: Return values of commands.
Example :	RIGHTFIX 1

1.11.3. SET_VACUUM

SET_VACUUM	
Description :	Command to turn on/off one of two vacuum channels installed on the optical table.
Syntax :	SET_VACUUM [INDEX],[ON/OFF]
Parameter :	[INDEX]: VACUUM INDEX (1~2) [ON/OFF]: ON/OFF
Response :	See Table 1.1: Return values of commands.
Example :	SET_VACUUM 1,ON

1.11.4. REARCAMERA

REARCAMERA	
Description :	It moves the rear camera to the front/back position.
Syntax :	REARCAMERA [FRONT/BACK]
Parameter(s) :	[FRONT/BACK]
Optional Parameter(s):	[DELAY=x(s)]: With this option, it finishes the command after the x seconds
Response :	See Table 1.1: Return values of commands.
Example :	REARCAMERA FRONT → It moves the rear camera to the front position by pneumatic action.

1.11.5. SET_SOLENOID

SET SOLENOID	
Description :	This command controls a pneumatic device at a certain port.
Syntax :	SET SOLENOID [PORT],[ON/OFF]
Parameter(s) :	[PORT]: Port number (1~5) [ON/OFF]: Open/Close a choosen solenoid valves
Response :	See Table 1.1: Return values of commands.
Example :	SET_SOLENOID 1,ON

1.11.6. EXECMACRO

EXECMACRO	
Description :	Command to use a MACRO.
Syntax :	EXECMARCO [MACRO_NAME or MACRO_INDEX]
Parameter(s) :	[MACRO_NAME or MACRO_INDEX]: The name/index of macro defined in a macro file.
Optional Parameter(s) :	[MACRO_INDEX]: This option should be given when a macro is called with an index number.
Response :	See Table 1.1: Return values of commands.
Example :	EXECMACRO MACRO_TEST → It executes the macro with the name of "MACRO_TEST". EXECMACRO 1,MACROIDX → execute the macro index number of 1

1.11.7. LOAD_MACRO

LOAD_MACRO	
Description :	This command load a specified "macro" file to the program. The default macro file name is ./System/Device/seqmacro.ini.
Syntax :	LOAD_MACRO
Parameter(s) :	NONE
Optional Parameter(s) :	[NAME]: macro file name.
Response :	See Table 1.1: Return values of commands.
Example :	LOAD_MACRO NAME=seqmacro2 LOAD_MACRO NAME=./System/Device/seqmacro2

1.11.8. PAUSE

PAUSE	
Description :	Command to pause the ongoing operation for a certain period of time.
Syntax :	PAUSE [TIME(Sec)]

Parameter(s) :	[TIME]: Pause time in second
Response :	See Table 1.1: Return values of commands.
Example :	PAUSE 10

1.11.9. STEP

STEP	
Description :	Command to determine step size of the stage control window.
Syntax :	STEP [STAGE_ASSY],[X_STEP],[Y_STEP],[Z_STEP],[ThX_STEP],[ThY_STEP],[ThZ_STEP]
Parameter(s) :	[STAGE_ASSY]: See Name of stage ass'y [X_STEP] .. [ThZ_STEP]: Step sizes in μm unit or degree for the corresponding axes
Optional Parameter(s)	[STEPMODE]: this option changes the stage control mode to “STEP” when the previous mode is “CONTINUOUS”
Response :	See Table 1.1: Return values of commands.
Example :	STEP LEFT,10,10,10,0.1,0.1,0.1

1.11.10. SET_COORDINATE_MODE

SET_COORDINATE_MODE	
Description :	This command sets coordinate mode to be displayed on the control panel.
Syntax :	SET_COORDINATE_MODE [MODE]
Parameter(s) :	[MODE]: - REL: Set the mode to “relative coordinate mode”. all the coordinate values becomes zeros when this command is issued. - STD(ABS): Change the mode to the absolute coordinate mode
Response :	See Table 1.1: Return values of commands.
Example :	SET_COORDINATE_MODE REL SET_COORDINATE_MODE ABS SET_COORDINATE_MODE STD

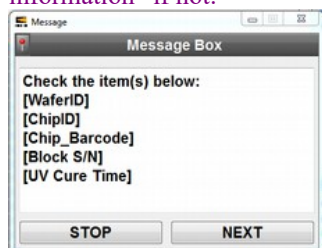
1.11.11. SET_GUI_LOCK

SET_GUI_LOCK	
Description :	This command locks/unlocks some of control panels related with stage movement. This function is useful to avoid the possible damage caused by manipulating wrong control panels. The initial lock/unlock states of the control panels are set by the system file “ViewInfo.ini”(for more information, check the appendix)
Syntax :	SET_GUI_LOCK [LOCK/UNLOCK],[GUIGROUP]
Parameter(s) :	[LOCK/UNLOCK] - LOCK: It locks the specified panel. - UNLOCK: It unlocks the specified panel. [GUIGROUP] - JOG_LEFT: Control panel for the left stage assembly - JOG_S1: Control panel for the s1 stage assembly - JOG_RIGHT: Control panel for the right stage assembly - JOG_UPPER: Control panel for the upper stage assy(tool positioner) - JOG_CENTER: Control panel for the center stage assy - STAGE_ALL: Control panel for all stage assemblies - SEARCH_LEFT: Control panel for the L Search - SEARCH_S1: Control panel for the S1 Search - SEARCH_RIGHT: Control panel for the R Search - POSITION_TAB: The tab to allows for a user to set/update/save/load sets of stage

	positions for predefined actions. The information of the position sets is included a sequence file.
	- CONTROL_TAB: The tab which has the predefined tool buttons - MACRO_TAB: The tab which has the macro buttons - EDIT_SQF: The button to open the sequence file currently loaded
Response :	See Table 1.1: Return values of commands.
Example :	SET_GUI_LOCK LOCK,SEARCH_LEFT → It locks the control panel for Search left related action.

1.11.12. CHECK_PROCESSINFO

CHECK_PROCESSINFO	
Description :	This command checks whether all the items set as “INFO_CHECK_CLEAR=y” in the “ProcessInfo.ini” are not blank. Also it checks whether the number of character in items set as “CHECK_LETTERCOUNT='n'” in the “ProcessInfo.ini” is the same as 'n'. If not, it creates a message window to inform the user. For more information, check the appendix part.
Syntax :	CHECK_PROCESSINFO
Parameter(s) :	NONE
Optional Parameter(s) :	[MSG=“Message”]: The message that users want to have in window
Response :	See Table 1.1: Return values of commands.
Example :	CHECK_PROCESSINFO MSG=Check the Process information → It checks if all the necessary items are not blank, and creates a window with the message “Check the Process information” if not.



1.11.13. CLEAR_PROCESSINFO

CLEAR_PROCESSINFO	
Description :	Clear the selected items of the process Information window, and set those value with the default values. The items to be affected by this command are marked as “INFO_CHECK_CLEAR=y” in the “ProcessInfo.ini”. For more information, check the appendix part.
Syntax :	CLEAR_PROCESSINFO
Parameter :	NONE
Optional Parameter(s) :	[ALL]: With this option, it clears all items. the settings in “ProcessInfo.ini” are ignored
Response :	See Table 1.1: Return values of commands.
Example :	CLEAR_PROCESSINFO CLEAR_PROCESSINFO ALL

1.11.14. MDU_CURRENT

MDU_CURRENT	
Description :	Command to set a current level of the current source in MDU1000

Syntax :	SET_PROBE [CHANNEL],[CURRENT]
Parameter(s) :	[CHANNEL]: Source channel (1~4) [CURRENT]: Current level (0~300 mA) (*Note: The maximum current output is 200mA/ch when all the channels are driven. The maximum current for single channel driving is 300 mA.)
Response :	See Table 1.1: Return values of commands.
Example :	MDU_CURRENT 1,30 MDU_CURRENT 1,300 → It sets the current level of channel #1 to “300 mA”. The user should turn off other channels.

1.11.15. SET_OPTICALSOURCE

SET_OPTICALSOURCE	
Description :	Command to turn on/off the light source of MDU1000
Syntax :	SET_OPTICALSOURCE [ON/OFF]
Parameter(s) :	[ON/OFF]: ON/OFF
Response :	See Table 1.1: Return values of commands.
Example :	SET_OPTICALSOURCE ON

1.11.16. CTS_ZEROING

CTS_ZEROING	
Description :	Command for zeroing the measurement modules of PM2100. When the electrical offset of the current meter/ power meter modules of PM2100 is relatively high, the measurement error for low current/power range is inaccurate. Especially the current meter module(CM2104) has high level of offset due to the long length of electrical cable connected to the probe. This command is useful to cancel electrical offset of the current meter/ power meter modules for more accurate measurement of small current/optical power. The zeroing takes normally 1 second, and the result is temporally saved to the PM2100 and used to calculate current/power. The zeroing becomes invalid once the PM2100 is turned off. [Warning] When this command is used for the CM2104 current meter module, the electrical probe must be electrically disconnected to the chip. For CM2104 power meter module, the optical ports of the module must be blocked by caps to make sure any stray light does not go into the detectors. The power meter module(CM2104) does NOT require zeroing in normal cases.
Syntax :	CTS_ZEROING [Module#]
Parameter(s) :	[Module#]: 0 .. N (N+1: is the number of modules installed). If not given, the zeroing will be done with module #0.
Response :	See Table 1.1: Return values of commands.
Example :	CTS_ZEROING → perform zeroing for module# 0 (current meter) CTS_ZEROING 0 → perform zeroing for module# 0 (current meter) CTS_ZEROING 1 → perform zeroing for module# 2 (2nd power meter module.)

1.11.17. CTS_UNIT

CTS_UNIT	
Description :	Command to change the units of power/current readings in the data monitoring window
Syntax :	CTS_UNIT [dB/M/U]
Parameter(s) :	[dB]: dB(m) [M]: mW or mA [U]: uW or uA

Response :	See Table 1.1: Return values of commands.
Example :	CTS_UNIT DB

1.11.18. SELECT_SPEEDTABLE

SELECT_SPEEDTABLE	
Description :	Command to load a new speed table for the stages. Two speed tables, TYPE1 and TYPE2, are prepared for a system in forms of file, and the software loads the TYPE1 table by default when it starts. This command allows to load one of two speed tables into current settings. Each speed table provides eight speed settings for each stage. Five of those setting are for alignment commands, two of those are for user control, and the last one is for homing process. The names of the files with the speed tables are: - TYPE1: ./System/MC/mc_speed_table_type1.ini - TYPE2: ./System/MC/mc_speed_table_type2.ini
Syntax :	SELECT_SPEEDTABLE [TYPE]
Parameter(s) :	[TYPE]: TYPE1/TYPE2
Response :	See Table 1.1: Return values of commands.
Example :	SELECT_SPEEDTABLE TYPE1

1.11.19. MONITOR_POWER

MONITOR_POWER	
Description :	Command to turn on/off the periodic power reading on the data monitoring window.
Syntax :	MONITOR_POWER [ON/OFF]
Parameter(s) :	[ON/OFF]: ON/OFF
Response :	See Table 1.1: Return values of commands.
Example :	MONITOR_POWER ON

1.11.20. GET_GUIDE_POS

GET_GUIDE_POS	
Description :	It displays the position of the pneumatic positioner(Epoxy syringe holder, UV head holder and the rear camera) on the sequence process window. (for maintenance)
Syntax :	GET_GUIDE_POS [UV/EPOXY/REARCAM/ALL]
Parameter(s) :	[UV]: Check UV head holder position [EPOXY]: Check epoxy syringe holder position [REARCAM]: Check the rear camera position. [ALL]: With this option, the positions of the UV head holder, epoxy syringe holder and rear camera are checked.
Response :	See Table 1.1: Return values of commands.
Example :	GET_GUIDE_POS ALL

1.11.21. TERMINAL_SHOW

TERMINAL_SHOW	
Description :	Command to show a terminal window to communicate with SC7000 or MDU1000. (for maintenance only)
Syntax :	TERMINAL_SHOW
Parameter(s) :	NONE
Response :	See Table 1.1: Return values of commands.

Example :	TERMINAL SHOW
-----------	---------------

1.11.22. SET_LOOPCNT

SET_LOOPCNT	
Description :	The sequence manager provides the function to repeat the command(s) in the “EXECUTION” section of a sequence file by certain times. This command sets the number of the repetition.
Syntax :	SET_LOOPCNT [LOOP_NUM]
Parameter(s) :	[LOOP_NUM]: No. of repetition
Response :	See Table 1.1: Return values of commands.
Example :	SET_LOOPCNT 5

1.11.23. WAIT_FOR_STOP

WAIT_FOR_STOP	
Description :	It waits until all of motorized stage stops, then it terminates the process.
Syntax :	WAIT_FOR_STOP
Parameter :	None
Response :	See Table 1.1: Return values of commands.
Example :	WAIT_FOR_STOP

1.11.24. SEQUENCE_STOP

SEQUENCE_STOP	
Description :	It stops sequence running and returns “RET_STOP”.
Syntax :	SEQUENCE_STOP
Response :	See Table 1.1: Return values of commands.
Example :	SEQUENCE_STOP → It stops sequence running.

1.11.25. UPDATE_POSITION

UPDATE_POSITION	
Description :	It updates the current positions of certain stages to a predefined position in “Position window” tab.
Syntax :	UPDATE_POSITION [POSITION_TAB_TITLE]
Response :	See Table 1.1: Return values of commands.
Example :	UPDATE_POSITION UOPTIC_LOADING → It updates the current position to “UOPTIC_LOADING”.

1.11.26. SAVE_POSITION

SAVE_POSITION	
Description :	It updates the current positions of certain stages to a predefined position in “Position window” tab.
Syntax :	SAVE_POSITION [POSITION_TAB_TITLE]
Response :	See Table 1.1: Return values of commands.
Example :	SAVE_POSITION TX UOPTIC_LOADING → It saves “UOPTIC_LOADING” to the sequence file currently loaded on the program. SAVE_POSITION ALL → It saves all the predefined positions to the sequence file currently loaded on the program.

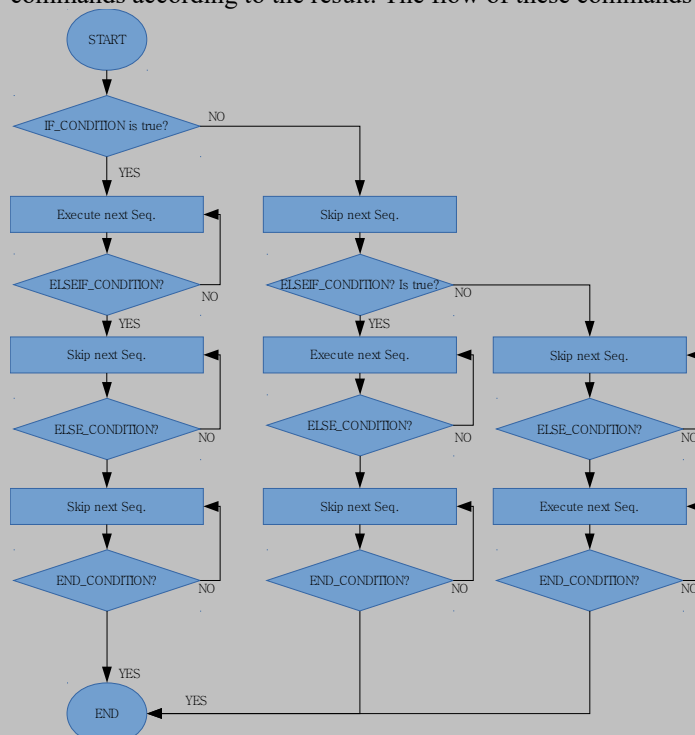
1.11.27. IF_CONDITION / ELSE_CONDITION / ELSEIF_CONDITION / END_CONDITION

IF_CONDITION / ELSE_CONDITION / ELSEIF_CONDITION / END_CONDITION

Description : These commands support conditional execution of a series of commands. These commands are valid only they are executed by “RUN” or “RESUME”.

“IF_CONDITION” command supports the operators of ‘<=’, ‘<’, ‘>’, ‘>=’, ‘==’, ‘<>’, ‘&&’, ‘||’ and the numeric operations of ‘+’, ‘-’, ‘*’, ‘/’, as well as system variables and user variables.

As an example IF_CONDITION \$(PMCH1) * \$(PMCH2) / 2 >= -15 command checks if the sum of the channel 1 and the channel 2 is larger than or equal to -15, and execute the commands according to the result. The flow of these commands is shown below.



Syntax : IF_CONDITION [CONDITION]
ELSE_CONDITION
ELSEIF_CONDITION
END_CONDITION

Parameter : [CONDITION]: Conditional statement

Response : See Table 1.1: Return values of commands.

Example : IF_CONDITION \$(PMCH1) > -5
MSG A
ELSE_CONDITION
MSG B
END_CONDITION
→ If the power of the channel 1(\$PMCH1)) is larger than -5, execute “MSG A”.
Otherwise execute “MSG B”

IF_CONDITION \$(PMCH2) <= -50
MSG A
END_CONDITION
→ If the power of the channel 2 is less than or equal to -50, execute “MSG A”

```

IF_CONDITION $(PMCH2) > -50 && $(PMCH2) < -40
MSG A
ELSEIF_CONDITION $(PMCH2) > -30 && $(PMCH2) < -20
MSG B
ELSE_CONDITION
MSG C
END_CONDITION

```

→ If the power of the channel 2(PMCH2) is larger than -50 and smaller than -40, then it will execute “MSG A”. If the power is larger than -30 and smaller than -20, then it will execute “MSG B”. Otherwise it will execute “MSG C”

1.11.28. CHECK_CONDITION

CHECK_CONDITION

Description : These commands checks the condition in the syntax,

- If the condition is TRUE, then it returns “RET_SUCCEEDED”.
- If the condition is FALSE, then it shows the text message window. If the user click “NEXT” button, then it checks the condition again.

With the option “USE_SKIP”, then message box shows the “SKIP” button to return “RET_WARNING” and runs next command during sequence running.

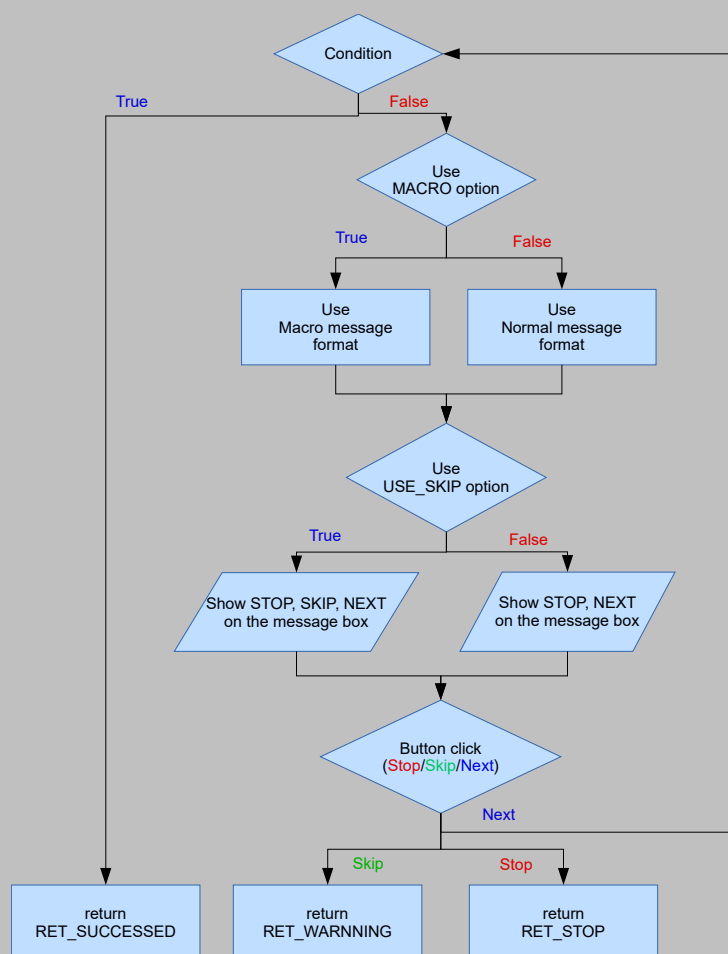


Fig. 1.4: Flow chart of CHECK_CONDITION command

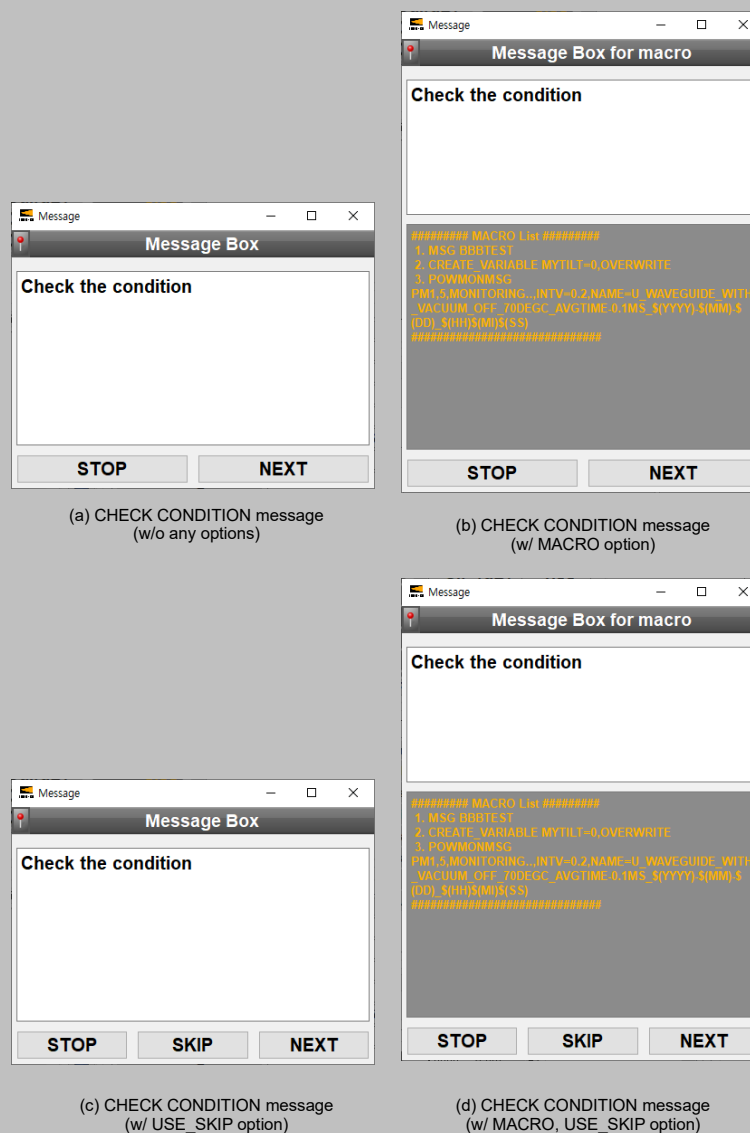
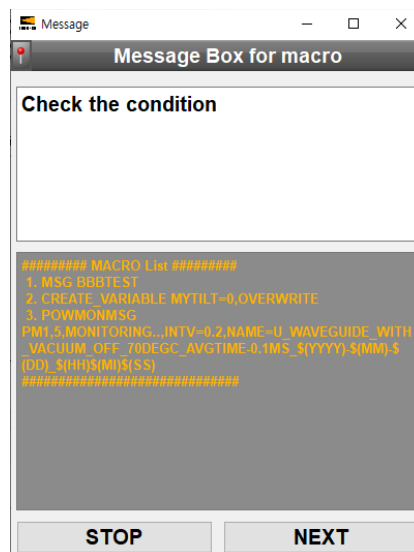


Fig. 1.5: Types of CHECK_CONDITION messages
(Condition=FALSE)

Syntax :	CHECK_CONDITION [CONDITION], [MESSAGE]
Parameter :	[CONDITION]: Condition the operators of '<=', '<', '>', '>=', '==', '<>', '&&', ' ' and the numeric operations of '+', '-', '*', '/', as well as system variables and user variables. [MESSAGE]: Text message
Optional Parameter(s)	[MACRO="macro name"]: The macro name executed if the condition is false [USE_SKIP]: With this option, program displays the "SKIP" button on the message window. [RET_VAR="variable name"]: With this option, it records the return code of the command to the variable.
Response :	See Table 1.1: Return values of commands.
Example :	CHECK_CONDITION \$(PMCH1)>-50, Check power, MACRO=XYSEARCH

→ It checks the powermeter channel #1 is higher than -50 dBm. If it is not true, then it shows the macro message that can run XYSEARCH macro with clicking “NEXT” button.



