BXC-CBB

Command reference manual

Version 2

Before using this document

- **The content described in this document is subject to change without notice for the purpose of system improvement.**
- **It is prohibited to copy a part or all of the content described in this document without permission except for personal use. It is, also, strictly prohibited to reprint without prior notice.**
- **Please note that we shall not be responsible for any damage or lost profit caused by using this document or any claim from third parties.**

Table of contents

1. INTRODUCTION

CBB is the firmware of the control box BXC-CBB. This document describes the external specifications of CBB.

1.1. Rule of Notation

This document complies with the following notations.

Table 1 Rule of Notations

Notation	Example	Description
All text	**Commands cannot be received.**	Underline or **Block** , or both of **Underline and Block** are used to emphasize or to call attention to the meaning of the text.
All text	*number*	If representing the element in the special group, those terms are shown in *Italics.* *Number* represents the element in the numeral group. When the numeral group consists of 1, 2, 3, 4, 5, and 6, *number* describes one of those.
	command	*command* represents the element in Command group.
Area of variables	(0 - 1000)	The area between 0 to 1000. 0 <= x <= 1000
	(ON, OFF, STANDBY)	Any one of ON, OFF and STANDBY.
Radix of value	0x01, 0x55, 0xAA	prefix 0x shows hexadecimal.
	0b01, 0b0101, 0b10101010	prefix 0b shows binary.
Units	[0.1%]	Multiplying the set value by 0.1 indicates the actual percentage. Example on the left shows that the set value 2 equals to 0.2%.
	[um]	The unit of the set value is um (micro meter).
	[s]	The unit of the set value is s (second).
Command format	EBNF(***E****xtended* ***B****ackus-****N****aur* ***F****orm*)	
	α := *β*	α is defined as β.
	α | *β*	Select α or β
	[*α*]	α or nothing
	{ *α* }	Repeat α for 0 times or more.
	<< *α* >>	Repeat α once or more
	(*α*)	α
	α - *β*	A value which is no less than α and no more than β. Both.α and β must be integers. α <= x <= β
	Identifier := *Alphabet* { *Alphabet* | Numeral }	An identifier is defined as a string beginning with an alphabet, after which 0 or more alphabets and/or numeric characters follow.
	Repeated abbreviated notation … *Foo p1*,…	Repeats the last sorted data more than once. *Foo p1*,*p1* *Foo p1*,*p1*,*p1* *Foo p1*,*p1*,*p1*,*p1* are shown.