1.11.29. MDU CONTROL

MDU_CONTROL	
Description :	Communicate with MDU1000(debug purpose) The response from MDU1000 is displayed on the sequence processing window.
Syntax :	MDU_CONTROL [COMMAND]
Parameter(s) :	[COMMAND]: Command
Response :	See Table 1.1: Return values of commands.
Example :	MDU_CONTROL MDU1000 idn? 1\n

1.11.30. SET PM WAVE

SET_PM_WAVE	
Description :	It sets the wavelength used for optical power calculation..
Syntax :	SET_PM_WAVE [WAVE]
Parameter(s) :	[WAVE]: Wavelength
Response :	See Table 1.1: Return values of commands.
Example :	SET PM_WAVE 1280→ It sets the wavelength of the powermeter to 1280nm.

1.11.31. SET PM RANGE UPPER

SET PM RANGE UPPER	
Description :	It sets the measurement range of the powermeter. This command is supported only for the PM5020 powermeter. [NOTE] • The specified value is used to select an appropriate measurement range. • The actual range applied by the power meter may differ from the specified value. • Executing this command sets the power meter to Manual Range mode. • Auto Range is automatically disabled when this command is executed.

Syntax :	SET_PM_RANGE_UPPER [POWER]
Parameter :	[POWER]: Specifies the power value used to select the measurement range.
Optional Parameter(s) :	[UNIT=DBM or W] - DBM: Interprets the [POWER] value in dBm. The value is internally converted to watts before setting the measurement range. - W: Interprets the [POWER] value in watts. This is the default unit when the UNIT option is omitted. [WRITE]: Saves the specified range value as the default alignment range. The saved value is used when a Range option is not specified in the alignment command. If the WRITE option is not specified, the range is applied only to the current power meter setting and is not saved.
Response :	See Table 1.1: Return values of commands.
Example :	SET_PM_RANGE_UPPER 0.001 SET_PM_RANGE_UPPER -3, UNIT=DBM

1.11.32. GET_PM_RANGE_UPPER

GET PM RANGE UPPER	
Description :	It returns the current measurement range upper value of the power meter. This command is supported only for the PM5020 power meter.
	[NOTE] - The specified value is used to select an appropriate measurement range. - The actual range applied by the power meter may differ from the specified value. - Executing this command sets the power meter to Manual Range mode. - Auto Range is automatically disabled when this command is executed.
Syntax :	GET_PM_RANGE_UPPER
Parameter :	None
Optional Parameter(s) :	[WIDER=(n)]: Specifies how many range steps above the current measurement range should be returned. n must be 0 or greater. If the requested number of steps exceeds the available range, the maximum measurement range is returned. [Examples]: - WIDER=0: Returns the current measurement range. - WIDER=1: Returns one range step above the current measurement range. - WIDER=2: Returns two range steps above the current measurement range.
Response :	See Table 1.1: Return values of commands.
Example :	SET_PM_RANGE_UPPER 0.001 SET_PM_RANGE_UPPER -3, UNIT=DBM

1.11.33. SET_PM_AUTO_RANGE

SET PM AUTO RANGE	
Description :	It enables or disables the Auto Range function of the power meter. This command is supported only for the PM5020 power meter.

[NOTE]

- When Auto Range is enabled, the power meter monitors the measured signal and automatically selects an appropriate measurement range.
- Changing the measurement range may interrupt measurement for up to 10 ms.
- Measurement resumes automatically after the range change is completed.
- Setting a manual measurement range automatically disables Auto Range.

Syntax :	SET_PM_AUTO_RANGE [STATE]
Parameter(s) :	[STATE]: Specifies the auto range state. ☐ ON: Enable ☐ OFF: Disable
Response :	See Table 1.1: Return values of commands.
Example :	SET_PM_AUTO_RANGE ON

1.11.34. PM_COMM

PM_COMM	
Description :	This command sends a raw command to the Power-meter connected to the IFA-600 software. If the command contains a question mark (“?”), it operates in Query mode, and the response is displayed in the Sequence Monitoring window. When the [VAR] option is used in Query mode, the response value is stored in the specified variable. *The format and meaning of the raw command entered in the [Command] parameter should be referenced from the Power-meter manual.
Syntax :	PM_COMM [COMMAND]
Parameter :	[COMMAND]: Raw command to be sent to the Power-meter device (Query or Write).
Optional Parameter(s) :	[QUERY]: With this option, it operates in Query mode. [VAR=name of variable]: Specifies the user variable in which the response value will be stored when the command is a Query. [SLEEP=]: Wait time for the response when the command is a Query (in milliseconds). The default value is 50 ms.
Response :	see table 1.
Example :	PM_COMM idn?

1.11.35. SET_PM_AOUT

SET_PM_AOUT	
Description :	It outputs the power signal of the specified channel [CHANNEL] in the specified module [MODULE] to the analog output channel [AOUT_CH] of the powermeter(PM2100).
Syntax :	SET_PM_AOUT [AOUT_CH],[MODULE],[CHANNEL]
Parameter(s) :	[AOUT_CH]: Analog channel(1~2) [MODULE]: Module index(0~4) [CHANNEL]: Channel index(1~4)
Response :	See Table 1.1: Return values of commands.
Example :	SET_PM_AOUT 1,0,1 → It outputs the signal of channel 1 in the module 0 to the analog channel 1.

1.11.36. SET_PM_GAIN

SET_PM_GAIN	
Description :	It sets the gain of the powermeter(PM2100).
Syntax :	SET_PM_GAIN [GAIN]
Parameter(s) :	[GAIN]: 1~5
Response :	See Table 1.1: Return values of commands.
Example :	SET_PM_GAIN 1 → It sets the gain of the powermeter to 1.

1.11.37. SET_FAN

SET_FAN	
Description :	It controls On/Off the fan devices installed on the center jig.
Syntax :	SET_FAN [ON/OFF]
Parameter(s) :	[ON/OFF]: ON/OFF
Response :	See Table 1.1: Return values of commands.
Example :	SET_FAN ON

1.11.38. HANDLE_COLLISION

HANDLE_COLLISION	
Description :	When FAB jig is pushed backward beyond a certain limit, the displacement sensor output triggers the safety logic of the motion controller(SC7000) and freezes the controller. This command releases the freeze state and retreats the axis.
Syntax :	HANDLE_COLLISION [AXIS],[BACK_DISTANCE]
Parameter(s) :	[Axis]: Stage axis [Distance]: retreat distance in μm unit
Response :	See Table 1.1: Return values of commands.
Example :	HANDLE_COLLISION L_Z, 500

1.11.39. STAGE_CONTROL_MODE

STAGE_CONTROL_MODE	
Description :	This command sets the control mode of the stage control buttons on GUI panel. To avoid possible collision, currently it can set the mode only to “STEP” mode. Change to “CONTINUOUS” control mode is not supported.
Syntax :	STAGE_CONTROL_MODE [MODE]
Parameter :	[MODE]: STEP or CONTINUOUS(Not supported)
Optional Parameter(s)	[GROUP]: STAGE_ASSY (LEFT/RIGHT)
Response :	See Table 1.1: Return values of commands.
Example :	STAGE_CONTROL_MODE STEP, STAGE_CONTROL_MODE STEP,GROUP= LEFT STAGE_CONTROL_MODE STEP,GROUP=LEFT RIGHT

1.11.40. SELECT_TAB

SELECT_TAB	
Description :	This commands activates the specified tab window.
Syntax :	SELECT_TAB [Type]
Parameter(s) :	[Type]: PM/CM (PM: optical power reading, CM: current reading)

	• OPERATION_TAB/POSITION_TAB/AXIS_CONTROL_TAB/MACRO_BUTTON_TAB/CONTROL_PANEL_TAB/SEQUENCE_TAB/JOG_CONTROL_TAB • OPERATION_TAB: “Operation” tab window in the main window • POSITION_TAB: Position” tab in the main window • AXIS_CONTROL_TAB: “Axis Control” tab in the bottom dock window • MACRO_BUTTON_TAB: “Macro button” tab in the bottom dock window • CONTROL_PANEL_TAB: “Control Panel” tab in the bottom dock window • SEQUENCE_TAB: Sequence” tab in the right dock window • JOG_CONTROL_TAB: “JOG Control” tab in the right dock window
Optional Parameter(s)	[SUBTAB]: With this option, it shows sub-level of the specified tab. • OPERATION_TAB: GRAPH_A_TAB/GRAPH_B_TAB/GRAPH_C_TAB/GRAPH_D_TAB/GRAPH_3D_TAB • POSITION_TAB: START_POSITION_TAB/TOOL_POSITION_TAB
Response :	See Table 1.1: Return values of commands.
Example :	SELECT_TAB CM

1.11.41. SET_SW_LIMIT

SET_SW_LIMIT	
Description :	This command set/clear software limits in coordinates of the specified stage. When the software limit is set, the corresponding stage can only move between the lower limit and the higher limit. Basic software limits are set in the environmental file “mc_stageinfo_table.txt”, and applied to the motion controller when the program starts. Once the software limits are set, those values are valid until the motion controller(SC7000) is turned off. This command has two parameters of “set / clear” to modify the software limits. If “set” parameter is chosen, this command overrides the software limits set at the start of the program, and the new values are valid until the program restarts or this command is executed again. If “clear” parameter is chosen, the command restore the software limits to the values set in the environmental file. For details on the file “mc_stageinfo_table.txt”, check the appendix part “Stage information file”.
Syntax :	SET_SW_LIMIT [SET/CLEAR],[STAGE],[LOW_LIMIT],[HIGH_LIMIT]
Parameter :	[SET/CLEAR]: set or clear software limit [STAGE]: stage name. (“ALL” can be used with “CLEAR”) [LOW_LIMIT]: lower software limit in um or degree unit [HIGH_LIMIT]: higher software limit in um or degree unit
Response :	See Table 1.1: Return values of commands.
Example :	SET_SW_LIMIT SET, R_Y, -5000, 19000 → It sets the software limits of R_Y stage to be -5000um, 19000um. SET_SW_LIMIT CLEAR, R_Y → It restores the software limits of R_Y from the environmental file “mc_stageinfo_table.txt” SET_SW_LIMIT CLEAR, ALL → It restores the software limits of all of the stages from the environmental file “mc_stageinfo_table.txt”.

1.11.42. USB_COMM

USB_COMM	
Description :	This command communicates with a specified usb device that uses user-mode-usb driver. In order to use this command, the vendor id and the product id of the usb device are used to specify the device. The response for the written command is displayed on the sequence monitoring window

	and saved as a specified variable.
Syntax :	USB_COMM [VID], [PID], [Command]
Parameter :	[VID]: vendor id of the usb device [PID]: product id of the usb device [Command]: command to be sent to the device
Option parameter(s) :	[VAR=\$(var1)]: The name of the variable that is to contain the response of the usb device.
Response :	See Table 1.1: Return values of commands.
Example :	USB_COMM 2125, 7001, mlx temp, VAR=\$(tempearture_var)

1.11.43. GPIB_COMM

GPIB_COMM	
Description :	It perform the communication for the instrument connected the GPIB interface. If the command included question mark("?"), then it communicates query method. And the response value is displayed on the sequence monitoring window. With [VAR] option, the value is recorded on the variable.
Syntax :	GPIB_COMM [BoardNo.],[PAD],[Command]
Parameter(s) :	[BoardNo.]: Index of the access board for the device [PAD]: The primary GPIB address of the device [Command]: GPIB command
Optional Parameter(s) :	[VAR=\$(name of variable)]: With this option, The response by the query is recorded in the variable.
Response :	See Table 1.1: Return values of commands.
Example :	GPIB_COMM 0,16, AVG? 0 1, VAR=\$(average) → It sends the command “AVG? 0 1” to the device which has 16 as the address # and is connected “board 0”. And record the response in the user variable “average”. GPIB_COMM 0,16, 1, AVG 0.1 → It sends the command “AVG 0 1” to the device which has 16 as the address # and is connected “board 0”.

1.11.44. CREATE_RANDOM

CREATE_RANDOM	
Description :	This command creates a random value within the specified range, and assigns the value to the specified user variable. The user variable must exist beforehand.
Syntax :	CREATE_RANDOM [VAR], [Low limit], [High limit]
Parameter(s) :	[VAR]: The name(s) of variables [Low limit]: lower limit of the range [High limit]: higher limit of the range
Optional Parameter(s):	[DECIMAL=n]: the number of digits below the floating point (default: 0) [CREATE]: With this option, the command creates a variable with the specified name, and assign the newly created random number.
Response :	See Table 1.1: Return values of commands.
Example :	CREATE_RANDOM RANDNUM, -10, 10 → create a random number between -10 and 10, and assign the number to the variable \$ (RANDNUM).

1.11.45. SAVE_LAYOUT

USB_COMM	
Description :	This command saves the current layout information on main GUI to the configuration file.
Syntax :	SAVE_LAYOUT [FileName]

Parameter :	[FileName]: File name. (File path: .\Layout\[FileName])
Response :	See Table 1.1: Return values of commands.
Example :	SAVE_LAYOUT LAYOUTFILE → It saves current layout information on .\Layout\LAYOUTFILE.

1.11.46. LOAD_LAYOUT

USB COMM	
Description :	It applies the layout information recorded in the specific layout file in “.Layout” folder. If the user doesn’t put the [FileName] option, then it applies the layout information on the “DefaultLayout” file.
Syntax :	LOAD_LAYOUT
Parameter :	None
Optional Parameter(s) :	[FileName]: File name.
Response :	See Table 1.1: Return values of commands.
Example :	LOAD_LAYOUT → “./Layout/Default.Layout” LOAD_LAYOUT LAYOUTFILE → “./Layout/LAYOUTFILE.Layout”

1.12. System test commands

1.12.1. TEST_XYSEARCH

TEST_XYSEARCH	
Description :	Command to check the repeatability of X/Y alignment. It performs “XYSEARCH [Stage],[CH],[Step],[MaxStep]” command, and records the results on the report file. [NOTE]: Test results are saved as the below: - Report file path: C:\IFA-600-AUCME3\TESTCMD\XYSEARCH\ - Report file name: TEST_XYSEACH_[ASSY]_[yyyyMMdd][hhmmss].txt
Syntax :	TEST_XYSEARCH [Repeat Count],[StageAssy],[CH],[Step],[MaxStep]
Parameter :	[Repeat Count]: Repeat count of “XYSEARCH” command. [StageAssy]: LEFT/RIGHT/S1 [CH]: Channel (See Names of powermeter/currentmeter channels) [Step]: Step size of the scan search in μm unit [MaxStep]: Maximum step allowed during alignment (NOTE: Only the value STEP_SIZE*MAX_STEP is meaningful in the command.)
Optional Parameter(s) :	[XYSearch's option]: See “XYSEARCH” section.
Response :	See Table 1.1: Return values of commands.

Example : TEST_XYSEARCH 12,LEFT,PM1,1,15

```

### XYSearch_TEST BEGIN ###

- Repeat times: 12
- Parameters: XYSEARCH LEFT,PM1,1,15

Count  PM1    L_X    L_Y    Time(ms)
1   -23.331  -2083.50-1832.155694
2   -23.332  -2083.45-1832.155658
... ..
11  -23.307  -2083.40-1832.555658
12  -23.347  -2083.40-1832.705674

UNI 0.04      0.45    0.80
MIN -23.35    -2083.85-1832.95
MAX  -23.31   -2083.40-1832.15
AVE -23.34    -2083.57-1832.55
STD 0.01      0.17    0.21
### XYSearch_TEST END ###

```

Fig. 1.6 The result of “TEST_XYSEARCH” command

1.12.2. TEST_APPROACH

TEST_APPROACH	
Description :	Command to check the repeatability of gap control and measure the sensitivity of the displacement sensor. It performs “CUSTOM_APPROACH [AXIS],[BackdistanceAfterApproach (A)],[Back Distance(B)],[Forward_step1(C)],[Backward_step(D)],[Forward_step2(E)]” command, and records the results on the report file. In the second test, it performs approach the z stage, then move forward by 0.5 um till total distance becomes 30 um, finally moves the stage back to the last position of the first test. It record the sensor values for each z positions and the calculation results of the slope coefficient of the curve. [NOTE]: Test results are saved as the below: - Report file path: C:\IFA-600-AUCME3\TESTCMD\APPROACH\ - Report file name: TEST_APPROACH_[ASSY]_[yyyyMMdd][hhmmss].txt
Syntax :	TEST_APPROACH [Repeat Count],[Stage],[BackdistanceAfterApproach],[BackDistance],[Forward_step1],[Backward_step],[Forward_step2]
Parameter :	[Repeat Count]: Repeat count of “APPROACH” command. [Stage]: Stage name (See Name of stages) [BackdistanceAfterApproach (A)]: See “APPROACH” section. [Back Distance(B)]: See “APPROACH” section. [Forward_step1(C)]: See “APPROACH” section. [Backward_step(D)]: See “APPROACH” section. [Forward_step2(E)]: See “APPROACH” section.
Optional Parameter(s) :	[SP]: Speed (1~5, Default speed: 5)
Response :	See Table 1.1: Return values of commands.

Example :

```

TEST APPROACH 5,L_Z,10,10,2,2,0.2

### Approach_TEST BEGIN ###

- Repeat times: 5
- Parameters: APPROACH L_Z,10,10,2,2,0.2

- Repeatability
count   L_Z   Time(ms)
1    -36351.7  5747
2    -36351.5  6589
3    -36351.5  6770
4    -36351.5  6771
5    -36351.5  6727

UNI 0.2
MIN -36351.7
MAX  -36351.5
AVE -36351.5
STD 0.080000

- Measurement of the sensitivity

Count   L_Z   Sensor(mV)
1      0  4007.33
2     0.5 4000.99
... ...
60   29.5   3659.38
61   30  3653.94

slope (mV/um) : 11.8314

### Approach_TEST END ###

```

Fig. 1.7 The result of “TEST_APPROACH” command

1.12.3. TEST_CUSTOM_APPROACH

TEST_CUSTOM_APPROACH	
Description :	This command is to control the gap between a FAB and a chip precisely. First it pulls back the specified stage of “AXIS” by a certain distance(B). Secondly, it moves the stage toward the chip by a certain step(C) while the changes in the displacement sensor value is within a certain value(Threshold_2), then pull the stage backward with a step size(D) until the sensor value is smaller than (The initial value +Threshold_2). Then it repeats the previous process with a step size(E) and a threshold value(Threshold_3). Finally it pulls back the stage by a certain distance(A). <i>*The threshold values are defined in the section “SENSOR” of the file “DeviceInfo.ini”.</i>
Syntax :	TEST_CUSTOM_APPROACH [Repeat Count],[AXIS],[BackdistanceAfterApproach (A)],[Back Distance(B)], [Forward_step1(C)],[Backward_step(D)],[Forward_step2(E)]
Parameter(s) :	[AXIS]: See Name of stages [BackdistanceAfterApproach (A)]: See description above [Back Distance(B)]: See the description above. [Forward_step1(C)]: See the description above [Backward_step(D)]: See the description above [Forward_step2(E)]: See the description above

* The units of the distances, and positions are “μm” * The units of the angle, and positions are “degree”	
Optional Parameter(s):	[SP]: Speed (1~5) [TWISTBACK]: Only applicable to z stage. The software rotates ThX stage by 0.5 degree in negative direction before retreating the z stage, and restores the ThX stage position after the retreat.
Response :	See Table 1.1: Return values of commands.
Example :	TEST_CUSTOM_APPROACH 5,L_Z,50,0,10,10,0.5,SP=3 <pre> #### Approach_TEST BEGIN #### - Repeat times: 5 - Parameters: CUSTOM_APPROACH L_Z,10,10,2,2,0.2 - Repeatability count L_Z Time(ms) 1 -36351.7 5747 2 -36351.5 6589 3 -36351.5 6770 4 -36351.5 6771 5 -36351.5 6727 UNI 0.2 MIN -36351.7 MAX -36351.5 AVE -36351.5 STD 0.080000 - Measurement of the sensitivity Count L_Z Sensor(mV) 1 0 4007.33 2 0.5 4000.99 60 29.5 3659.38 61 30 3653.94 slope (mV/um) : 11.8314 #### Approach_TEST END #### </pre>

Fig. 1.8 The result of “TEST_CUSTOM_APPROACH” command

1.12.4. TEST_ANGLESEARCH

TEST_ANGLESEARCH	
Description :	Command to check the repeatability of ThX/ThY alignment. It performs “ANGLESEARCH [StageAssy], [Push Distance], [Angle_thy], [Angle_thx]” command, and records the results on the report file. [NOTE]: Test results are save in the file selected by a user during the test. - Report file path: C:\IFA-600-AUCME3\TESTCMD\ANGLESEARCH\ - Report file name: TEST_ANGLESEARCH_[ASSY]_[yyyyMMdd][hhmmss].txt
Syntax :	TEST_ANGLESEARCH [Repeat Count],[StageAssy],[Push Distance],[Angle_thy],

	[Angle thx]
Parameter :	[Repeat Count]: Repeat count of “ANGLESEARCH” command. [StageAssy]: LEFT/RIGHT [Push Distance]: distance to push FAB [Angle thy]: Total rotation angle of ThY stage [Angle thx]: Total rotation angle of ThX stage
Optional Parameter(s) :	[ANGLESEARCH's option]: see “ANGLESEARCH” section.
Response :	See Table 1.1: Return values of commands.
Example :	TEST ANGLESEARCH 3,LEFT,40,3,6 ### Anglesearch_TEST BEGIN ### - Repeat times: 3 - Parameters: ANGLESEARCH LEFT,40,3,6 count L_THX L_THY Time(ms) 1 2.842 -0.626 17710 2 2.852 -0.626 17610 3 2.842 -0.624 17609 UNI 0.010 0.003 MIN 2.842 -0.626 MAX 2.852 -0.624 AVE 2.85 -0.63 STD 0.004 0.001 ### ANGLESearch_TEST END ###

Fig. 1.9 The result of “TEST_ANGLESEARCH” command

Fig. 1.9 The result of “TEST_ANGLESEARCH” command

1.12.5. TEST_CUSTOM_ANGLESEARCH

TEST_CUSTOM_ANGLESEARCH	
Description :	Command to check the repeatability of angle alignment. It performs “CUSTOM_ANGLESEARCH [Axis1],[Axis2],[Push axis],[Angle_axis1],[Angle_axis2],[Push Distance],[Sensor channel]” command, and records the results on the report file. [NOTE]: Test results are save in the file selected by a user during the test. - Report file path: C:\IFA-600-AUCME3\TESTCMD\ANGLESEARCH\ - Report file name: TEST_ANGLESEARCH_[Axis1]_[Axis2]_[yyyyMMdd][hhmmss].txt
Syntax :	CUSTOM_ANGLESEARCH [Repeat count],[Axis1],[Axis2],[Push axis],[Angle_axis1],[Angle_axis2],[Push Distance],[Sensor channel]
Parameter(s) :	[Axis1]: Name of stages [Axis2]: Name of stages [Push axis]: Name of stages [Angle_axis1]: Total rotation angle of axis1 [Angle_axis2]: Total rotation angle of axis2 [Push distance]: push distance [Sensor channel]: Displacement sensor channel 1: Left 2: Right
Optional Parameter(s)	[CUSTOM_ANGLESEARCH's option]: see “CUSTOM_ANGLESEARCH” section.
Response :	See Table 1.1: Return values of commands.

Example : TEST_CUSTOM_ANGLESEARCH 12,L THX,L THZ,L Y,2,2,50,1

```

### Anglesearch_TEST BEGIN ###

- Repeat times: 12
- Parameters: CUTOM_ANGLESEARCH L_THX,L_THZ,L_Y,2,2,50,1

count   L_THX L_THY Time(ms)
1    2.842   -0.626 17710
2    2.852   -0.626 17610
3    2.842   -0.624 17609

UNI 0.010    0.003
MIN 2.842    -0.626
MAX 2.852    -0.624
AVE 2.85     -0.63
STD 0.004    0.001
### ANGLESearch_TEST END ###

```

Fig. 1.10 The result of “TEST_CUSTOM_ANGLESEARCH” command

1.12.6. TEST_ROLLSEARCH

TEST_ROLLSEARCH	
Description :	Command to check the repeatability of ThZ alignment. It performs “ROLLSEARCH [Stage],[CH_1],[CH_N],[CHAN_SEP],[SCAN_DIST]” command, and records the results on the report file. [NOTE]: Test results are save in the file selected by a user during the test. - Report file path: C:\IFA-600-AUCME3\TESTCMD\ROLLSEARCH\ - Report file name: TEST_ROLLSEACH [ASSY] [yyyyMMdd][hhmmss].txt
Syntax :	TEST_ROLLSEARCH [Repeat Count],[STAGE],[CH_1],[CH_N],[CHAN_SEP],[SCAN_DIST]
Parameter :	[Repeat Count]: Repeat count of “ROLLSEARCH” command. [StageAssy]: LEFT/RIGHT [CH_1]: Channel (See Names of powermeter/currentmeter channels) [CH_N]: Channel (See Names of powermeter/currentmeter channels) [CHAN_SEP]: Distance in um unit between channel 1 and channel N. [SCAN_DIST]: Distance used for “XYSEARCH” command in um unit (WARNING: This value must be an INTEGER value)
Optional Parameter(s)	[ROLLSEARCH's option]: see “ROLLSEARCH” section
Response :	See Table 1.1: Return values of commands.
Example :	TEST_ROLLSEARCH 12,LEFT,PM4,PM1,750,50

```

### Rollsearch_TEST BEGIN ###

- Repeat times: 12
- Parameters: ROLLSEARCH LEFT,PM4,PM1,750,50

count  PM4    L_X    L_Y    L_THZ  Time(ms)
1    -26.881  -2082.45 -1835.80 0.046  22454
2    -26.119  -2082.65 -1835.45 0.069  13841
... ..
11   -26.482  -2082.25 -1834.70 0.126  13961
12   -26.386  -2082.40 -1834.70 0.126  13905

UNI 1.52      0.45    1.20    0.092
MIN -27.01    -2082.70 -1835.90 0.034
MAX  -25.49   -2082.25 -1834.70 0.126
AVE -26.32    -2082.48 -1835.30 0.081
STD 0.56      0.00    0.00    1835.376
### RollSearch_TEST END ###

```

Fig. 1.11: The result of “TEST_ROLLSEARCH” command

1.12.7. TEST_CUSTOM_ROLLSEARCH

TEST_CUSTOM_ROLLSEARCH	
Description :	Command to check the repeatability of Roll alignment. It performs “CUSTOM_ROLLSEARCH [Axis1],[Axis2],[Rotation Axis],[CH_1],[CH_N],[CHAN_SEP],[SCAN_DIST]” command, and records the results on the report file. [NOTE]: Test results are save in the file selected by a user during the test. - Report file path: C:\IFA-600-AUCME3\TESTCMD\ROLLSEARCH\ - Report file name: TEST_ROLLSEACH_[Axis1]_[Axis2]_[yyyyMMdd][hhmmss].txt
Syntax :	TEST_CUSTOM_ROLLSEARCH [Repeat Count],[Axis1],[Axis2],[Rotation Axis],[CH_1],[CH_N],[CHAN_SEP],[SCAN_DIST]
Parameter :	[Repeat Count]: Repeat count of “CUSTOM_ROLLSEARCH” command. [Axis1]: Name of stages [Axis2]: Name of stages [Rotation Axis]: Roll axis direction (Name of stages) [CH_1]: Channel (See Names of powermeter/currentmeter channels) [CH_N]: Channel (See Names of powermeter/currentmeter channels) [CHAN_SEP]: Distance in um unit between channel 1 and channel N. [SCAN_DIST]: Distance used for “PEAK_SEARCH” command in um unit (WARNING: This value must be an INTEGER value)
Optional Parameter(s)	[CUSTOM_ROLLSEARCH's option]: see “CUSTOM_ROLLSEARCH” section
Response :	See Table 1.1: Return values of commands.
Example :	TEST_CUSTOM_ROLLSEARCH 12,L_Z,L_X,L_THY,PM4,PM1,750,50

```

#### Rollsearch_TEST BEGIN ####

- Repeat times: 12
- Parameters: CUSTOM_ROLLSEARCH L_X,L_Z,L_THY,PM4,PM1,750,50

count   PM4    L_X    L_Z    L_THY  Time(ms)
1    -26.881  -2082.45 -1835.80 0.046  22454
2    -26.119  -2082.65 -1835.45 0.069  13841
... ..
11   -26.482  -2082.25 -1834.70 0.126  13961
12   -26.386  -2082.40 -1834.70 0.126  13905

UNI 1.52      0.45    1.20    0.092
MIN -27.01    -2082.70 -1835.90 0.034
MAX  -25.49   -2082.25 -1834.70 0.126
AVE -26.32    -2082.48 -1835.30 0.081
STD 0.56      0.00    0.00    1835.376
#### RollSearch_TEST END ####

```

Fig. 1.12: The result of “TEST_CUSTOM_ROLLSEARCH” command

1.12.8. TEST_U_ROLLSEARCH

TEST_U_ROLLSEARCH	
Description :	Command to check the repeatability of ThZ alignment. It performs “U_ROLLSEARCH [Stage Ass’y],[Channel],[StepDegree],[ScanRange],[XYSearch_Step],[XYSearch_MaxStep]” command, and records the results on the report file. [NOTE]: Test results are save in the file selected by a user during the test. - Report file path: C:\IFA-600-AUCME3\TESTCMD\U_ROLLSEARCH\ - Report file name: TEST_U_ROLLSEACH [ASSY] [yyyyMMdd][hhmmss].txt
Syntax :	TEST_U_ROLLSEARCH [Repeat Count],[Stage Ass’y],[Channel],[StepDegree],[ScanRange],[XYSearch_Step],[XYSearch_MaxStep]
Parameter :	[Repeat Count]: Repeat count of “U_ROLLSEARCH” command. [Stage Ass’y]: LEFT/RIGHT [CHANNEL]: See Names of powermeter/currentmeter channels [StepDegree]: Step size of the scan search in degree unit. [ScanRange]: Total scan length in degree unit. [XYSearch_Step]: Step size of the scan search in μm unit for “XYSearch”. [XYSearch_MaxStep]: Maximum step allowed during alignment for “XYSearch”. [NOTE]: Only the value “STEP_SIZE*MAX_STEP” is meaningful in the command.
Optional Parameter(s)	[U_ROLLSEARCH's option]: see “U_ROLLSEARCH” section
Response :	See Table 1.1: Return values of commands.
Example :	TEST_U_ROLLSEARCH 12,LEFT,PM4,PM1,750,50

```

### U_Rollsearch_TEST BEGIN ###

- Repeat times: 12
- Parameters: U_ROLLSEARCH LEFT,PM4,PM1,750,50

count  PM4    L_X    L_Y    L_THZ  Time(ms)
1    -26.881  -2082.45 -1835.80 0.046  22454
2    -26.119  -2082.65 -1835.45 0.069  13841
... ..
11   -26.482  -2082.25 -1834.70 0.126  13961
12   -26.386  -2082.40 -1834.70 0.126  13905

UNI 1.52      0.45    1.20    0.092
MIN -27.01    -2082.70 -1835.90 0.034
MAX  -25.49   -2082.25 -1834.70 0.126
AVE -26.32    -2082.48 -1835.30 0.081
STD 0.56      0.00    0.00    1835.376
### U_Rollsearch_TEST END ###

```

Fig. 1.13: The result of “TEST_U_ROLLSEARCH” command

1.12.9. SEARCHZEXTREMA

SEARCHZEXTREMA	
Description :	This command finds the closed power minima of [CHANNEL_A] and [CHANNEL_B] in z-direction by moving the z stage step by step within the specified range, and finally moves the z stage to the minima of [CHANNEL_A].
Syntax :	SEARCHZEXTREMA [STAGE_ASSY],[CHANNEL_A],[CHANNEL_B],[STEP],[RANGE]
Parameter(s) :	[STAGE_ASSY]: Name of stage ass'y [CHANNEL_A]: one of the channels (See Names of powermeter/currentmeter channels) [CHANNEL_B]: one of the channels (See Names of powermeter/currentmeter channels) [STEP]: Step size in μm unit for the minima search [RANGE]: Maximum scan length in μm unit
Optional Parameter(s) :	[SKIPXY]: Skip XYSEARCH after a step move [NOFASTSCAN]: Use only step search for XY direction after a step move in z-direction. [ANALYSIS]: Save positions of stages for every step in Z direction(location of file: : SYSTEM DRIVE:\IFA6XX\Analysis Report\SEARCHZEXTREMA) [XYMAXSTEP]: MAXSTEP value of XYSEARCH used in this algorithm [FINDTOP]: Find maxima instead of minima [XYSTEP]: Set the step of XYSEARCH
Response :	See Table 1.1: Return values of commands.
Example :	SEARCHZEXTREMA RIGHT,PM1,PM2,0.05,1,SKIPXY,ANALYSIS → It finds the closest minima of CH1 for scans with the right stage in Z-direction. The scan will be done within 1 μm range with the step size of 0.05 μm . For every step, the alignment using XYSEARCH is skipped and the data is saved to a file.

1.12.10. TEST_ZSCAN

TEST_ZSCAN	
Description :	It moves the specified stage backward with the specified step size for certain range, and measure/save power values for every step.
Syntax :	TEST_ZSCAN [STAGE_ASSY],[STEP],[RANGE],[CHANNEL N]
Parameter :	[STAGE_ASSY]: Name of stage ass'y

	[STEP]: Step size in z-direction in um unit [RANGE]: Scan range in um unit [CHANNEL N]: Channel number to be used for alignment for every step(See Names of powermeter/currentmeter channels)
Optional Parameter(s)	[SKIPXY]: Skip XYSEARCH after a step move [NOFASTSCAN]: Use only step search for XY direction after a step move in z-direction [XYMAXSTEP]: MAXSTEP value of XYSEARCH used in this algorithm [XYPE]: The step size in μm unit used for XYSEARCH [FWD]: Move Z stage forward instead of backward * Location of data file: SYSTEM DRIVE:\IFA-600-ACAFA\Analysis Report\TEST_ZSCAN
Response :	See Table 1.1: Return values of commands.
Example :	TEST_ZSCAN RIGHT,0.05,15,PM1 → It moves the right stage backward for total 15 μm range with 0.05 μm step size. After a step move, XYSEARCH alignment is preformed, and the power of CH1 is measured/saved.

1.13. Image processing commands

The image processing requires optimization of various parameters, and the parameters are set in the file “/System/Device/ImageProcessInfo.ini”. The details on the parameter are described in a separated section.

```
[SYSTEM_INFO]
PORT=1024
[CAMERA_INFO]
CAMERA_COUNT=4
[CAMERA_1]
CAMERA_TYPE=GIGE
CAMERA_NAME=U_STD
...
[CAMERA_2]
CAMERA_TYPE=USB
CAMERA_NAME=U_DET
...
[CAMERA_3]
CAMERA_TYPE=USB2
CAMERA_NAME=REAR
...
[CAMERA_4]
CAMERA_TYPE=USB
CAMERA_NAME=FRONT
[ALGORITHM]
...
```

Fig. 1.14: Configuration file for image processing

For convenience, the image processing commands use “Camera Name” or “Camera Index” as a command parameter, and the definitions can be found from the file as shown in the figure.

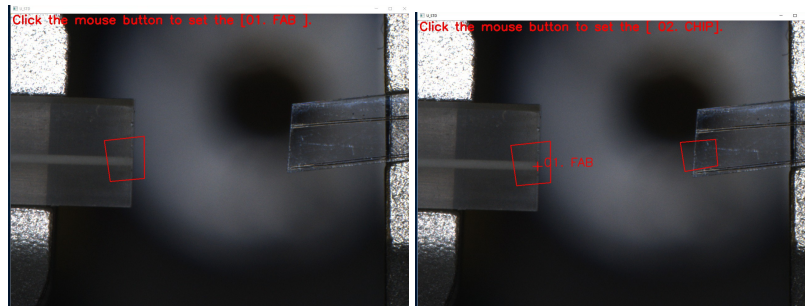
The default values are

Description	Camera index	Camera Name
Upper camera with low magnification(1.5x)	1	U_STD
Upper camera with high magnification(8.0x)	2	U_DET
Rear camera(1.5x)	3	REAR
Front camera(6.0x)	4	FRONT

1.13.1. V_SET_COORDINATE

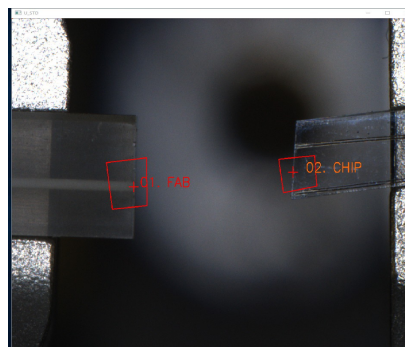
V SET COORDINATE	
Description :	This command is prepared for convenient rough alignment based on user selection. Based on the user selected positions and the coordinate of the stages, it calculate a coordinate to be used for the rough alignment, and memorize the position with the specified name("POSITION_NAME"). The obtained position can be used by "MOVE_TO_POSITION" command. In order to avoid mistakes in the user selections, various options such as "TEXT", "ROIBOX", "COLOR" are prepared. Details are described below. Moreover, the positions selected by the user should be always within the ROI box if it is used.
Syntax :	V SET COORDINATE [STAGE_ASSY],[CAM],[POSITION_NAME] V SET COORDINATE [STAGE_ASSY],[CAM],[DummyPositionName],[ROIBOX="NAME"], ROIBOX=SET
Parameter(s) :	[STAGE_ASSY]: Name of stage ass'y [CAM]: Name of cameras (if rear camera used, then the command should be used with "REARVIEW" option that allows record ONLY "Y" position.) [POSITION_NAME]: The name of the position to be used by "MOVE_TO_POSITION". With "ROISET" option, this parameter is meaningless. [DummyPositionName]: placeholder for syntax
Optional Parameter(s) :	[REARVIEW]: With this option, it record y position values in the [POSITION_NAME]. (*It should be used with REAR as [CAM] parameter.) [TEXT]: the titles of the positions selected by user mouse click, which are shown on the camera window upon each mouse click. For example, 'TEXT=FAB CHIP FAB CHIP' option displays the tiles in the order of 'CHIP'->'FAB'->'CHIP'->'FAB'->'CHIP' next to the selected point. [COLOR]: this option determines the colors of the titles of the positions to be displayed. The default settings are in the order of RED->ORANGE->YELLOW, and it is equivalent to COLOR=RED ORANGE YELLOW. The available colors are RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, PURPLE, BLACK, WHITE. [ROIBOX=ROINAME]: The name of the ROI box to be used/set in this command. When this option is not given, the ROI box will not be shown on the screen. [ROISET]: This option can be used to define ROI box with the name of "ROINAME". If a ROIBOX is set with the same name, a message window will be displayed as shown below. [ROITETRAGON]: Tetragons are used to define the regions of interest(ROIs) instead of squares. Four clicks on the screen are required for single tetragon. This option is valid only with "ROISET" option. [ROI ZOOM]: With this option, this command displays the magnified camera image with ROIs overlayed, which may help the user click on the screen more precisely. [ROI SIZE]: the size of "ROIBOX" in pixel unit can be set when this option is used with "ROISET" option. The size remains valid until this option is used.
Response :	See Table 1.1: Return values of commands.
Example :	V SET COORDINATE LEFT, UPPER, Input_Position, ROIBOX=LEFT_ALIGN,

COLOR=RED|ORANGE, TEXT=01.FAB|02.Chip → When this command is executed, it shows the first ROI box in the upper camera window for the user selection of the position related with the left FAB(a). After the user's selection, it show the second ROIBOX to assist the operator's selection of the position related with the chip. When the user click a position in the camera window, a cross line is drawn with a title. The titles are "01.FAB", "02.Chip". After all the selection are done, the message box for confirmation is displayed(d). Finally the program calculate the position of the FAB with the user selection and the coordinates of the stages.

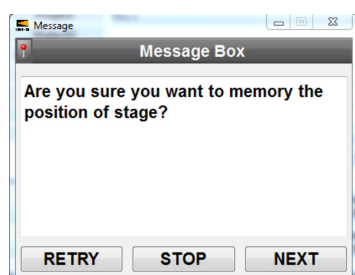


(a) ROI box for FAB position

(b) ROI box for chip position

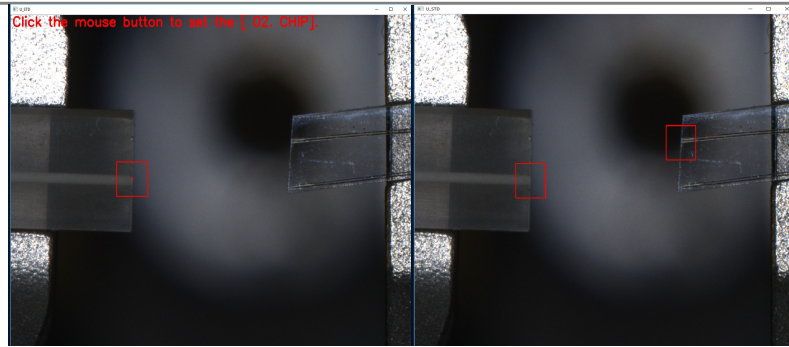


(c) ROI boxes in the upper camera window



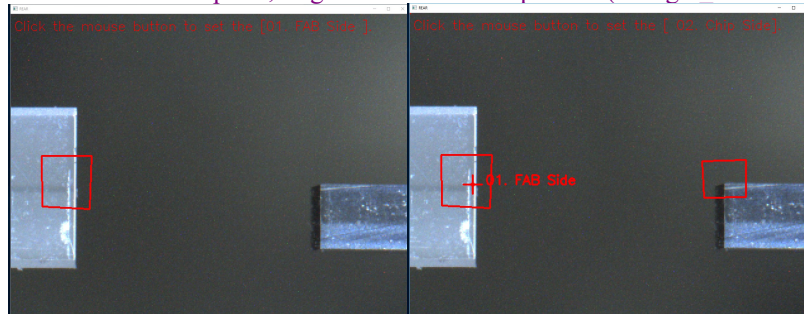
(d)

V_SET_COORDINATE LEFT, UPPER, dummy, ROIBOX=LEFT_ALIGNMENT, ROISSET, ROISIZE=100 → This command is to register ROI boxes for the L_X,L_Z stages alignment. Upon on each user click, it shows a ROI box with the size of 200 pixel in each side. With ROISSET option, it ignores the name of position("dummy")



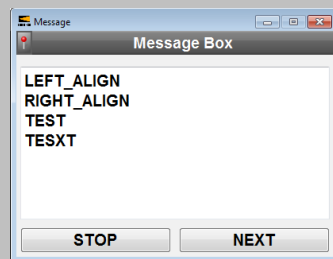
V_SET_COORDINATE LEFT,U_STD,Height_Position, ROIBOX=REAR_ALIGN, REARVIEW, COLOR=RED|ORANGE, TEXT=01.FAB Side|02.Chip side →

It is to register ROI boxes for the L_Y,L_Z stages alignment. Upon on each user click, it shows a ROISSET option, it ignores the name of position("Height Position").



1.13.2. SHOW_ROIBOX_LIST

SHOW ROIBOX LIST	
Description :	Display the list of regions of interest(ROIs) created by V_SET_COORDINATE command on a message window.
Syntax :	SHOW_ROIBOX_LIST
Parameter(s) :	NONE
Response :	See Table 1.1: Return values of commands.
Example :	SHOW_ROIBOX_LIST



1.13.3. V_DRAWCROSSLINE

V_DRAWCROSSLINE	
Description :	It draws a cross line on the specified camera viewer.
Syntax :	V_DRAWCROSSLINE [CAM],[ON/OFF]
Parameter(s) :	[CAM]: Camera name (See Name of cameras) or index(1~3) [ON/OFF]: Show/hide the cross line
Optional	[TYPE]: CENTER / MOUSE / POS(Default: CENTER)

Parameter(s) :	- CENTER: With this option, it draws a cross line at the center - MOUSE: With this option, it draws a cross line based on mouse input - POS: With this option, it draws a cross line based on the specified position(POSX, POSY)
Response :	See Table 1.1: Return values of commands.
Example :	V_DRAWCROSSLINE 1,ON V_DRAWCROSSLINE UPPER_1, ON, TYPE=POS, POSX=0, POSY=0

1.13.4. V_ADDCROSSLINE

V_DRAWCROSSLINE	
Description :	This command allows a user to draw/delete a group of lines on the specified camera viewer by using coordinates and angle or mouse drag and click. The lifetime of all the line drawn by this command ends when the program is terminated.
Syntax :	V_ADDCROSSLINE [CAM],[ADD/DEL],[GROUP NAME],[ANGLE],[X coordinate],[Y coordinate]
Parameter(s) :	[CAM]: See Name of cameras or index (1~3) [ADD/DEL]: adding/deleting a line or a group of lines [GROUP NAME]: a name of a group of line, which is used to add/delete line(s) [ANGLE]: Angle of the line to be drawn (no need for DRAG option) [X coordinate]: X coordinate of the line passes (no need for DRAG/CLICK option) [Y coordinate]: Y coordinate of the line passes (no need for DRAG/CLICK option)
Optional Parameter(s) :	[DRAG]: With this option, it draws a line using mouse drag [CLICK]: With this option, it draws a line using mouse click [RELATIVE]: With this option, the program draws a line with the relative angle of "ANGLE". [COLOR]: line color (RED/GREEN/BLUE/BLACK/WHITE or R(0~255):G(0~255):B(0~255)); Default:RED [THICK]: line thickness; default:1
Response :	See Table 1.1: Return values of commands.
Example :	V_ADDCROSSLINE 1, ADD, LINEGROUP1, 45, 300, 240, COLOR=BLUE, THICK=2 → It draws a blue line with the thickness of 1 that passes the position (300, 240) with 45 degree angle. the line belongs to the group "LINEGROUP1" V_ADDCROSSLINE 1, ADD, LGROUP2, 90, CLICK → It draws a line passing the mouse click point with 90 degree. the line belongs to the group LGROUP2. V_ADDCROSSLINE 1, ADD, LGROUP2, DRAG → It draws a line in the group LGROUP2 using mouse drag. V_ADDCROSSLINE 1, DEL, LGROUP1 → It deletes all the lines in the group LGROUP1.

1.13.5. V_IMAGEDISPLAYMODE

V_IMAGEDISPLAYMODE	
Description :	It changes the mode of a camera viewer.
Syntax :	V_IMAGEDISPLAYMODE [CAM],[MODE]
Parameter(s) :	[CAM]: See Name of cameras or index (1~3) [MODE]: 1- Show the image in 1:1 scale, 0- Show the image in reduce size
Response :	See Table 1.1: Return values of commands.
Example :	V_IMAGEDISPLAYMODE 1,1

1.13.6. V_SYNC_CAMERAPOS

V_SYNC_CAMERAPOS	
Description :	It records the coordinates of the image, the stage assembly and the tool positioner for the

point(OBJECT_NAME) on the stage assembly(Stage Ass'y) shown in the specific camera viewer(CAM). The information recorded by this command is used for positioning the specific point based on vision processing.

For the operation, the user follows the next procedure:

1. The user moves the camera to the position shown the point(e.g. the edge of the FAB, tip of the gripper, center of the pattern, etc) on the specific stage assembly.
2. Then run the command, the selected camera viewer is emphasized.
3. The user clicks the specific point using the left mouse button on the viewer, then program records the coordinates of the image, the stage assembly and the tool positioner.

[NOTE]:

Please refer to the section 2.2.3 to prepare image processing.

Syntax :	V_SYNC_CAMERAPOS [Stage Ass'y],[CAM],[OBJECT_NAME]
Parameter :	[Stage Ass'y]: See Name of stage ass'y [CAM]: Camera name or index(1~2) [OBJECT_NAME]: Object name
Response :	See Table 1.1: Return values of commands.
Example :	V_SYNC_CAMERAPOS LEFT,U_STD,FIBER_TIP → It records the information to the address "FIBER_TIP" on the coordinates for the specific point on the left stage assembly.

1.13.7. V_DECODE_BARCODE

V_DECODE_BARCODE	
Description :	It reads a barcode ID from the specified camera viewer. The image processing is done with the upper camera with high magnification, and the result is displayed in the processing information window. With options, this command retry to read either the same type of barcode or different type(binary barcode type).
Syntax :	V_DECODE_BARCODE [CAM],[TYPE]
Parameter(s) :	[CAM]: camera name or index(1~3) [TYPE]: • DATAMATRIX: It identifies the "DATAMATRIX" type of barcode on the camera viewer. • CODE128: It identifies the "Code 128" type of barcode on the camera viewer.
Response :	See Table 1.1: Return values of commands.
Optional Parameter(s) :	[VAR=name of variable]: With this option, The response by the decode is recorded in the variable
Example :	V_DECODE_BARCODE U_STD,DATAMATRIX → It reads 2D matrix barcode of the chip, and update the reading to the process information. V_DECODE_BARCODE U_STD,CODE128,VAR=CHIPID → It reads code 128 barcode of the chip, and update the reading to the process information and the user variable "CHIPID".

1.13.8. SELECT_CAMERA

SELECT_CAMERA	
Description :	It selects one of the upper cameras as a source for the upper camera viewer.(Not applicable to rear camera)
Syntax :	SELECT_CAMERA [CAM]
Parameter(s) :	[CAM]: Upper camera name (See Name of cameras) or index(1~2)
Response :	See Table 1.1: Return values of commands.
Example :	SELECT_CAMERA 1

1.13.9. V_MOVE_TO_POINT

V_MOVE_TO_POINT	
Description :	This command moves the stages to the position selected by the mouse. Before using this command, “V_SYNC_CAMERAPOS” and “V_CALIB_IMAGEPIXELSIZE” commands must be done. [NOTE]: When this command is used, the camera should be in the position where the synchronization of the camera image coordinates and a certain stage assembly’s coordinates was performed previously, unless “MOVECAM” option is given.
Syntax :	V_MOVE_TO_POINT [STAGE_ASSY],[OBJECT_NAME],[CAM]
Parameter(s) :	[STAGE_ASSY]: LEFT/RIGHT/S1 [OBJECT_NAME]: The name of the variable used with V_SYNC_CAMERAPOS [CAM]: See Name of cameras or index (1~3)
Optional Parameter(s) :	[SP]: Speed (1~5) [NOSYNC]: With this option, the user can move the specified stage by selecting two points one after another. The program moves the stage based the calculated distance from the two points and the scale factor. The synchronization of the camera image space and the physical space with V_SYNC_CAMERAPOS command is not required to use this option. [MOVECAM]: The command moves the camera to the position where the synchronization was performed before taking user’s mouse input. [TEXT]: the titles of the positions selected by user mouse click, which are shown on the camera window upon each mouse click. For example, ‘TEXT=PROBE_EDGE CHIP_LEFT’ option displays the tiles in the order of ‘PROBE_EDGE’ → ‘CHIP_LEFT’ next to the selected point. [COLOR]: this option determines the colors of the titles of the positions to be displayed. The default settings are in the order of RED→ORANGE, and it is equivalent to COLOR=RED ORANGE. The available colors are RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, PURPLE, BLACK, WHITE [OFFSET_X=distance(μm)]: With this option, the X stage moves by “the distance between two selected points + the offset distance”. [OFFSET_Z=distance(μm)]: With this option, the Z stage moves by “the distance between selected two selected points + the offset distance”. [REARVIEW]: This option should be given if the REAR camera is chosen. [SINGLE_AXIS=X or Z]: With this option, only the selected stage moves.
Response :	See Table 1.1: Return values of commands.
Example :	V_MOVE_TO_POINT RIGHT, Lensedfiber, U_STD, MOVECAM → First camera moves to the position the coordinate synchronization was done, then wait for the input from a user. Finally moves the target position. V_MOVE_TO_POINT LEFT,Lensedfiber,U_STD, NOSYNC, OFFSET_X=100 → First, select two points(object, target) on the U_STD viewer. Then • L_X stage moves by “the difference between two points + 100 μm”. • L_Z stage moves by “the difference between two points”. V_MOVE_TO_POINT LEFT, GRIPPER, REAR, REARVIEW → First, select two points(object, target) on the REAR viewer. Then ONLY L_Y stage moves by the difference between two points. V_MOVE_TO_POINT LEFT, GRIPPER, U_STD, SINGLE_AXIS=X → First, select two points(object, target) on the U_STD viewer. Then ONLY L_X stage moves by the difference between two points.

1.13.10. V_MOVE_TO_PATTERN

V_MOVE_TO_PATTERN	
Description :	This command performs template matching process with the template image taken by “V_ACQUIRE_TEMPLATEIMAGE”, and moves the stages to the specified position.

In order to handle different devices with different layout, a pattern list can be used instead of single template image. The lists of the pattern should be saved in the file “./system/Device/PatternList.ini” in advance. An example of the file content is as follows. <pre>[LD_PAT_LIST] → the name of the pattern list 1=LD_PAT_1:100:-100 → Index=PatternName:Offset_X:Offset_Z. 2=LD_PAT_2:100:-100 ... 5=LD_PAT_5:100:-100</pre>	
Syntax :	V_MOVE_TO_PATTERN [ASSY], [OBJECT_NAME], [CAM], [FILENAME_OR_PATTERNLIST],[OFFSET_X],[OFFSET_Z]
Parameter :	[ASSY]: Name of stage ass'y [OBJECT_NAME]: The name of the variable used with V_SYNC_CAMERAPOS [CAM]: Camera name (See Name of cameras) or index(1~3) [FILENAME_OR_PATTERNLIST]: The file name of template image or the name of a pattern list [OFFSET_X]: x offset from the top left corner of the template image in um unit [OFFSET_Z]: z offset from the top left corner of the template image in um unit
Optional Parameter(s)	[SKIPCHECK]: With this option, the program directly moves the detector stage without taking user selection. [MSG=string]: Message to be shown before the specified stage moves to the target position [ROI_COORD=x:y:width:height]: The region of interest(ROI) for pattern search. [CALIB_ANGLE]: The template image is rotated by the angle measured from the features in the current camera image, before it is used by template matching. [PAT_LIST]: the name of pattern list to be used for template matching.
Response :	See Table 1.1: Return values of commands.
Example :	V_MOVE_TO_PATTERN RIGHT,TX,1,TXPAT,10,20 → This command searches the template image(Template_TXPAT.png) from the camera image, and move “Probe positioner” to the position corresponding to the offset position. V_MOVE_TO_PATTERN LEFT, GRIPPER_OBJ, U_STD, LD_PAT_LIST, 0, 0, PAT_LIST, CALIB_ANGLE → 1. It searches for “LD_PAT_LIST” section from the file(“./system/Device/PatternList.ini”), then obtains the template image name(‘LD_PAT_1’) and the offset positions(100:-100) from the first item.. 2. Next, it searches for “LD_PAT_1” pattern from the camera image. If the pattern is found successfully, it moves the gripper to the position with the offset of 100um in X axis and -100um in Z axis from the pattern. 3. If the pattern is not found, it gets the information on the next pattern(e.g. LD_PAT_2) and tries the process 2. This step repeats until one of the registered pattern is found or no more pattern is not available for the try.

1.13.11. V_CALIB_IMAGEPIXELSIZE

V_CALIB_IMAGEPIXELSIZE	
Description :	This command to for the calibration of pixel size. First run the command “V_CALIB_IMAGEPIXELSIZE distance_μm”, and selects two points corresponding the distance. The calibration data is saved to the PIXELPERDIST item in the file “System/Device/ImageProcessInfo.ini”. The calibration must be done before using “V_MOVE_TO_POINT”.
Syntax :	V_CALIB_IMAGEPIXELSIZE [CAM],[DIST]
Parameter(s) :	[CAM]: See Name of cameras or index (1~3) [DIST]: Physical distance in μm unit
Response :	See Table 1.1: Return values of commands.
Example :	V_CALIB_IMAGEPIXELSIZE 1,8000

1.13.12. V_ACCUMULATE_IMAGE

V_ACCUMULATE_IMAGE	
Description :	This command captures an overlapped screenshot with the specified camera. First run “V_ACCUMULATE_IMAGE [CAM],START” to start, then run “V_ACCUMULATE_IMAGE [CAM],SHOT to take a shot as many as required. Finally run “V_ACCUMULATE_IMAGE 1,END” to save the overlapped screenshot. The image is save under /image_log”, and the format of the file name is “AccumulateImage_YYMMDDhhmmss.png”
Syntax :	V_ACCUMULATE_IMAGE [CAM],[TYPE]
Parameter :	[CAM]: See Name of cameras or index (1~3) [TYPE]: START/SHOT/END/SINGLESHOT
Optional Parameter(s) :	[NAME = “File Name”]: Set the file path of the image. [POSTPROCESS]: Save the image with the overlaid objects such as cross lines and texts
Response :	See Table 1.1: Return values of commands.
Example :	V_ACCUMULATE_IMAGE 1,START → Initiation V_ACCUMULATE_IMAGE 1,SHOT → First shot V_ACCUMULATE_IMAGE 1,SHOT → Second shot V_ACCUMULATE_IMAGE 1,END → Finalizing the screenshot V_ACCUMULATE_IMAGE 1,SINGLESHOT → Taking single shot V_ACCUMULATE_IMAGE 1,SINGLESHOT, NAME=test_image.png (file path: C:\IFA-600-AUCME3\image_log\test_image.png) V_ACCUMULATE_IMAGE 1,SINGLESHOT, NAME=../SAVE_IMAGE/test_image.png (file path: C:\SAVE_IMAGE\test_image.png)

1.13.13. V_ACQUIRE_TEMPLATEIMAGE

V_ACQUIRE_TEMPLATEIMAGE	
Description :	Take a template image to be used for the command “V_MOVE_TO_PATTERN”. The image is saved under “./image” with the format of “Template_filename.png” for the user given parameter of “filename”. The template images may be retaken if the illumination condition changes significantly.
Syntax :	V_ACQUIRE_TEMPLATEIMAGE [CAM],[FILENAME]
Parameter(s) :	[CAM]: See Name of cameras or index (1~3) [FILENAME]: file name
Response :	See Table 1.1: Return values of commands.
Example :	V_ACQUIRE_TEMPLATEIMAGE 1,TX → It take the template image from the U_STD viewer. The file name is Template_TX.png.

1.13.14. V_SHOW_FRAMERATE

V_SHOW_FRAMERATE	
Description :	This command shows the frame rate on the top-left side of the selected camera.
Syntax :	V_SHOW_FRAMERATE [CAM],[ON/OFF]
Parameter(s) :	[CAM]: See Name of cameras or index (1~3) [ON/OFF]: ON/OFF
Response :	See Table 1.1: Return values of commands.
Example :	V_SHOW_FRAMERATE 1, ON

1.13.15. SET_CAMERAOPTION

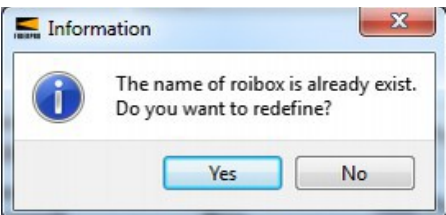
SET_CAMERAOPTION	
Description :	This command is to change the camera parameters such as GAIN, SHUTTER, and FRAMERATE. The zero value in a parameter set the parameter to automatic value. The parameters changed by this command are valid until the current program ends. For permanent change of the parameters, the parameters in the configuration file “./System/Device/ImageProcessInfo.ini” must be changed.
Syntax :	SET_CAMERAOPTION [CAM],[TYPE],[VALUE]
Parameter(s) :	[CAM]: See Name of cameras or index (1~3) [TYPE]: GAIN / SHUTTER / FRAMERATE
Response :	See Table 1.1: Return values of commands.
Example :	SET_CAMERAOPTION 1, SHUTTER, 3.5

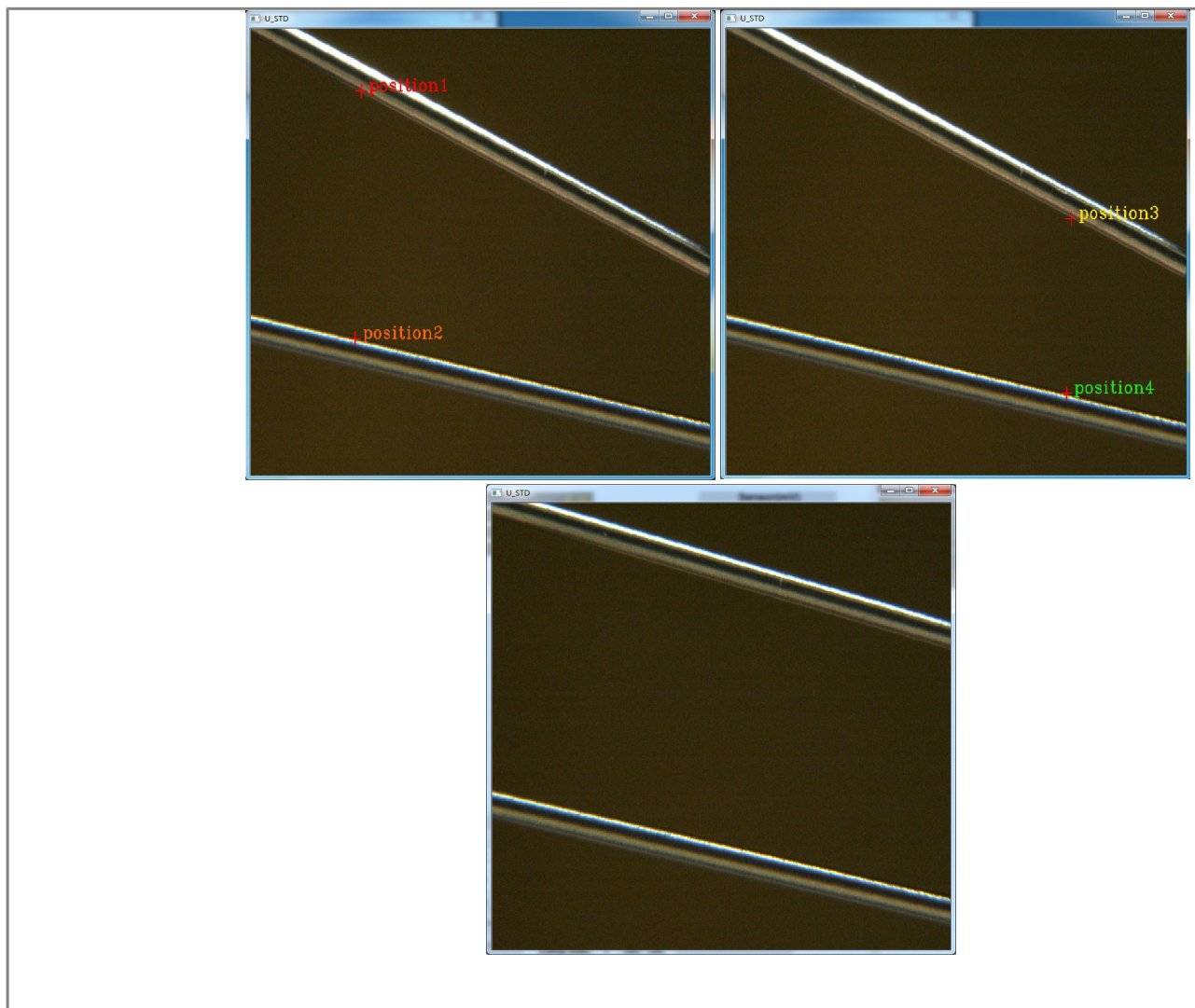
1.13.16. V_CALIB_INSTERROR

V_CALIB_INSTERROR	
Description :	This command can be used to compensate the angle misalignment between the camera axes and physical axes. Before using this command, any object whose major shape is aligned to the physical axes should be placed inside the camera view. This command takes an image, and calculates the misaligned angle by using the mathematical analysis. The calculated misaligned angle is save to [CAMERA_x.CAMERA_INSTALLATION_ERROR] section of “./System/Device/ImageProcessInfo.ini”, and used for image processing after restarting the program.
Syntax :	V_CALIB_INSTERROR [CAM]
Parameter(s) :	[CAM]: See Name of cameras or index (1~3)
Response :	See Table 1.1: Return values of commands.
Example :	V_CALIB_INSTERROR 1

1.13.17. V_SET_LINE_PARALLEL

V_SET_LINE_PARALLEL	
Description :	This commands is prepared to make an object parallel to another object with the interaction with a user. It determines the position to move stages based on the user's mouse click on the camera window. When this command starts, the program is ready to get the input from the user. The procedure the user should follow is 1. The user select a point on the outline of the object to be moved 2. The user select a point on the outline of the reference object 3. The user select another point on the outline of the object to be moved 4. The user select another point on the outline of the reference object After the user input is done, the program calculates the target position and moves the stages. With “DRAG” option, the user requires to draw two lines to be parallelized with click-and-drag actions instead of four point selection. With “MOVINGCAM_DIST=x” option, UPPER_Z stage moves by “x” μm after the first two user's point selection. This option allows to make long objects parallel. In order to avoid mistakes in the user selections, various options such as “TEXT”, “ROIBOX”, “COLOR” are prepared. Details are described below.
Syntax :	V_SET_LINE_PARALLEL [STAGE],[CAM]
Parameter(s) :	[STAGE]: Name of stages [CAM]: See Name of cameras or index (1~3)

Optional Parameter(s) :	[TEXT]: the titles of the positions selected by user mouse click, which are shown on the camera window upon each mouse click. For example, 'TEXT=TUBE_RIGHT BASE_RIGHT TUBE_LEFT BASE_LEFT' option displays the tiles in the order of 'TUBE_RIGHT'-'>'BASE_RIGHT'-'>'TUBE_LEFT'-'>'BASE_LEFT' next to the selected point. [COLOR]: this option determines the colors of the titles of the positions to be displayed. The default settings are in the order of RED->ORANGE->YELLOW, and it is equivalent to COLOR=RED ORANGE YELLOW. The available colors are RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, PURPLE, BLACK, WHITE. [ROIBOX=ROINAME]: The name of the ROI box to be used/set in this command. When this option is not given, the ROI box will not be shown on the screen. [ROISET]: This option can be used to define ROI box with the name of "ROINAME". If a ROIBOX is set with the same name, a message window will be displayed as shown below.
	
	[ROITETRAGON]: Tetragons are used to define the regions of interest(ROIs) instead of squares. Four clicks on the screen are required for single tetragon. This option is valid only with "ROISET" option. [ROIZoom]: With this option, this command displays the magnified camera image with ROIs overlayed, which may help the user click on the screen more precisely. [ROISIZE]: the size of "ROIBOX" in pixel unit can be set when this option is used with "ROISET" option. The size remains valid until this option is used. [MOVINGCAM_DIST=distance(μm)]: The distance to move "UPPER_Z" stage after the first two selection of the points. Unit: μm. [DRAG]: parallelize two objects by drawing two lines with click-and-drag action. This option can not be used with "MOVINGCAM_DIST" option. [SP]: speed (1~5, Default:4) [ANGLE="Angle" (Unit: degree)]: The target angle of an object. With this option, the user requires to draw line to measure the angle of the object with click and drag action. And it makes the angle of the object to be target angle. This option should be used with "DRAG" option.
Response :	See Table 1.1: Return values of commands.
Example :	V_SET_LINE_PARALLEL L_THY, U_STD → It makes two object parallel by moving L_THY stage. The user needs to select 4 positions in the order of position 1, 2, 3, 4 as shown in the following images



1.13.18. V_SET_PARALLEL

V SET PARALLEL

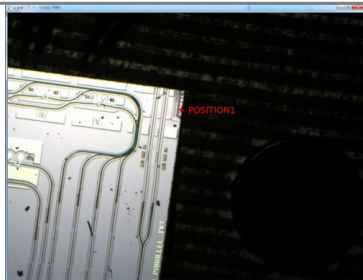
Description :

This command is prepared to make the facet of a die/bar under the test parallel to the moving direction of the CENTER_X stage by adjusting the CENTER_THY axis. However the parameters of this command are not limited to us of the stages described above(CENTER_X/CENTER_THY), and designed for the used of general cases.

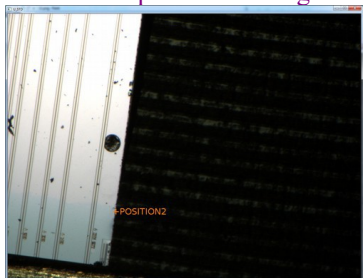
The procedure that needs to be taken after executing this command is as follows:

- When the command is executed, the program is ready to get a user mouse click at the camera window
- The user clicks a point at the edge of the object to be rotated
- Then the program moves the specified stage by [MOVING STAGE] parameter by [MOVING DISTANCE].
- The user clicks another point at the edge of the object.
- The program calculates the rotation angle for the parallelization, and rotate the corresponding stage to make angle between the line connecting the two points and the direction of [MOVING STAGE] to be [ANGLE]. If ANGLE=0, the line and the moving direction become parallel each other.

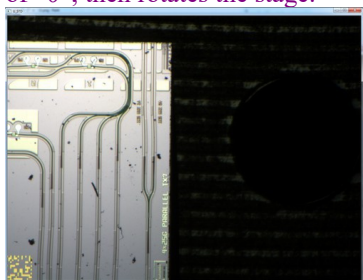
	In order to avoid mistakes in the user selections, various options such as “TEXT”, “ROIBOX”, “COLOR” are prepared. Details are described below.
Syntax :	V_SET_PARALLEL [STAGE],[CAM],[MOVING STAGE],[MOVING DISTANCE],[ANGLE]
Parameter(s) :	[STAGE]: Axis name of the LEFT/RIGHT stage ass’y (See Name of the stages and directions) [CAM]: upper camera name (See Name of cameras) or index(1~2) [MOVING STAGE]: The name of the stage to move for the parallelization(In most cases “CENTER_X”) [MOVING DISTANCE]: The distance for [MOVING STAGE] to move in μm unit. [ANGLE]: The target angle between the line connecting to selected positions and the moving direction of the specified stage.
Optional Parameter(s) :	[TEXT]: the titles of the positions selected by user mouse click, which are shown on the camera window upon each mouse click. For example, ‘TEXT=TUBE_RIGHT BASE_RIGHT TUBE_LEFT BASE_LEFT’ option displays the tiles in the order of ‘TUBE_RIGHT’->‘BASE_RIGHT’->‘TUBE_LEFT’->‘BASE_LEFT’ next to the selected point. [COLOR]: this option determines the colors of the titles of the positions to be displayed. The default settings are in the order of RED->ORANGE->YELLOW, and it is equivalent to COLOR=RED ORANGE YELLOW. The available colors are RED, ORANGE, YELLOW, GREEN, BLUE, INDIGO, PURPLE, BLACK, WHITE. [ROIBOX=ROI NAME]: The name of the ROI box to be used/set in this command. When this option is not given, the ROI box will not be shown on the screen. [ROISET]: This option can be used to define ROI box with the name of “ROI NAME”. If a ROIBOX is set with the same name, a message window will be displayed as shown below. [ROITETRAGON]: Tetragons are used to define the regions of interest(ROIs) instead of squares. Four clicks on the screen are required for single tetragon. This option is valid only with “ROISET” option. [ROI ZOOM]: With this option, this command displays the magnified camera image with ROIs overlayed, which may help the user click on the screen more precisely. [ROI SIZE]: the size of “ROIBOX” in pixel unit can be set when this option is used with “ROISET” option. The size remains valid until this option is used. [DRAG]: Use click and drag action to define the object edge instead. In this case [MOVING DISTANCE] value should be zero. [SP]: speed (1~5); default:4 [PATTERN]: With this option, a predefined pattern is automatically detected instead of using manual mouse clicks. The pattern is detected before and after moving the stage by [MOVING DISTANCE], and alignment is performed based on the center position of the detected patterns. [PATTERN2]: With this option, the system uses [PATTERN] before moving the stage, and detects [PATTERN2] after moving the stage by [MOVING DISTANCE].
Response :	See Table 1.1: Return values of commands.
Example :	V_SET_PARALLEL CENTER_THY,U_STD,CENTER_X,-5000,0,SP=5 A-1. The user selects a point on the edge of the object to be parallelized on the display with a mouse.



A-2: The program moves the specified stage(CENTER_X) by -5000 μm . The user selects another one point on the edge of the object.

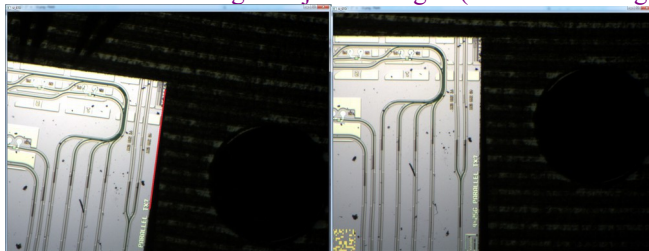


A-3. The program calculates the angle to rotate the CENTER_THY stage for the target angle of "0", then rotates the stage.



V_SET_PARALLEL CENTER_THY,U_STD,CENTER_X,0,0,SP=5, DRAG

→ Use click-and-drag to adjust the angle. (Here the moving distance value should be 0.)



1.13.19. CALIB_STAGETILTTHY

CALIB_STAGETILTTHY

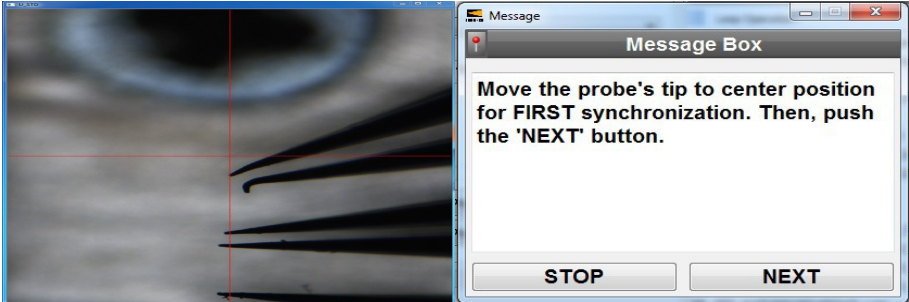
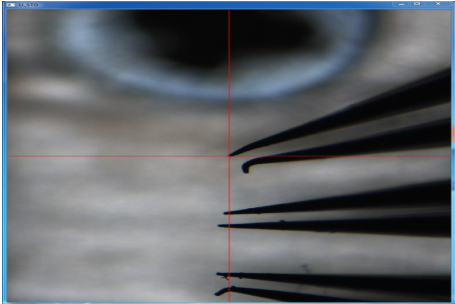
Description :

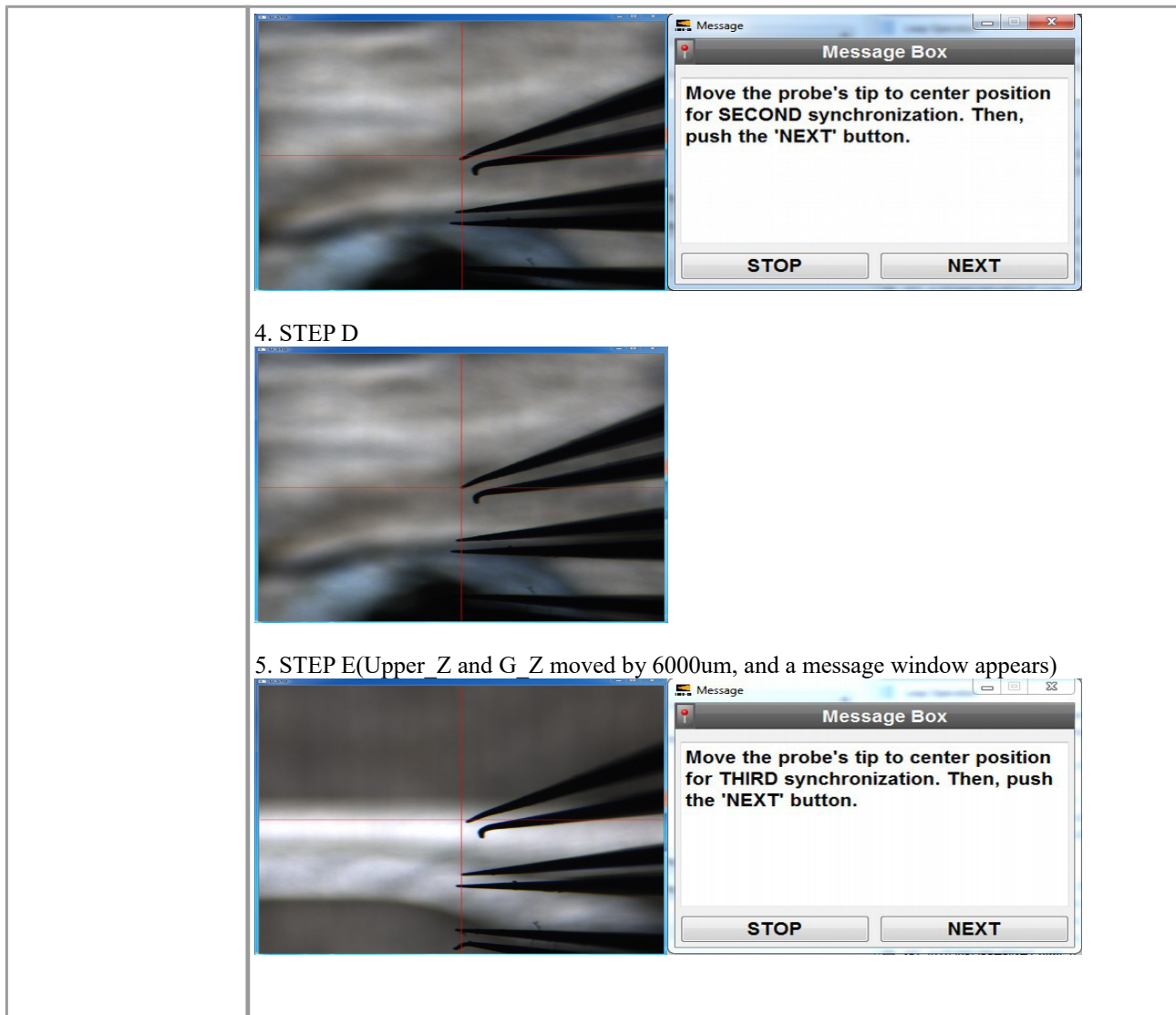
This command generates the data to compensate the alignment error in ThY direction between the XZ plane of the upper (camera) stages and either that of the alignment stage modules. The data generated by this commands must exist before one of the commands among "V_MOVE_TO_PATTERN", "V_MOVE_TO_POINT" is executed. The data is valid unless the related stages are reassembled.

This command should be used following the procedure below:

- A: The user executes this command. The program shows a cross line

- B: The user moves the stage to be used so that a specified part (tip of the part is preferred) is aligned with the cross drawn at center, and presses “NEXT” button on the screen.
- C: The program moves the upper stage and the related stage in the x direction by [X_DIST], and displays a cross on the display.
- D: The user adjusts the related stage so that the specified part is aligned with the cross drawn at center, and presses “NEXT” button on the screen.
- E: The program moves the upper stage and the related stage by in the z direction by [Z_DIST], and displays a cross on the display.
- F: The user adjusts the related stage so that the specified part is aligned with the cross drawn at center, and presses “NEXT” button on the screen.
- G: The program calculates coefficients and saves into a configuration file.

Syntax :	CALIB_STAGETILTTHY [Stage Ass'y],[CAM],[X_DIST],[Z_DIST]
Parameter :	[Stage Ass'y]: Name of stage ass'y [CAM]: See Name of cameras or index(1~3) [X_DIST]: distance to move in x direction in the step C. (Unit: um) [Z_DIST]: distance to move in z direction in the step E. (Unit: um)
Optional Parameter(s)	[SP]: speed (1~5); default:5
Response :	See Table 1.1: Return values of commands.
Example :	CALIB_STAGETILTTHY GRIPPER, U_STD, 100000, 6000 1. STEP A  2. STEP B  3. STEP C(Upper_X and G_X moved by 100000um, and a message window appears)



1.13.20. V_AUTOFOCUS

V_AUTOFOCUS	
Description :	This command performs automatic focusing function by controlling the specified motorized stage. This command measures the sharpness related values for different camera heights, moves the stage to the position with the maximum sharpness value. *This command takes the following steps • First, the program moves the Upper_Z stage backward by half scan distance. • Next, it moves the stage forward by a given step size and measures the sharpness value. • It repeat the previous procedure until the total scan distance reaches a given value. With “BEYONDMAX=n” option, it finishes the sharpness measurement after taking n steps once it passes the maximum sharpness. • Then it moves the stage to the position with the maximum sharpness. If the target position is near the boundary of the scan range, it retry the command once again from the position(the maximum number of retry is set with “RETRYMAX option”.
Syntax :	V_AUTOFOCUS [STAGE],[CAM],[STEP],[SPAN]
Parameter :	[STAGE]: See Name of stages

	[CAM]: camera name or index(1~3) [STEP]: step size(um) [SPAN]: maximum distance to scan(um)
Optional Parameter(s) :	[BWD]: With this option, it moves the stage in backward direction [LOGGING]: With this option, it saves data acquired during the operation into a file [ROI25/ROI50/ROI75]: The portion of region of interest(ROI) in the image. ROI50 implies 50% of full size in each direction [AVERAGECNT]: average count of the image acquisition(default: 1). [SAMPLINGCNT]: the number of data points used for the calculation of the optimal height (default: 2) [BEYONDMAX]: The number of steps to move the stage after the sharpness value passed the maximum value. [THRESHOLD]: Sharpness value regarded as noise level. Internal calculation does not use the measured value smaller than the threshold value. (default:0.3, range:0.0~1.0) [SP]: speed (1~5) [RETRYMAX]: The maximally allowed number of retry. [ROI_COORD=x:y:width:height]: The ROI is set by the parameters given here. [ROI_DRAG]: With this option, the user sets the ROI by dragging the mouse in the beginning of the command. [MOVING_STAGE=STAGE2]: With this option, the stage given here(STAGE2) moves to the position of the best focusing. In detail, it performs auto-focusing with the primary stage("STAGE"), then moves the primary stage back to the initial position, and finally moves the secondary stage(STAGE2) to the best focusing position with some calculation.
Response :	See Table 1.1: Return values of commands.
Example :	V_AUTOFOCUS UPPER_Z, U_DET, 10, 300, BEYONDMAX=3 → It performs auto-focus function with U_DET camera and the UPPER_Z stage with the step size of 10 um and the span size of 300 um. It stops the fine search after passing the maximum sharpness position, and goes to the maximum position. V_AUTOFOCUS UPPER_Z, U_DET, 10, 400, MOVING_STAGE=L_Y → It moves the L_Y stage for the best focusing of U_DET camera at the end after performing auto-focusing by using UPPER_Z stage with 400 um of scan range and 10 um of step size.

1.14. Remote control commands

1.14.1. GET_MASTER_PERMISSION

GET_MASTER_PERMISSION	
Description :	This command is to get the permission to control the system. IFA-600 software supports multiple clients, however, only one client can get the permission control the system.
Syntax :	GET_MASTER_PERMISSION
Parameter(s) :	NONE
Response :	See Table 1.1: Return values of commands.
Example :	GET_MASTER_PERMISSION

1.14.2. HANDOVER_MASTER_PERMISSION

HANDOVER_MASTER_PERMISSION	
Description :	This command is to return the master permission to the system, so that the operator can control the system with GUI window.
Syntax :	HANDOVER_MASTER_PERMISSION
Parameter(s) :	NONE
Response :	See Table 1.1: Return values of commands.

Example :	HANDOVER_MASTER_PERMISSION
-----------	----------------------------

1.14.3. CHECK_LASTERROR

CHECK_LASTERROR	
Description :	This command returns the error number of the last command. NOTE: The remote control commands described in this section(2.14) are excluded from the commands that generate the error number.
Syntax :	CHECK_LASTERROR
Parameter(s) :	NONE
Response :	See Table 1.1: Return values of commands.
Example :	CHECK_LASTERROR

1.14.4. CHECK_SEQSTATUS

CHECK_SEQSTATUS	
Description :	This command is to check if a sequence is running on the system. The return value is 1 if a sequence is running, and 0 otherwise.
Syntax :	CHEKC_SEQSTATUS
Parameter(s) :	NONE
Response :	1: running 0: not running
Example :	CHECK_SEQSTATUS

1.14.5. CHECK_MSGSTATUS

CHECK_MSGSTATUS	
Description :	This command is to check if a message window is active or not.
Syntax :	CHECK_MSGSTATUS
Parameter(s) :	NONE
Response :	1: a message window is active 0: no message window is active
Example :	CHECK_MSGSTATUS

1.14.6. SEQUENCE_CONTROL RUN / RESUME / STOP

SEQUENCE_CONTROL RUN / RESUME / STOP	
Description :	This command runs/stops/resumes the current sequence. This command is identical to pressing “RUN/STOP/RESUME” button on the main GUI.
Syntax :	SEQUENCE_CONTROL [RUN/RESUME/STOP]
Parameter(s) :	[RUN]: Run the current sequence. [RESUME]: Resume the current sequence. [STOP]: Stop the current sequence.
Response :	See Table 1.1: Return values of commands.
Example :	SEQUENCE_CONTROL RUN

1.14.7. SEQUENCE_LOAD

SEQUENCE_LOAD	
Description :	This command loads a sequence from a file with the given file path.
Syntax :	SEQUENCE_LOAD [SEQ_PATH]
Parameter(s) :	[SEQ_PATH]: File path of the sequence file to be loaded

Response :	See Table 1.1: Return values of commands.
Example :	SEQUENCE_RUN ./SEQ/remote_seq.sqf

1.14.8. ISBUSY?

ISBUSY?	
Description :	This command checks whether the server is ready to take a command from a client, and return a value 1 for busy state or 2 for not busy state.
Syntax :	ISBUSY?
Parameter(s) :	NONE
Response :	1 : Busy 2 : Not busy
Example :	ISBUSY?

1.14.9. CONTROL_MSGBOX

CONTROL_MSGBOX	
Description :	This command is to remotely control the message window active on the main GUI, so that the current sequence gets released from the holding status and execute next commands. It emulates the operator's clicking "NEXT" button or "STOP" button.
Syntax :	CONTROL_MSGBOX [NEXT/STOP]
Parameter(s) :	[NEXT/STOP]: - NEXT: Running the next command - STOP: Stopping sequence
Response :	See Table 1.1: Return values of commands.
Example :	CONTROL_MSGBOX NEXT → It emulates "NEXT" button click on the message window. CONTROL_MSGBOX STOP → It emulates "STOP" button click on the message window.

1.14.10. GET_VARIABLE

GET_VARIABLE	
Description :	It returns the value of the specified variable(s).
Syntax :	GET_VARIABLE [VARIABLE]
Parameter(s) :	[VARIABLE]: The name(s) of variables
Response :	the value of the specified variable(s)
Example :	GET_VARIABLE Get powers: \$(PMCH1):\$(PMCH2) → response: It gets powers: -5.123:-6.123: It returns power values of channel 1 and 2.

1.14.11. CHECK_STATUSMASK

CHECK_STATUSMASK	
Description :	This command returns an integer value that including various status of the system in forms of bit flags. Bit order is as following: [MSB] [LSB] [Emergency status][Busy status][Msg dialog status][Seq running status][Permission status] Each bit contains the following status: [Emergency status]: 1=Emergency, 0= No emergency

	[Busy status]: 1=busy, 0= No busy [Msg dialog status]: 1= msg enable state 0= msg disable state [Seq running status]: 1=seq running 0= seq none busy [Permission status]: 1=Master permission 0= No master permission
Syntax :	CHECK_STATUSMASK
Parameter(s) :	NONE
Response :	Integer value containing various status of the system in the form of bit flags
Example :	CHECK_STATUSMASK → Return value: 11 (in decimal number) → 0b00001011 → [Emergency status - 0][Busy status - 1][Msg dialog status- 0][Seq running status - 1] [Permission status -1]

1.15. Return codes for remote control commands

When a client runs a command through the remote operation, the server returns a return code. The return codes are described in the table below.

TYPE	Number	Description
RET_NOBUSY	2	The system is ready to receive a command. Return code only for “ISBUSY?” command
RET_BUSY	1	The system is executing a command, and not ready to take another command. Return code only for “ISBUSY?” command.
RET_SUCCEEDED	0	A command is completed with success.
RET_FAILED	-1	A command is completed with failure.
RET_STOP	-100	A command is completed by user interrupt Return code obtained by “CHECK_LASTERROR” command after the last command has been stopped by “SEQUENCE_CONTROL_STOP” command.
RET_EIO	-99	I/O error in communication with peripherals
RET_EIO_READ	-98	I/O read error in communication with peripherals
RET_EIO_WRITE	-97	I/O write error in communication with peripherals
RET_EINVALARG	-96	Invalid argument
RET_EFILE	-95	File handling error
RET_UNKNOWNCMD	-94	Unknown command
RET_VAR_ALREADY_EXISTED	-92	Failed to create a variable since a variable of the same name is existing.
RET_VAR_NOT_EXISTED	-91	The variable does not exist.
RET_VAR_NOT_PERMITTED	-90	No permission to handle the variable
RET_VAR_ISINTERNAL	-89	Tried to change a system variable.
RET_VAR_ISSTRING	-88	Failed to perform a numeric operation.
RET_VAR_INVALID_DELIMITER	-87	Invalid delimiter
RET_HOMING_REQUIRED	-56	The system needs to do the homing process.
RET_NO_PATTERN	-55	Failed to find the specified pattern from the image.

2. Using Control Software

2.1. Using software

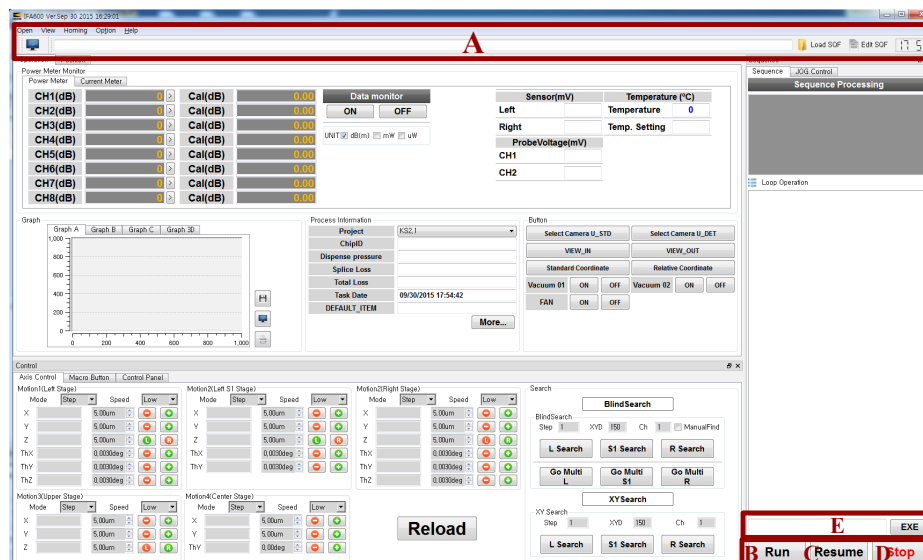


Fig. 2.1: Main window

2.1.1. Loading a sequence

The software allows to run a series of commands(“Sequence”). Normally a sequence for the specific processes is saved in a file. A user can load a sequence file by pressing the “Load SQF” button in the sequence bar(A) and selecting a file (*.SQF). The path of the selected file is shown in the sequence bar. A user can edit loaded sequence file by pressing the “Edit SQF” button in the sequence bar(A).

2.1.2. Run/Stop/Resume of a sequence

A user can run the selected sequence by pressing the “RUN” button(B) on the right bottom side of the window. Pressing the “STOP” button(D) interrupts the command currently under operation. Pressing the “RESUME” button(C) restarts the sequence from the command selected/marked by a green icon. The user can select a command in the sequence window by double-clicking a command.

2.1.3. Using command execution window

The command execution window(E) can be used to execute a command by entering the command with parameters into the edit window and pressing the “EXE” button.

Double-clicking one of the commands in the sequence manager loads the selected command onto the edit window. Pressing the upper arrow button

2.2. Preparing image processing

In order to use the commands related with image processing(Vision processing), some preparations need to be done in advance.

2.2.1. Acquiring template images

The system supports a function(“template matching”) that finds a small image(“template pattern”) from the camera viewer. Therefore the template patterns need to be taken from the camera images with “V_ACQUIRE_TEMPLATEIMAGE” command(see 1.13.13).

The examples are:

- `V_ACQUIRE_TEMPLATEIMAGE U_STD,SAMPLE1`
- `V_ACQUIRE_TEMPLATEIMAGE U_DET,SAMPLE2`

The example commands create the images files “Template_SAMPLE1.png” and “Template_SAMPLE2.png”

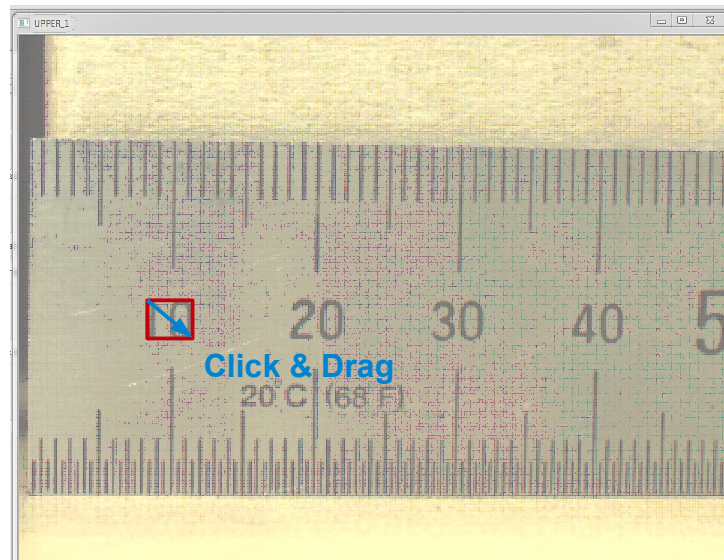


Fig. 2.2: Acquiring a template image

The procedure is:

- First, the user executes the command. Then the camera window related with the selected camera expands and waits for the user selection for the image acquisition.
- Next the user moves the mouse cursor to the top-left corner of the template image to be taken. Then press the left mouse button and drag the mouse to the bottom-right corner of the image.
- Finally the user releases the button. After the selection, a template image is saved to a file following the naming convention.

2.2.2. Camera pixel size calibration

For image processing, the physical size corresponding to an image pixel (“scale factor”) is one of important parameters required in advance. The command “V_CALIB_IMAGEPIXELSIZE” can be used to obtain the scale factor. The procedure to obtain the scale factor is:

- First the user executes the command “V_CALIB_IMAGEPIXELSIZE [CAMERA],[DISTANCE]”, where [DISTANCE] is the physical distance between two points to be chosen from the camera image in μm unit.
- Then the camera window related with the selected camera expands and waits for the user selection of two points.
- The user clicks two points with the mouse. The physical distance between the points should be equal to [DISTANCE]
- After the selection, the software calculates the scale factor for the selected camera and saves the value into a file.

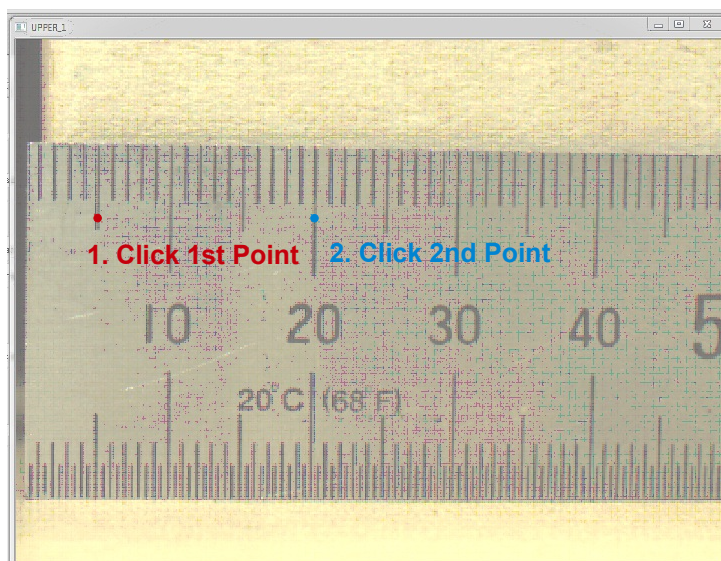


Fig. 2.3: Obtaining scale factor of camera image

2.2.3. Mapping camera image position to real position

Mapping a position on a camera image to a physical position is one of the important thing to be done in advance. Normally it needs to be done with thin or sharp object. The procedure is as follows:

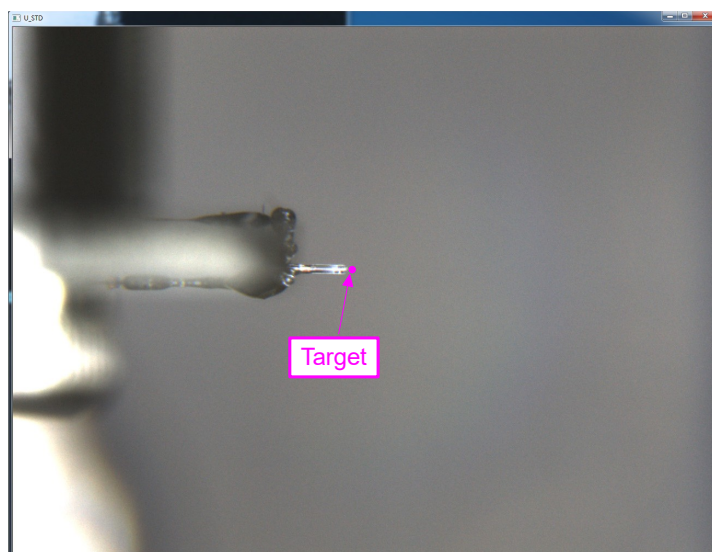


Fig. 2.4: Mapping a position on a camera image to a physical position

1. The user adjusts the height of the upper camera for the sharp image of a probe tip.
2. The user executes “V_SYNC_CAMERAPOS [STAGE_ASSY],[CAM],[OBJECT_NAME]” to start the mapping process.
3. The user clicks the target point with left button of the mouse. The information on the coordinates for the target point is recorded.

[WARNING] The mapping data become INVALID if the camera position is adjusted manually. Therefore manual position adjustment of the cameras must be avoided if possible.

2.3. Example: Sequence for alignment/bonding

2.3.1. Example sequence

An example sequence for the lens alignment/bonding is shown below. (Edge coupling with lensed fiber)

```

POSITION
POS_START_LEFT:4495.6000:2330.3000:-10233.8800:-1.3000:0.1596:0.0000
POS_START_S1:10000.0000:0.0000:13741.5200:0.0000:0.0000:0.0000
POS_START_RIGHT:7525.7600:3488.9600:-11622.0000:-1.3000:2.5783:0.0000
POS_START_CENTER:-6292.6000:0.0000:0.0000
POS_MULTI_LEFT:0:0:0:0:0
POS_MULTI_S1:0:0:0:0:0
POS_MULTI_RIGHT:0:0:0:0:0
POS_TOOL_INPUT_ALIGN:10004.4000:5037.6800:20997.2000:13517.7800:159.3200:-
3319.3200:0.0000:0.0000:10000.0000:0.0000:13741.5200:0.0000:0.0000:0.0000
POS_TOOL_OUTPUT_ALIGN:1004.4000:4887.6800:44977.0000:7742.1200:1421.5000:-5890.3400:0.0000:0.0000:0.0000
VIEW_IN:1004.4000:4508.4800:28285.2000:1
VIEW_OUT:1004.4000:4508.4800:38122.4000:1
VIEW_DUT:::1
TOOL_01:::1
TOOL_02:::1
POLARIZER_POSITION_01:::1
POLARIZER_POSITION_02:::1
UV_CURING_IN:::1
UV_CURING_OUT:::1
EPOXY_IN:::1
EPOXY_OUT:::1
BARCODE_READING:::1
ALIGN_CENTER_POS:::1
POSITIONEND

SETUPOPERATION
Set Initial Condition // Command group name
    SPLASH_MSG START // This command displays a message window for a certain period of time.(Default: 5 seconds)
    REARCAMERA BACK// It moves the rear camera to the back position.
    V_ADDCROSSLINE U_STD,DEL,ALL// Remove all the lines on the U_STD viewer.
    V_ADDCROSSLINE REAR,DEL,ALL // Remove all the lines on the REAR viewer.
    CLEAR_PROCESSINFO // Clear the selected items of the process Information window, and set those value with the
default values.
    RELOAD // This command moves the stages to a predefined position(Reload Position)
Camera Setting
    SET_LAMP 1,100 // Set the brightness of the LED lamp mounted on the upper camera(1.5x).
    SET_LAMP 2,100 // Set the brightness of the coaxial lamp mounted on the upper camera(8.0x).
    SET_LAMP 3,10 // Set the brightness of the LED lamp mounted on the rear camera.(1.5x)
    SET_LAMP 4,0 // Set the brightness of the coaxial lamp mounted on on the upper camera(1.5x).
    SET_CAMERAOPTION U_STD,GAIN,10 // Set the camera condition to automatic gain control mode.
    SET_CAMERAOPTION U_STD,SHUTTER,30 // Set the camera condition to automatic shutter control mode.
SETUPOPERATIONEND

LOOPOPERATION
Loading Device
    MSG Load the lensed fiber jig on the left and right stage assy.
    SET_VACUUM 2,ON // Command to turn on vacuum channel installed on the optical table.
    MSG Load the device on the center jig.
    SET_VACUUM 2,OFF // Command to turn off vacuum channel installed on the optical table.
    MSG Load the center jig on the center stage assy.
        MOVE_TO_POSITION VIEW_IN // Move a tool positioner to predefined position to see input side of the
bar by an upper camera(1.5x).
        STARTPOSITION CENTER_X,CENTER_Y,CENTER_THY // Move X/Y/ThY axis of the center stage assy. to the
predefined position.

```

V_SET_PARALLEL CENTER_THY,U_STD,CENTER_X,-5000,0,SP=5 // Make the facet of a bar parallel to the moving direction of the CENTER_X stage by adjusting the CENTER_THY stage.

Set Fiber Angle

INPUT_ALIGN_POS // It moves tool positioner, left stages to the predefined position for set the fiber angle.

REARCAMERA FRONT

ABSMOVE L_THX,-1.3 // Make a lensed fiber loaded on the Left stage assy. to be tilted 16.7 degrees to the ThX direction.

V_SET_LINE_PARALLEL L_THY,U_STD,DRAG,ANGLE=0 // Make a lensed fiber loaded on the Left stage assy. to be tilted 0 degrees to the ThY direction.

REARCAMERA BACK

OUTPUT_ALIGN_POS // It moves tool positioner, right stages to the predefined position for set the fiber angle.

REARCAMERA FRONT

ABSMOVE R_THX,-1.3 // Make a lensed fiber loaded on the Right stage assy. to be tilted 16.7 degrees to the ThX

V_SET_LINE_PARALLEL R_THY,U_STD,DRAG,ANGLE=0 // Make a lensed fiber loaded on the Right stage assy. to be tilted 0 degrees to the ThY direction.

REARCAMERA BACK

Initial positioning

MOVE_TO_POSITION VIEW_IN

STARTPOSITION L_X,L_Y,L_Z,R_X,R_Y,R_Z,CENTER_X,CENTER_Y // Move X/Y/Z axis of left and right stage and X/Y/ThY axis of the center stage assy. to the predefined position

REARCAMERA FRONT

V_SET_COORDINATE LEFT,U_STD,Input_Position, ROIBOX=LEFT_ALIGN, COLOR=RED|ORANGE,
TEXT=01.LENSED FIBER|02.Chip,ROI_ZOOM // Prepare the rough alignment by user selection. (Refer to 1.12.1 V_SET_COORDINATE)

#V_SET_COORDINATE LEFT,U_STD,Input_Position, ROIBOX=LEFT_ALIGN, COLOR=RED|ORANGE,
TEXT=01.LENSED FIBER|02.Chip,ROISET,ROITETRAGON // Prepare the rough alignment by user selection. (Refer to 1.12.1 V_SET_COORDINATE)

MOVE_TO_POSITION Input_Position,OFFSET_Z=50,SP=4 // Make the lensed fiber loaded on the left stages to be closed to an input port.

MSG Manually adjust the L_Z/L_Y-position in detail.

#V_SET_COORDINATE LEFT,REAR,Input_Position_Y, ROIBOX=LEFT_ALIGN_YZ, COLOR=RED|ORANGE,
TEXT=01.LENSED FIBER|02.Chip,ROI_ZOOM,REARVIEW // Prepare the rough alignment by user selection. (Refer to 1.12.1 V_SET_COORDINATE)

#V_SET_COORDINATE LEFT,REAR,Input_Position_Y, ROIBOX=LEFT_ALIGN_YZ, COLOR=RED|ORANGE,
TEXT=01.LENSED FIBER|02.Chip,ROISET,ROITETRAGON,REARVIEW // Prepare the rough alignment by user selection. (Refer to 1.12.1 V_SET_COORDINATE)

#MOVE_TO_POSITION Input_Position_Y,OFFSET_Y=70,SP=4 // Make the lensed fiber loaded on the left stages to be closed to an input port.

REARCAMERA BACK

MOVE_TO_POSITION VIEW_OUT

REARCAMERA FRONT

V_SET_COORDINATE RIGHT,U_STD,Output_Position,ROIBOX=S1_ALIGN,COLOR=RED|ORANGE,
TEXT=01.LENSED FIBER|02.Chip,ROI_ZOOM // Prepare the rough alignment by user selection. (Refer to 1.12.1 V_SET_COORDINATE)

#V_SET_COORDINATE RIGHT,U_STD,Output_Position,ROIBOX=S1_ALIGN,COLOR=RED|ORANGE,
TEXT=01.LENSED FIBER|02.Chip,ROI_ZOOM,ROISET,ROITETRAGON // Prepare the rough alignment by user selection. (Refer to 1.12.1 V_SET_COORDINATE)

MOVE_TO_POSITION Output_Position,OFFSET_Z=70,SP=4 // Make the lensed fiber loaded on the right stages to be closed to an output port.

MSG Manually adjust the R_Z/R_Y-position in detail.

#V_SET_COORDINATE RIGHT,REAR,Output_Position_Y, ROIBOX=RIGHT_ALIGN_YZ, COLOR=RED|ORANGE,
TEXT=01.LENSED FIBER|02.Chip,ROI_ZOOM,REARVIEW

#V_SET_COORDINATE RIGHT,REAR,Output_Position_Y, ROIBOX=RIGHT_ALIGN_YZ, COLOR=RED|ORANGE,
TEXT=01.LENSED FIBER|02.Chip,ROISET,ROITETRAGON,REARVIEW

#MOVE_TO_POSITION Output_Position_Y,OFFSET_Y=70,SP=4

REARCAMERA BACK

Rough Optical Alignment

MOVE_TO_POSITION VIEW_IN

REARCAMERA FRONT

XYBLINDSEARCH LEFT,PM1,50,100,10,TRGSP=4,YTRIG,MANUALFIND // Align the lensed fiber loaded on the LEFT stages with the input port roughly.

```

REARCAMERA BACK
MOVE_TO_POSITION VIEW_OUT
REARCAMERA FRONT
XYBLINDSEARCH RIGHT,PM1,50,100,10,TRGSP=4,YTRIG,MANUALFIND // Align the lensed fiber loaded on the
RIGHT stages with the output port roughly.
REARCAMERA BACK
MOVE_TO_POSITION VIEW_IN
REARCAMERA FRONT
PAUSE 1
ZOPTIMIZE
LEFT,0.5,10,PM1,ZBEYONDMAX=2,FWD,STEPMIN=0.2,STEPPDIST=6,MAXSTEP=20,STEPSKIP,TRGSP=3 // Find a peak
positon to Z-direction of the input side roughly.
REARCAMERA BACK
MOVE_TO_POSITION VIEW_OUT
REARCAMERA FRONT
ZOPTIMIZE
RIGHT,0.5,10,PM1,ZBEYONDMAX=2,FWD,STEPMIN=0.2,STEPPDIST=6,MAXSTEP=20,STEPSKIP,TRGSP=3 // Find a peak
positon to Z-direction of the output side roughly.
REARCAMERA BACK
XYZSEARCH_LEFT // Find a peak position of the input light and move X/Y/Z axis of the LEFT stage assy. there.
STEPSEARCH L_X,PM1,0.2,4,STEPSP=2,SLEEP=50,POLY5,REFRESH_AT_END
STEPSEARCH L_Y,PM1,0.2,4,STEPSP=2,SLEEP=50,POLY5,REFRESH_AT_END
STEPSEARCH L_Z,PM1,0.5,6,STEPSP=2,sleep=150,POLY5,REFRESH_AT_END
STEPSEARCH L_X,PM1,0.1,2,STEPSP=2,SLEEP=50,POLY5,REFRESH_AT_END
STEPSEARCH L_Y,PM1,0.1,2,STEPSP=2,SLEEP=50,POLY5,REFRESH_AT_END
XYZSEARCH_RIGHT // Find a peak position of the output light and move X/Y/Z axis of the RIGHT stage assy. there.
STEPSEARCH R_X,PM1,0.2,4,STEPSP=2,SLEEP=50,POLY5,REFRESH_AT_END
STEPSEARCH R_Y,PM1,0.2,4,STEPSP=2,SLEEP=50,POLY5,REFRESH_AT_END
STEPSEARCH R_Z,PM1,0.5,6,STEPSP=2,sleep=150,POLY5,REFRESH_AT_END
STEPSEARCH R_X,PM1,0.1,2,STEPSP=2,SLEEP=50,POLY5,REFRESH_AT_END
STEPSEARCH R_Y,PM1,0.1,2,STEPSP=2,SLEEP=50,POLY5,REFRESH_AT_END
LOOPOPERATIONEND

```

3. Remote Control of IFA-600-AUCME3

3.1. Introduction

The IFA-600-AUCME3 software provides a network service based on TCP/IP for remote operation. A simple block diagram is shown in the figure below.

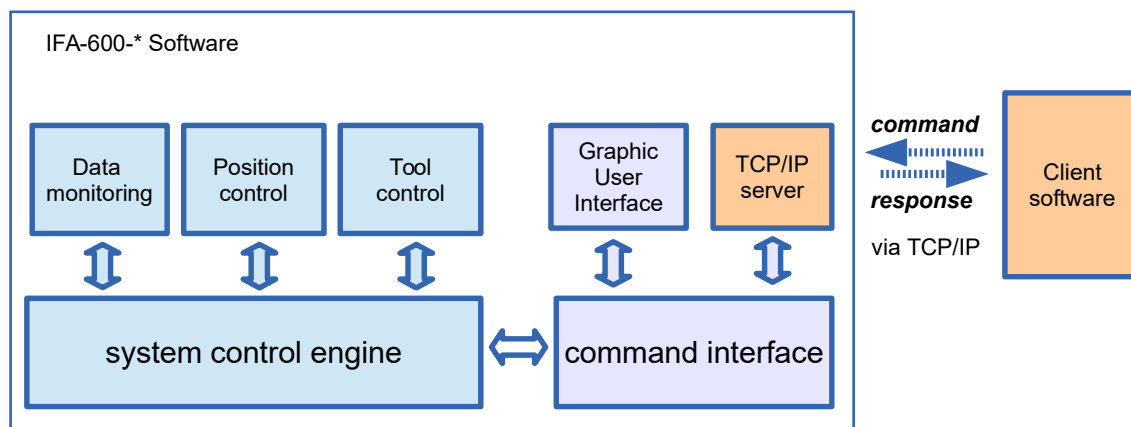


Fig. 3.1: Remote operation of IFA-600-AUCME3 software

The advantages of a TCP/IP based server are

- Easy to program/debug client programs
- Client program can run on a different host even with different operating system Readily-available terminal programs such as telnet client can be used for the remote operation

The user can control most of functions of IFA-600 including:

- moving motorized stages
- reading optical powers/ photo currents
- perform FAB alignments
- control of UV light source, dispenser, etc

The software supports the remote access for multiple clients, however only one client gets the permission to control the system at a time. The software opens a tcp port to listen to the clients at the initialization stage. The port number is defined in the [EXTERN_CLIENT_INFO] section of the file "ViewInfo.ini", and the default is 8485.

In order for a client software to control the system, the client needs to obtain the master permission using the "GET_MASTER_PERMISSION" command, which is described in the section 2.14 "Remote control commands". With the permission, the client is able to run all the commands described in the Chapter 2, and to check the result with a return value. The return codes are described in the appendix section.

The procedure for the remote control is

- Connect to the listening port
- Acquire the master permission using "GET_MASTER_PERMISSION"
- Send a command to run (e.g. SET_VACUUM 1, ON)
- Obtain the return code
- Send another command and get the return code
- Disconnect the connection

3.2. Example code

```
using namespace std;

int main()
{
    string strServerIP = "127.0.0.1";
    int port_num = 1025;
    int retv;
    const char* sztmpdata = strServerIP.c_str();
    char* szServerIP = (char *)sztmpdata;
    std::string msg;

    Client client(szServerIP);

    client.Send("GET_MASTER_PERMISSION");
    ret_v = client.Recv();
    client.Send("SET_VACUUM 1, ON");
    ret_v = client.Recv();
    client.Stop();
    return 0;
}
```

Fig. 3.2: Example code for the remote control

4. Variable Manager

Variable manager provides flexibility in data saving, command control, message dialog, and others. There are two types of variables: user-defined variable and system variable.

The user-defined variables can be created/modified/removed by the user. The lifetime of the user-defined variables ends when the software terminates. The most system variables can not be created/modified/removed by the user, and are updated by system during the operation.

The commands for handling user-defined variables are described in “Variable manage” section of “Control Commands”. In order to use a variable, for example, “MYVAR”, the user needs to call the variable in form of “\$ (MYVAR)”. For example, MSG \$(MYVAR) command creates a message dialog with the message stored in the variable “MYVAR”.

4.1. List of system variables

Name	Description	Modification by command	Linked to Process information windows
SYSID	System ID	N	N
PMCHx	Optical power value of the channel x(x=1..8) Ex) PMCH1 - PMCH4	N	N
PMCALx	Powermeter calibration value of the channel x Ex) PMCAL1 - PMCAL4	N	N
CMCHx	current value of the channel x Ex) PMCH1 - PMCH8	N	N
CMCALx	currentmeter calibration value of the channel x(x=1..4) Ex) CMCAL1 - CMCAL4	N	N
DAQ_CHx	Analog input channel x of SC7000(x= 3,4) Ex) DAQ_CH3 / DAQ_CH4	N	N
L X - UPPER Z	Coordinates of stages	N	N
LOOPMAX	The maximum number of repeat loop (Default=1)	Y	N
LOOPCNT	The number of loop repetition executed from the start	Y	N
ALL	Reserved keyword implying all user variables	N	N
DECIMAL0 - 6	Set number of digit below decimal point Ex) DECIMAL0 - DECIMAL6	N	N
YYYY	This year in 4 digits	N	N
YY	Last 2 digits of this year	N	N
MM	This month	N	N
DD	Date of today	N	N
HH	Hour	N	N
MI	Minute	N	N
SS	Second	N	N
TASKDATE	Current time(MM/dd/yyyy hh:mm:ss)	N	Y
UVCURINGTIME	UV curing time in second	Y	Y

SENSOR_CHx	Displacement sensor value of channel x(x=1.2) SENSOR_CH1: Displacement sensor of left assy. SENSOR_CH2: Displacement sensor of right assy. Ex) SENSOR_CH1, SENSOR_CH2	N	N
SP	Source power	N	N
TEMPERATURE	Measured temperature of the TEC	N	N
TARGET_TEMP	Target temperature of the TEC	N	N
SEQUENCE_NAME	Currently loaded sequence name	N	N
SYSTEM_NAME	System ID (IFA-600-AUCME3)	N	N

5. Appendix

5.1. Files used by software

5.1.1. Configuration file for the process information window

- FILENAME: ProcessInfo.ini
- LOCATION: ./System/Device/

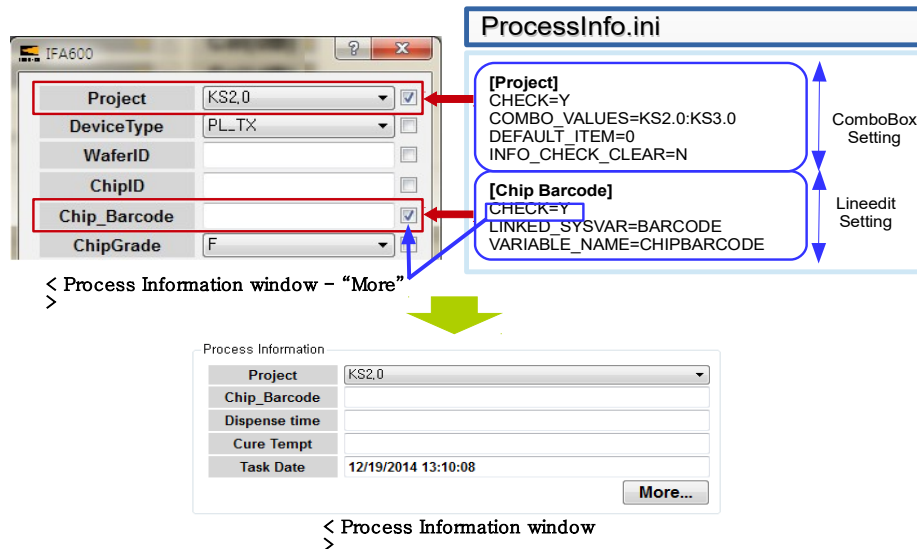


Fig. 5.1: File: ProcessInfo.ini

- [Section Name]: Sets the name of the item.
 - The items to be shown in the Process info window are defined in this section.
 - Each item needs the properties of CHECK, COMBO_VALUES, DEFALUT_ITEM, INFO_CHECK_CLEAR, LINKED_SYSVAR, FIXNAME, VARIABLE_NAME, LONGNAME, to be set, where n is the id of the item.
 - **CHECK**: Sets whether the item is to be show in the main window.
 - “Y”: The item is shown in the main window.
 - ” N”: The item is shown only in the process info window
 - **COMBO_VALUES**: Set the values of the item in the combo box form. The values of the combo box should be given in the “:” separated format(e.g. COMBO_VALUES=ProjectA:ProjectB). If this prop-erties is not defined, the item will be shown in “Editline” type.
 - **DEFAULT_ITEM**: Set default item index of the combo box. For “LineEdit” type, it sets default string to be displayed.
 - **DISPLAY_VAIRABLE**: Set whether the value of the variable written on the “Default_Item” section is to be show on the section. And the displayed value is updated every 1 second.
 - “Y”: Display variable value on the section
 - “N”: Display the characters of the variable on the section instead of the value.
 - Examples:
 - Case 1. Monitoring single variable on the “Loop Count” section

```
[Loop Count]
CHECK=Y
INFO_CHECK_CLEAR=Y
DEFAULT_ITEM=LoopCont $(LOOPCNT)
DISPLAY_VARIABLE=Y
```
 - Case 2. Monitoring multiple variables on the “Display Variable” section

```
[Display Variable]
```

```

CHECK=Y
INFO_CHECK_CLEAR=Y
DEFAULT_ITEM=LoopCnt $(LOOPCNT) / CH1 $(PMCH1)
DISPLAY_VARIABLE=Y

```

- **INFO_CHECK_CLEAR:**
 - “Y”:
 - The “MAKE_REPORT” command fails when the corresponding item is empty.
 - The “CLEAR_PROCESSINFO” command clears the corresponding item and assign the default value.
 - “N”:
 - The system performs “MAKE_REPORT” command even if the corresponding time is empty.
 - “CLEAR_PROCESSINFO” has no effect on the corresponding time.
- **LONGNAME:** This parameter sets the title of the item to be written in a report file. If this parameter is empty, “Section Name” is used in the report file.
- **VARIABLE_NAME:**
 - The item value can be assigned to a variable. This parameter defines the name of the variable that can be used in report files and message boxes.
 - The variable name is case insensitive and cannot include space.
- **LINKED_SYSVAR:**
 - This keyword connects the corresponding item to a system variable. It allows for the system variables modified by some commands to be updated to the corresponding item in the “Process Information window” or in report files.

UVCURINGTIME	The UV curing time set by either “UV_SET_CURING_TIME” or GUI panel
TASKDATE	Current time with the format of (MM/dd/yyyy hh:mm:ss) This value is updated by the system all the time
BARCODE	Barcode reading value obtained from image processing. “V_DECODE_BARCODE” command updates the value
CHIPID	Chip ID obtained from the barcode reading. “V_DECODE_BARCODE” command updates the value
WAFERID	Wafer ID obtained from “Process Information”
FABID	FAB ID obtained from “Process Information”

- **CHECK_LETTERCOUNT:**

For “MAKE_REPORT, MAKE_CUSTOM_REPORT and CHECK_PROCESSINFO” commands, if one of the commands is performed and the number of the given string is different from the set value, then returns failure and informs of the item with wrong number of characters to user.

5.1.2. Sequence file

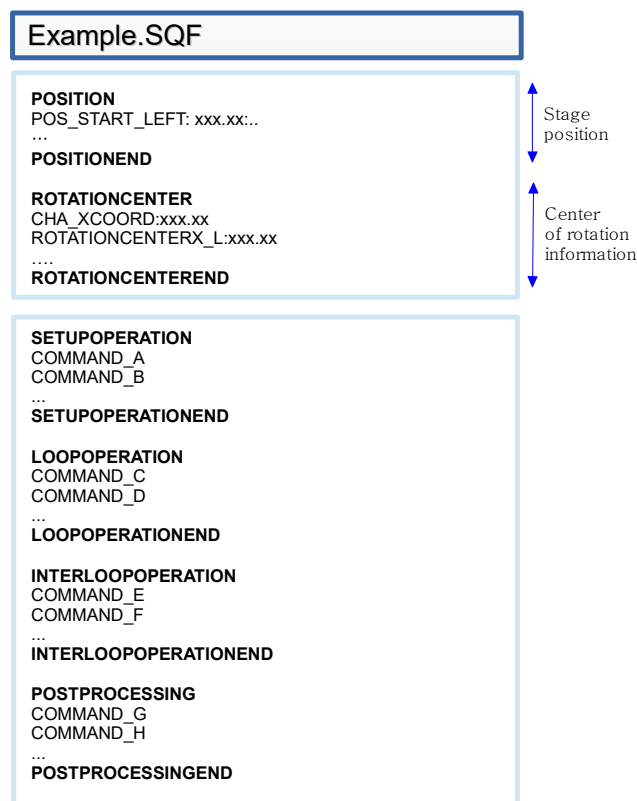


Fig. 5.2: Sequence file example

- FILENAME & LOCATION: Any
- The sequence file consists of the five sections which are:
 - POSITION
 - ROTATIONCENTER
 - SETUPOPERATION
 - LOOPOPERATION
 - INTERLOOPOPERATION
 - POSTPROCESSING
- Also the sections are located between “SectionName” and “SectionNameEND” keywords.
- The descriptions of the sections are as shown below:
 - POSITION
 - The position section is located between “POSITION” and “POSITIONEND” keywords
 - The information saved in the section is:
 - “Start position” of chip
 - Tool positions for vision processing, UV curing
 - ROTATIONCENTER
 - The section is located between “ROTATIONCENTER” and “ROTATIONCENTEREND”
 - The information saved in the section is center of rotation of the stage assembly in ThZ direction for the alignment.
 - The calculation is described and supported in “Calculate_rotation_center_Rev.02.html”.
 - SETUPOPERATION(Previous “CONFIG”)
 - The section is located between “SETUPOPERATION” and “SETUPOPERATIONEND” keywords
 - The section contains a series of commands for the setup process. Also “Number of repetition” is defined in this section for the loop operation.
 - LOOPOPERATION(Previous “EXECUTION”)

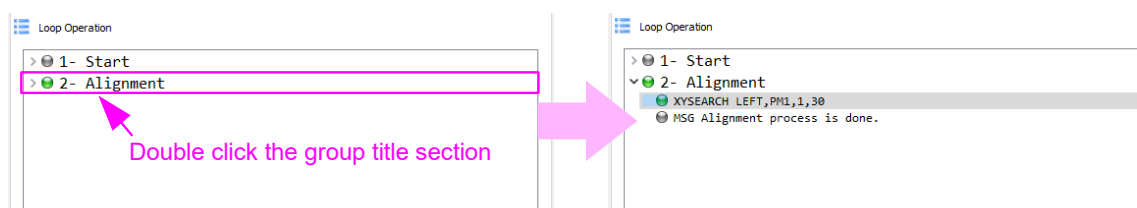
- The section is located between “LOOPOPERATION” and “LOOPOPERATIONEND” keywords
 - The section contains a series of commands for repetition process.
- INTERLOOPOPERATION
 - The section is located between “INTERLOOPOPERATION” and “INTERLOOPOPERATIONEND” keywords
 - The section contains a series of commands executed between the loop operation.
- POSTPROCESSING
 - The section is located between “POSTPROCESSING” and “POSTPROCESSINGEND” keywords.
 - The section contains a series of commands after the loop operation.
- The contents of sections (“SETUPOPERATION”, “LOOPOPERATION”, “INTERLOOPOPERATION”, “POSTPROCESSING”) are shown in the sequence manager window when the file is loaded.
- Expanding-folding command list
 - This function makes the sub-groups including several commands, and shows/hides the contents in the sub-groups by expanding and collapsing vertically in the sequence processing window.
 - A user can make sub-group with the following method:
 1. First write a group title(e.g. “2-Alignment”) as an independent line.
 2. Then add a command with a TAB at the beginning of the line.

```

LOOPOPERATION
1- Start
    MSG Start
2- Alignment
    XYSEARCH LEFT,PM1,1,50
    MSG Alignment process is done.
LOOPOPERATIONEND

```

<Example sequence file>



<Sequence processing window>

Fig. 5.3: Expanding-folding command list

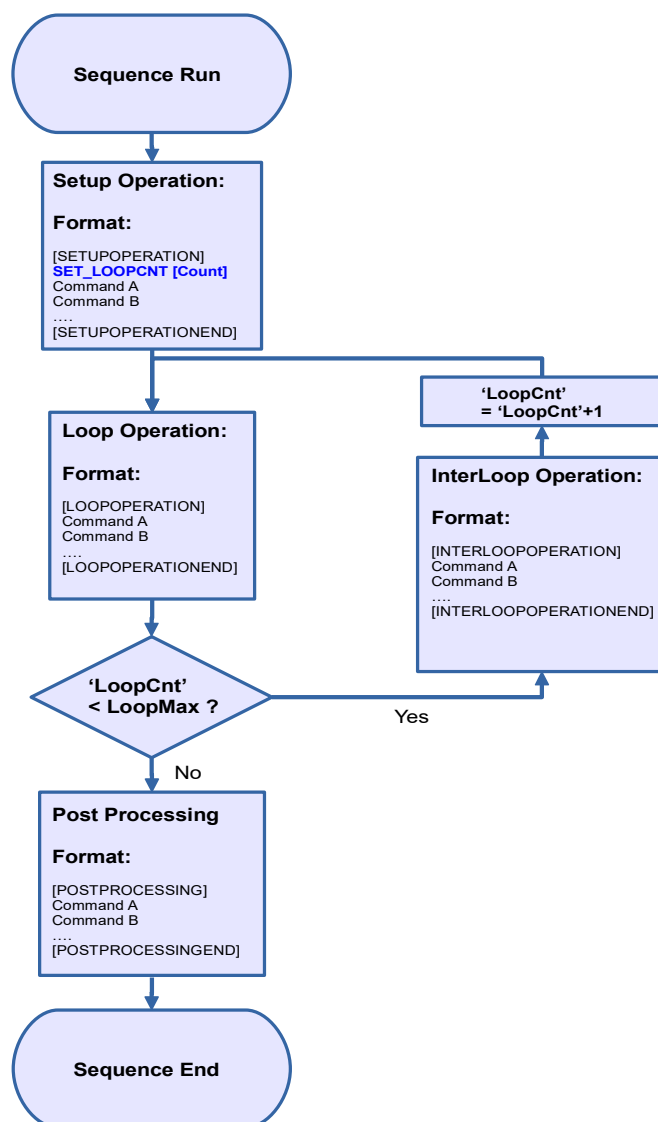
(a) LOOP OPERATION

Fig. 5.4: Block diagram of the loop operation

The sequence manager provides the function to repeat the command(s) in the “LOOPOPERATION” section of a sequence file by certain times.

The function is activated when the user run the sequence file including “SET_LOOPCNT” command in the SETUPOPERATION section.

The number of repetition(LOOPCNT) is defined the SET_LOOPCNT command. (see the programming manual)

The number of repetition is reset when the sequence is run.

- The user can see how many times repeats the sequence in the sequence monitoring window.

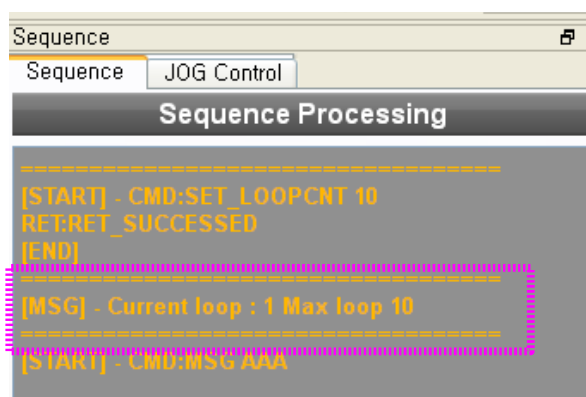


Fig. 5.5: Display the number of repetition

- “The maximum number of repeat loop” and “The number of loop repetition executed from the start” are saved specific system variables(“LOOPMAX”, “LOOPCNT”, see the programming manual)

(b) INCLUDE

- The function is activated when the user loads the sequence file including next conditions:
 1. The loaded sequence file which has no specific section.
 2. The loaded sequence file which has the “INCLUDE” section as shown below:

```
INCLUDE
FILENAME: "filename(full path)"
INCLUDEEND
```

Fig. 5.6: The format of "INCLUDE" section

3. The file has the information of the specific section which doesn't exist in the loaded sequence file.
- Specifications
 1. The information of the sections is loaded ONLY from one sequence file.
 2. The “POSITION” information loaded by “INCLUDE” function can be saved in the loaded sequence file with clicking “SAVE POSITION” button.
 - Example

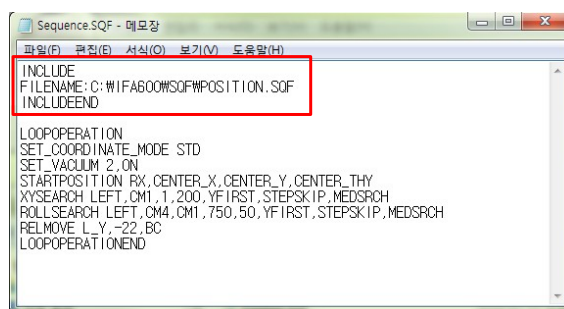


Fig. 5.7: Sequence file

1. Load the sequence file(Sequence.SQF) which has “INCLUDE” section and “LOOPOPERATION” section.

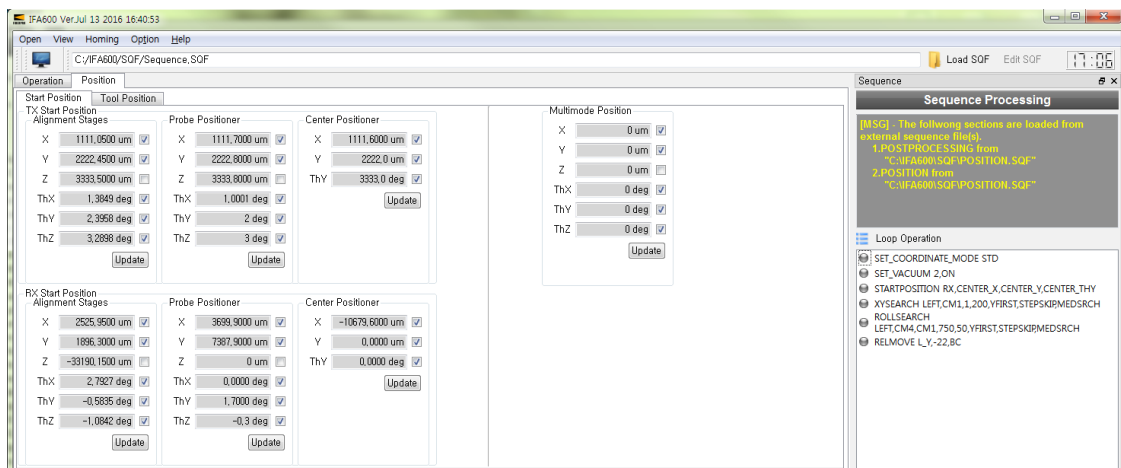


Fig. 5.8: Displayed message on the "Sequence Processing" window

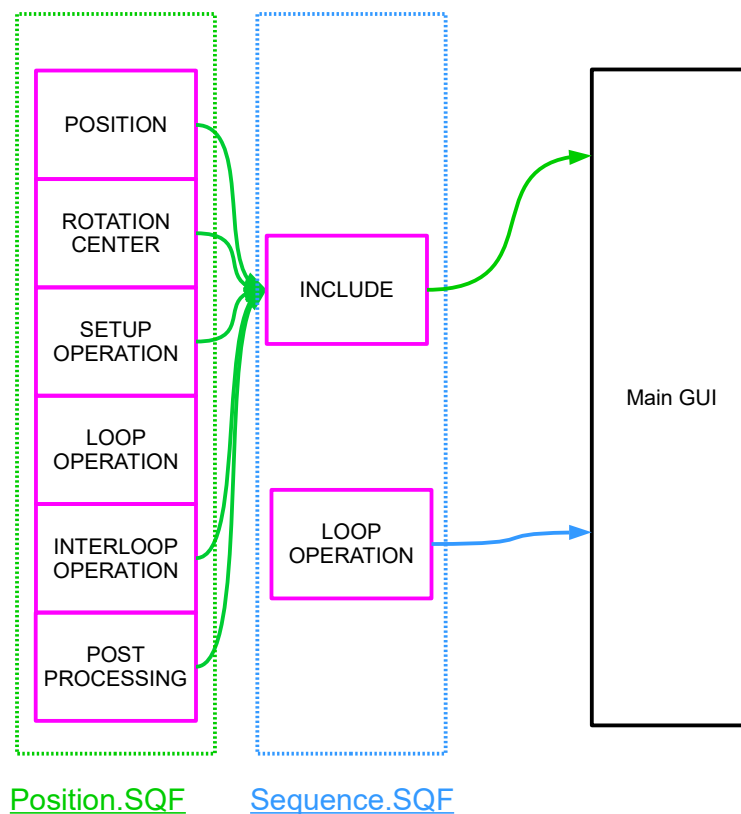


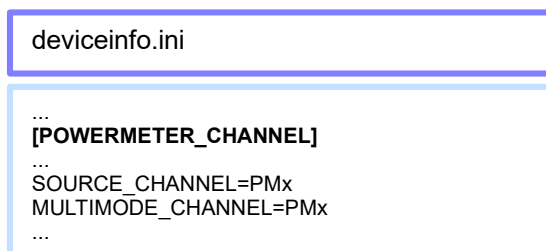
Fig. 5.9: Structure of the sequence with the "INCLUDE" function

- Then the GUI loads the information of "POSITION", "SETUPOPERATION", "INTERLOOPEPERATION", and "POSTPROCESSING" section from "POSITION.SQF"(Path: C:\IFA-600-AUCME3\SQF\) with messages notifying the sections are loaded from which sequence file "Sequee Processing" window.

* The position information loaded from "Position.SQF" file is saved in the "Sequence.SQF" with clicking the "SAVE POSITION" button.

5.1.3. Port setting “S.P”, “M.P”

- FILENAME: Deviceinfo.ini
- LOCATION: ./System/Device/

A diagram showing the structure of the Deviceinfo.ini file. It consists of two rectangular boxes. The top box is labeled 'deviceinfo.ini' and has a purple border. The bottom box is light blue and contains the following text: '[POWERMETER_CHANNEL]', '...', 'SOURCE_CHANNEL=PMx', 'MULTIMODE_CHANNEL=PMx', and '...'.

```
deviceinfo.ini

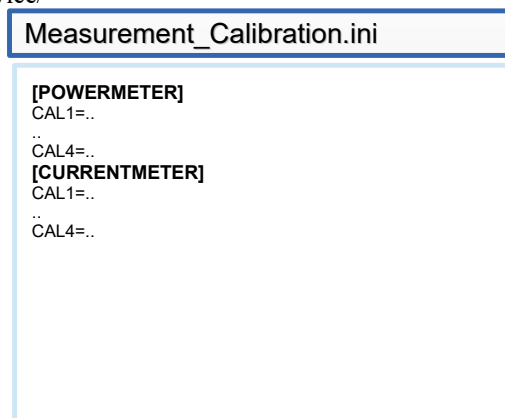
...
[POWERMETER_CHANNEL]
...
SOURCE_CHANNEL=PMx
MULTIMODE_CHANNEL=PMx
...
```

Fig. 5.10: Deviceinfo.ini

- SOURCE_CHANNEL: Powermeter channel to show “S.P” section on GUI (PMx, x=1,2,3...)
- MULTIMODE_CHANNEL: Powermeter channel to show “M.P” section on GUI (PMx, x=1,2,3...)

5.1.4. Calibration data file

- FILENAME: Measurement_Calibration.ini
- LOCATION: ./System/Device/

A diagram showing the structure of the Measurement_Calibration.ini file. It consists of two rectangular boxes. The top box is labeled 'Measurement_Calibration.ini' and has a blue border. The bottom box is light blue and contains the following text: '[POWERMETER]', 'CAL1=..', '..', 'CAL4=..', '[CURRENTMETER]', 'CAL1=..', '..', 'CAL4=..'.

```
Measurement_Calibration.ini

[POWERMETER]
CAL1=..
..
CAL4=..
[CURRENTMETER]
CAL1=..
..
CAL4=..
```

Fig. 5.11: Measurement_Calibration.ini

- The calibration data of power/current readings is saved in this file, which is shown in the data monitoring window in the GUI. The data for the optical power readings are found in the section [POWERMETER], [CURRENTMETER].

5.1.5. Stage information file

- FILENAME: mc_stageinfo_table.ini
- LOCATION: ./System/MC/

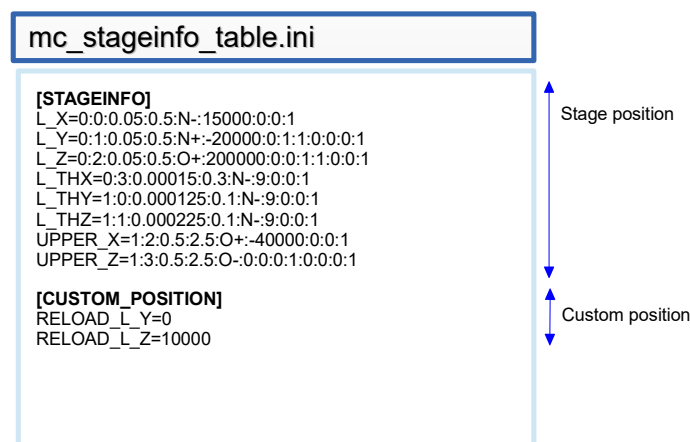


Fig. 5.12: Stage information file

This file includes information on all the stages used in the system. The file consists of two sections. The first section “STAGEINFO” includes the connection between the specified stage and the motion controller, stage setting information, etc. The second section “CUSTOM_POSITION” includes the absolute position the stages move to when “RELOAD” command is executed.

[WARNING]: Changing this file with inappropriate values can cause critical problems such as stage collisions. Consult Fiberpro before changing this file.

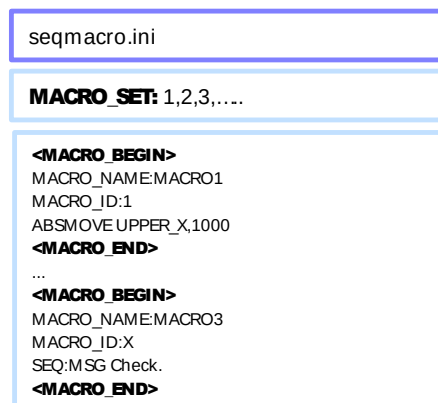
- “STAGEINFO”
 - Format – [StageName]=[BoardNum]:[AxisportNum]:[DistancePerPulse]:[Backoff]:[OriginType]:[OriginOffset]:[HomePosition]:[Reverse]:[Usable]:[Mirror]:[LowLimit]:[HighLimit]
 - Items -
 - [StageName]: the name of the corresponding stage.
 - Support stage name: Check the programming manual
 - *The name of the custom stage must be in the format “CUSTOM_”.
 - [BoardNum]: The board number of the motion controller(SC7000) the stage is connected to(For IFA-600-AUCME3, this number is 0 to 6.
 - *The custom stage is assigned to number 4.
 - [AxisportNum]: The port number of the motion controller board which the stage is connected to. It ranges 0 ~ 3.
 - [DistancePerPulse]: distance in um or angle in degree which the stage moves by one control pulse
 - [Backoff]: distance for backlash compensation
 - [OriginType]: Homing type(Origin type). There are four different types O+,O-, L+, L-
 - “O+” = homing with origin sensor. moving direction is “+”
 - “O-” = homing with origin sensor. moving direction is “-”
 - “L+” = homing with “+” Limit sensor.
 - “L-” = homing with “-” Limit sensor.
 - [OriginOffset]: distance to move before start homing process.
 - [HomePosition]: coordinate to move after homing process finishes.
 - [Reverse]: flip logical moving direction (FORWARD, BACKWARD direction is switched when this value is 1. (0: not reverse, 1: reverse)
 - [Usable]: set the stage is usable or not (0: not use, 1: use)
 - [Mirror]: flip sign of the absolute position. (0: not mirror, 1: mirror)
 - [LowLimit]: lower software limit position.
 - [HighLimit]: higher software limit position.
- “CUSTOM_POSITION”
 - Format – [RELOAD_STAGE_NAME]=[ABS Position]

- Items –
 - [RELOAD_STAGE_NAME]: stage name to be used by Reload command
 - Supported stage: RELOAD_L_Y, RELOAD_L_Z
 - [ABS Position]: absolute coordinate to move
- “CUSTOM_STAGE”

Users can add up to 4 motorized stages in IFA-600-AUCME3. STAGEINFO and CUSTOM_STAGE must be set to add the custom stage to the IFA-600-AUCME3 program.

 - [USE]: Whether to use the custom stage (Y/N)

5.1.6. Macro file



The diagram illustrates the structure of the seqmacro.ini file. It is divided into three sections by colored boxes: a purple box for the filename 'seqmacro.ini', a light blue box for the 'MACRO_SET' configuration, and a larger light blue box for the macro definitions. The 'MACRO_SET' section contains the text 'MACRO_SET: 1,2,3,...'. The macro definitions section contains two macro blocks. Each block starts with '<MACRO_BEGIN>', followed by 'MACRO_NAME:MACRO1' and 'MACRO_ID:1' for the first macro, and 'ABS MOVE UPPER_X,1000'. The second macro block starts with '<MACRO_BEGIN>', followed by 'MACRO_NAME:MACRO3', 'MACRO_ID:X', and 'SEQ:MSG Check.'. Each macro block ends with '<MACRO_END>'. Ellipses '...' are used between the two macro blocks to indicate they are not exhaustive.

```
seqmacro.ini

MACRO_SET: 1,2,3,...

<MACRO_BEGIN>
MACRO_NAME:MACRO1
MACRO_ID:1
ABS MOVE UPPER_X,1000
<MACRO_END>
...
<MACRO_BEGIN>
MACRO_NAME:MACRO3
MACRO_ID:X
SEQ:MSG Check.
<MACRO_END>
```

Fig. 5.13: seqmacro.ini

- FILENAME: Any
- LOCATION: ./System/Device/
- The software supports macro function that a user can prepare a series of commands and run with a simple command or through UI.

The software loads “seqmacro.ini” file to set basic macro functions. The user also can load a macro file by using the “LoadMacro” button of “Macro Button” tab in the control panel window.

- “MACRO_SET”
 - Assign the macros to the macro buttons in the GUI
 - “MACRO_SET:4,5” assigns the macro 4 to the button 1, macro 5 to the button 2
- Defining a macro
 - A macro is defined between <MACRO_BEGIN> and <MACRO_END> keywords
 - A macro starts with “MACRO_NAME” and “MACRO_ID”. The “MACRO_ID” number and the “MACRO_NAME” value must be unique through a file.
 - MACRO_ID: Unique number of a macro
 - MACRO_NAME: A unique name of a macro. it is also shown in the corresponding macro button.
 - A series of commands follows the MACRO_ID.

5.1.7. Macro button

- FILENAME: ViewInfo.ini
- LOCATION: ./System/Device/

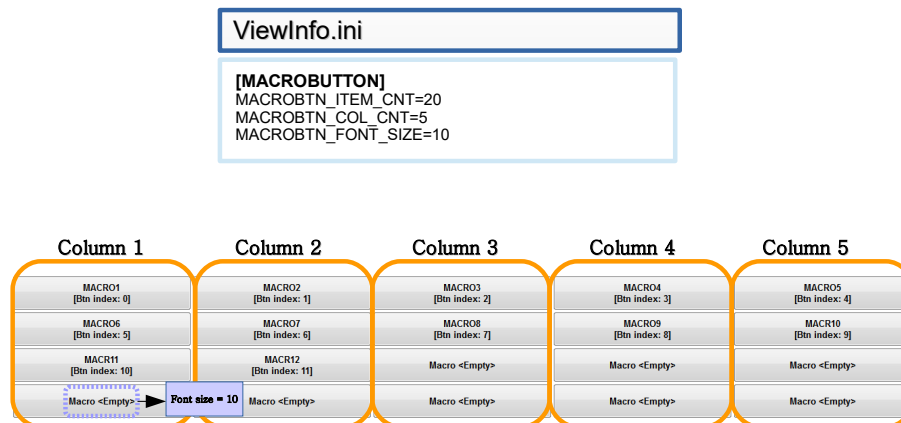


Fig. 5.14: Macro buttons

The section “[MACROBUTTON]” in the file is for the configuration of the macro buttons in the “Macro buttons” tab. In this section, the number of main buttons, button layout, and text size on the buttons can be set. The modification of the settings in this section becomes effective after restart of the program.

- MACROBTN_ITEM_CNT: The number of macro buttons displayed on the window. (Default = 12)
- MACROBTN_COL_CNT: The number of columns for the button layout. (Default =4)
- MACROBTN_ROW_CNT: The number of columns for the button layout. (Default =3)
- MACROBTN_FONT_SIZE: The font size of the text shown in the macro buttons.(Default = 13)

5.1.8. Jog controller settings

- FILENAME: ViewInfo.ini
- LOCATION: ./System/Device

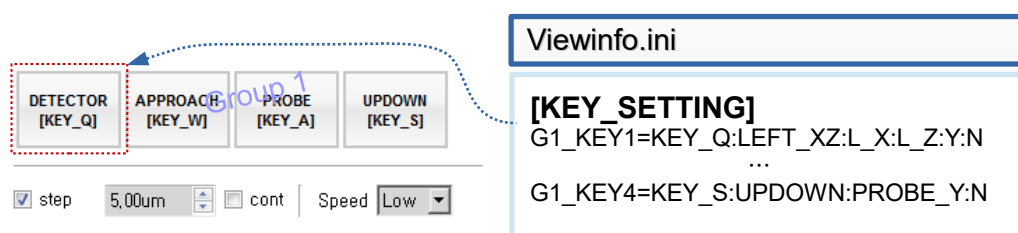


Fig. 5.15: File: ViewInfo.ini

This section is used to assign hot-keys on the jog controller for the control of stages or pneumatic devices. Details on the jog controller are described in the section 5.3.7 in the operation manual. An example of the section is shown below.

```
[KEY_SETTING]
G1_KEY1=KEY_Q:LEFT:L_Y,L_X:N:Y
G1_KEY2=KEY_W:APPROACH:LEFT_Z:Y
G1_KEY3=KEY_A:U_FRONTBACKWARD:UPPER_X:Y
G1_KEY4=KEY_S:U_LEFTRIGHT:UPPER_Z:N
```

Fig. 5.16: Jog controller setting example

[Group 1 panel indicators]

The group 1 indicators are linked to hotkeys for the stage control, and maximum 12 hot keys can be assigned in the group. For each hot key, maximum 2 stages can be controlled.

The format to set the hot key for one stage control is as following:

Format:
<code>G1_KEY(n)=[Key name]:[Control title]:[Stage name]:[Reverse stage]</code>
Parameters:
• G1_KEY(n): Hot key item ID. n=1~12 (examples: G1_KEY4) • Key name: The name of key to be used in the format of "KEY_[A-Z0-9]". (examples: KEY_S, KEY_9) • Control title: The name of the control to be displaced on the indicator • Stage name: The name of the stage to be controlled by up/down arrow keys • Reverse stage(Y/N): When "Y" flag is set, the upward arrow key moves the specified stage in the positive direction. With "N" flag, the moving direction is reversed.
Example:
<code>G1_KEY4=KEY_S:U_LEFTRIGHT:UPPER_Z:N</code>

The format to set the hot key for dual stage control is as following:

Format:
<code>G1_KEY(n)=[Key name]:[Control title]:[Stage1 name]:[Stage2 name]:[Reverse Stage1]:[Reverse Stage2]</code>
Parameters:
• G1_KEY(n): Hot key item ID. n=1~12 (examples: G1_KEY1) • Key name: The name of key to be used in the format of "KEY_[A-Z0-9]". (examples: KEY_S, KEY_9) • Control title: The name of the control to be displaced on the indicator • Stage1 name: The name of the stage to be controlled by up/down arrow keys • Stage2 name: The name of the stage to be controlled by left/right arrow keys • Reverse1 stage(Y/N): When "Y" flag is set, the upward arrow key moves the specified stage in the positive direction. With "N" flag, the moving direction is reversed. • Reverse2 stage(Y/N): When "Y" flag is set, the right arrow key moves the specified stage in the positive direction. With "N" flag, the moving direction is reversed.
Example:
<code>G1_KEY1=KEY_Q:LEFT:L_Y,L_X:N:Y</code>

5.1.9. GUI Lock

FILENAME: ViewInfo.ini

LOCATION: ./System/Device/

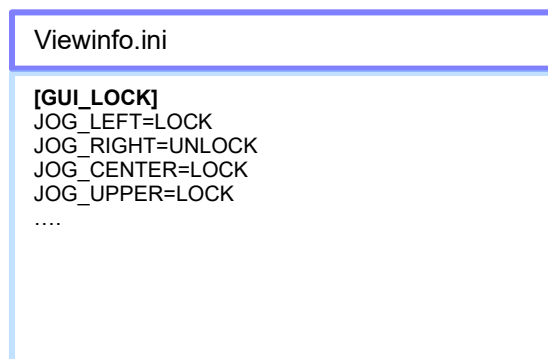


Fig. 5.17: GUI Lock setting

Some of control panels in the GUI window can be locked/unlocked by either a setup file or “SET_GUI_LOCK” command. This function is useful to avoid the possible damage caused by manipulating wrong control panels. The control panels that can be locked/unlocked are shown in the figure below.



Fig. 5.18: GUI Lock item #1

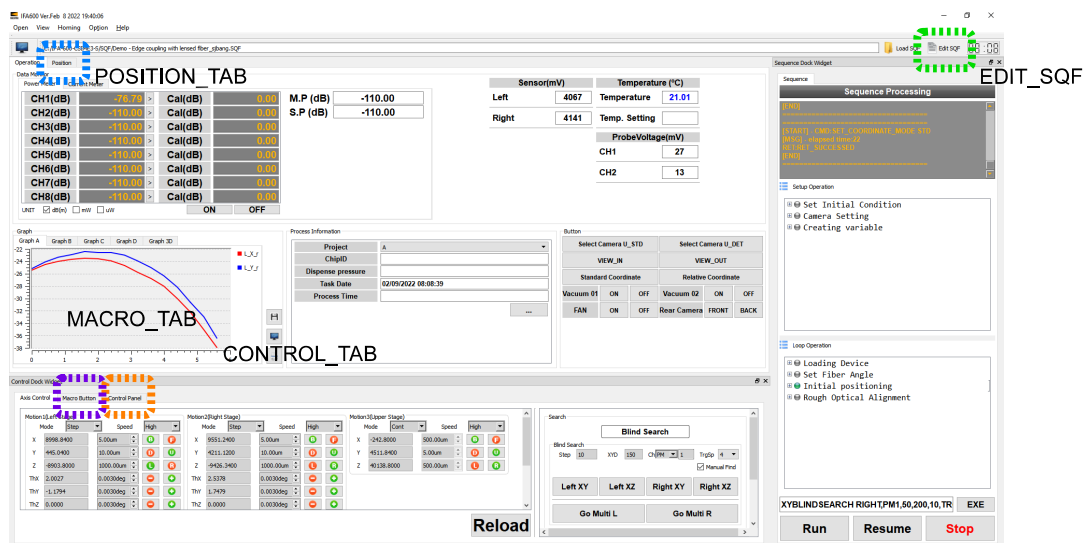


Fig. 5.19: GUI Lock item #2

- JOG_LEFT: Control panel for the left stage assembly
- JOG_RIGHT: Control panel for the right stage assembly
- JOG_S1: Control panel for the s1 stage assembly
- JOG_UPPER: Control panel for the upper stage assy(tool positioner)
- JOG_CENTER: Control panel for the center stage assy
- STAGE_ALL: Control panel for all stage assemblies
- SEARCH_LEFT: Control panel for the L Search
- SEARCH_RIGHT: Control panel for the R Search
- MULTIMODE: Control panel for the special position control of the multimode fiber installed on the left, right assembly
- POSITION_TAB: The tab to allows for a user to set/update/save/load sets of stage positions for predefined actions. The information of the position sets is included a sequence file.
- CONTROL_TAB: The tab which has the predefined tool buttons
- MACRO_TAB: The tab which has the macro buttons
- EDIT_SQF: The button to open the sequence file currently loaded

5.1.10. Danger zone

FILENAME: ViewInfo.ini

LOCATION: ./System/Device/

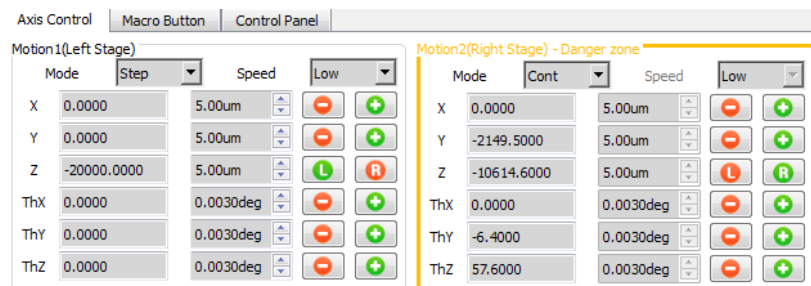


Fig. 5.20: Danger zone

The user can set the area which should ONLY control the stage moving with “Low” speed. We call it “Danger zone”, and it is made for avoiding collision by user inattention. It ONLY works when the user controls stage moving with the “Axis Control”. If some stage is in the “Danger zone”, the yellow box will be displayed around the motion group box joined the stages.

“Danger zone” can be set as following:

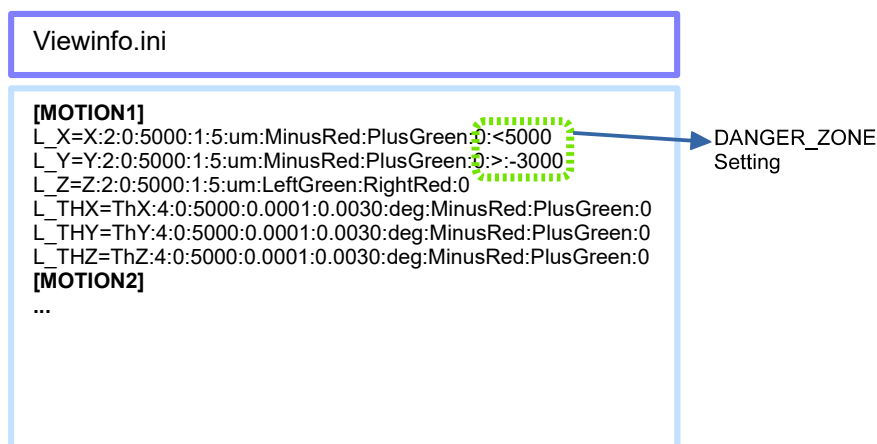


Fig. 5.21: Setting “Viewinfo.ini” for the danger zone

- Syntax of the stage control panel:
 [Stage Name]=[View title][Decimal place]:[Min]:[Max]:[Step size]:[Default]:[Unit]:[IconType1]:[IconType2]:
 [Reverse]:[Sign of inequality<Danger zone>]:[Reference <Danger zone>]
- 1. Open “Viewinfo.ini”(Path: ./IFA600/Device).
- 2. Set the danger zone by editing [Inequality sign of “Danger zone”] and [Reference of “Danger zone”].
 - The danger zone is defined by combination of [Inequality sign of “Danger zone”] and [Reference of “Danger zone”]. (* If last two sections of the syntax are empty, the danger zone doesn't be set.)
 - If the user set the “Danger zone” on the several stages, the overlapping area will be “Danger zone”.
 - [Inequality sign of “Danger zone”]: Braket(Allowed ONLY “>”, “<”).
 - [Reference of “Danger zone”]: A real number.
 - Case 1.
 R_Y=Y:2:0:5000:1:5:um:MinusRed:PlusGreen:0:<:-2000
 → “Danger zone” is where the height of right stage assy. is lower than -2000 um.
 - Case 2.
 R_Y=Y:2:0:5000:1:5:um:MinusRed:PlusGreen:0:<:-2000
 R_Z=Z:2:0:5000:1:5:um:RightGreen:LeftRed:0:>:-10000
 → “Danger zone” is where the height of right assy. is lower than -2000 um and R_Z value is higher than -10000 um.

5.2. FIBERPRO Limited Warranty

FIBERPRO

Limited WARRANTY

FIBERPRO warrants its products to be in conformance to mutually agreed upon written specifications and to be free from defects in material and workmanship. **FIBERPRO**'s product warranty provides a warranty on all products for a period of 12 months. The above warranty period shall begin on the date of shipment by **FIBERPRO** to the Buyer or, if the Buyer is an authorized retailer of such **FIBERPRO** products; from the date of shipment by the retailer to the retailer's original customer. The warranty shall apply solely to products that have been shown to function properly under instructions specified within the user manual, but malfunction due to defect with part(s) installed within the instrument or defect with the instrument itself. The warranty shall not apply to product failure or deficiency caused by misuse, abnormal or excessive use, neglect, alteration, improper installation, unauthorized repair or modification, improper testing, or failure to follow instructions on the user's part. The warranty shall only apply to products and software provided by **FIBERPRO**, and shall not apply to services provided by **FIBERPRO**. In the case of buying **FIBERPRO**'s products through a separate company, **FIBERPRO**'s warranty shall not apply to the pertaining product and **FIBERPRO** shall have no liability in compensation for damages in the case where the product is beyond repair. Replaced products and parts shall become the property of **FIBERPRO**. This warranty applies to all products that the buyer purchases from **FIBERPRO**, except for cases where a separate warranty agreement is made on a product, which will then have priority. Cables and other accessories shall not apply in the product warranty.

During the warranty period, in the case of an instrument malfunction during normal use due to defect with parts that affect the quality or performance of the instrument, the buyer can receive free of charge repair service. Also, if during normal use, the same part that **FIBERPRO**'s engineer has repaired is damaged again, the buyer may receive free of charge repair for that part within the next 12 months (this period starts on the date the part has been repaired by **FIBERPRO**).

The buyer may receive Charged Repair Service in the following cases. a) In the case where the warranty period has expired, b) In the case of malfunction due to unauthorized alteration/modification on the user's part, c) In the case where the buyer removes a "DO NOT OPEN" sticker and dismantles or opens up the instrument, regardless of the warranty period, d) In the case where after initial installation, a malfunction occurs when the buyer attempts to connect an additional product or attempts to reconnect the same product, e) In the case where a third party company's product has been purchased and the buyer requests installation, f) In the case where reinstallation is required due to the buyer moving the product, g) In the case where malfunction occurs due to a third party company's product (including softwares), h) In the case where malfunction occurs due to the buyer's negligence of the warning and caution items described in the user manual, i) In the case where malfunction occurs due to user's misuse of electrical capacity, j) In the case where a malfunction occurs due to usable or optional parts not designated by **FIBERPRO**, k) In the case where malfunction occurs due to repairs made by engineers other than those from **FIBERPRO**, l) In the case where malfunction occurs due to external shock or fall, m) In the case where malfunction occurs due to natural calamities (lightning, fire, earthquake, storm/flood, etc.), n) In the case where the lifecycle of a usable part has ended.

FIBERPRO's Service shall be handled by **FIBERPRO**'s technical support team, and user repair or replacement orders for a product must be filed to **FIBERPRO**'s Quality Control Team. In the case where the buyer wishes for an on-site service, buyer will be charged the travel fee according to **FIBERPRO**'s conditions and will then be able to receive on-site service.

In-house Service takes the form of the buyer sending in the product to **FIBERPRO** for repair. If the product is under warranty and the malfunction is due to a defect with the product, repair will be provided free of charge. In the case of paid repair, the fee will follow **FIBERPRO**'s fee structure.

For all cases including: when the user sends a defective product under warranty for in-house repair free of charge, when

the user sends a user-caused defective product under warranty for in-house repair for charged repair, and for all other in-house charged repair service, the user and **FIBERPRO** will share the burden of shipping once each.

In the case of a product defect within the warranty period, On-site service will be provided free of charge. In the case of paid repair, parts fee, repair fee, travel fee, and examination fee will be charged according to **FIBERPRO**'s fee structure.

However, even in the case that the product is still under its warranty period, if the malfunction is due to misuse or other reasons outlined above under paid repair, then a fee will be charged according to paid service fee structure.

If the product malfunctions within one month of purchase, it will be considered DOA (Dead On Arrival) and will be given top priority in warranty service repair.

FIBERPRO's warranty as set forth above shall not be enlarged, diminished or affected by, and no liability shall arise out of, **FIBERPRO**'s rendering of technical advice in connection with the buyer's order. The warranty set forth above is not assignable by the buyer. The buyer's sole remedy under the above warranty shall be repair or replacement as provided above.

FIBERPRO's sole and exclusive maximum liability for any claim by the buyer, or anyone claiming through or on behalf of the buyer arising out of the buyer's order for the above warranty, shall not in any event exceed the actual amount paid by the buyer to **FIBERPRO** for the product.

In no event shall **FIBERPRO** be liable for any indirect, incidental, collateral, exemplary, consequential or special damages or losses arising out of the buyer's order of products delivered under the above warranty, including, without limitation, loss of use, profits, or goodwill. Some countries do not allow the exclusion of limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to the buyer.

EXCEPT AS EXPRESSLY STATED ABOVE IN THIS AGREEMENT, THE FOREGOING WARRANTIES ARE IN LIEU OF ALL OTHER CONDITIONS OR WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, AN IMPLIED CONDITION OR WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND OF ANY OTHER WARRANTY OBLIGATION ON THE PART OF **FIBERPRO**.Rev01.

[END OF DOCUMENT]