2. System overview

2.1. Configuration of the System

CBB system

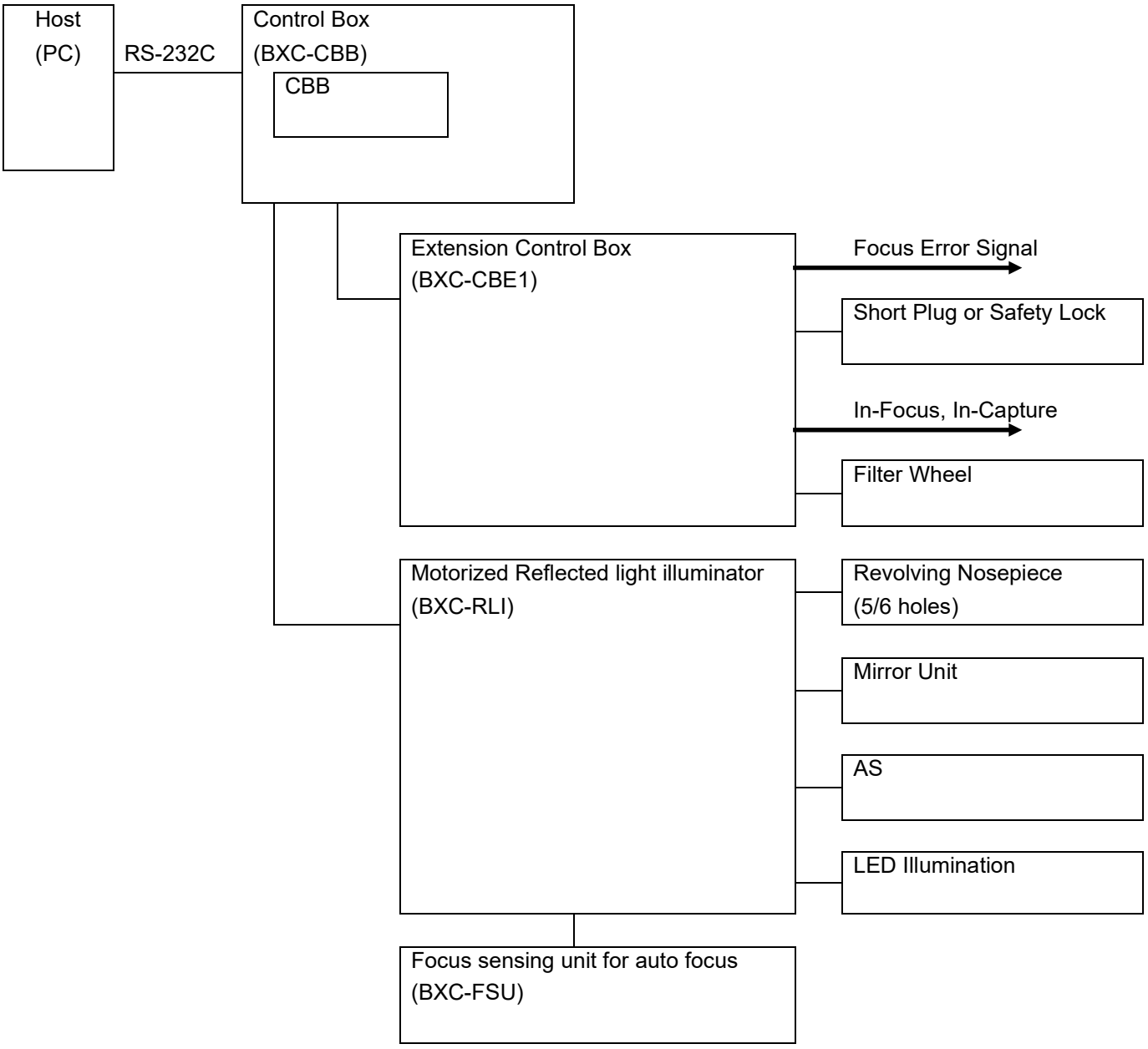


Fig 1 CBB system

3. I/F

BXC-CBB has the following I/F.

- DIP-SW
- Indicator LED
- Buzzer
- Host I/F
  - RS-232C

BXC-CBE1 has the following I/F.

- DIP-SW
- Indicator LED

3.1. Dip Switch

3.1.1. **BXC-CBB**

Refer to the instructions.

3.1.2. **BXC-CBE1**

Refer to the instructions.

3.2. Host I/F

CBB is controlled by commands from Host. I/F uses the serial UART communication via RS-232C. In the further descriptions, **responses/replies of CBB for commands from** Host **and notifications from CBB are expressed as Commands.**

3.2.1. Basic Actions of the Commands

3.2.1.1. Port Setting

Refer to the instructions.

3.2.2. **Command Format**

The following table shows the command format. The command character set uses ASCII codes and commands are expressed in EBNF (***E**xtended **B**ackus-**N**aur **F**orm*) notation. **The maximum length of a command is 64[B] (including a terminator).**

command* := *index tag* [*tag-delimiter data* { *data-delimiter data* }] *terminator

Table 2 Command components

Command component	Name	Description	code
index	Index	Selects the target system or unit. A numeric character.	CBB is 1.(Fixed)
tag	Tag	Classifies the application. string: uppercase alphabets and graphic characters.	Combination in variable length of 'A'~'Z' and '?' e.g. 'L', 'OB', 'U?', etc.
tag-delimiter	Tag delimiter	Delimiter between ***tag*** and ***data.*** A graphic character.	' ' Space(0x20)
data	Data	Data A graphic character, a string of numeric characters or a string of uppercase alphabet characters.	Combination in variable length of '+', '!', '-', '.', ':', '_', '0'~'9', 'A'~'Z'.~
data-delimiter	Data delimiter	Delimiter between ***data*** and ***data.*** A graphic character.	',' Comma (0x2C)
terminator	Terminator	Terminates ***command.*** 2 control character	CR+LF (0x0D 0x0A)

The command is a string terminated by control characters (CR+LF). **"*index*" and "*terminator*" are omitted in the further descriptions of commands.**

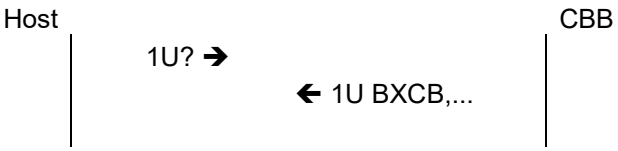
3.2.2.1. **Index and Target System**

Host must specify the target system(F/W) which is the transmission destination of commands by the index. The index of CBB is 1. **CBB ignores commands containing the index other than 1.**

Table 3 Index and Target System

Index	Target System	Description
1	CBB	

e.g.



3.2.2.2. **Command Response**

This section described the case that CBB ignores (discards) the received command. All commands of *FOO* which is sent from the host are ignored. The cases of below are limited to CBB. However, this is limited to the case that *FOO* index specifies CBB.

■Sequence

Host		CBB
s1. On initialization		Turn of the power Starts initialization
	nFOO ➔	On initialization (The command will be discarded.)
		Terminated initialization (The command can be accepted.)
s2. Numerous commands		
	nFOO1 ➔	When sending a command unilaterally before receiving a
	:	handshake command, the unit accepts up to 32 commands
	:	and ignores subsequent commands.
	nFOO32 ➔	The commands of *FOO1* to *FOO32* will be processed.
	nFOO33 ➔	The command of *FOO33* and later will be ignored.

The index of *FOO* specifies CBB, but it sends the invalid response if not interpretable.

■Sequence

Host		CBB
s1. Undefined command		
	nFOO ➔	*FOO* is not described in the command dictionary of the unit.
	➔ *n*x	Invalid (command) response will be sent.
s2. Overlong command		
	nFOO ➔	*FOO* exceeds the maximum command length.
	➔ *n*x	Invalid (command) response will be sent.

*) *n* represents the index.

3.2.2.3. **Command Sequence**

The commands sent by the host can be classified according to their purposes. The classified type determines whether handshake is used or not (whether the commands between the Host and CBB are synchronized or not).

Table 4 Handshake and Nest

Format*1	Type	Direction Host CBB	Description	Handshake
X parameters *X*	Control request <**R**>	➔	Request of control(action/setup)	Used
X + *X parameters*	Positive acknowledgement <PA>	➔	Normal completion of the control request	Used
X !,*error-code* *X* !,*error-code*,...	Negative acknowledgement <NA>	➔	Abnormal completion of the control request	Used
X?	Query <Q>	➔	Parameter/data query	Used
X parameters	Notification <N>	➔	Parameter/data notification	Used
X parameters	Active notification <AN>	➔	Parameter/data active notification	Not used
X error-code	Error notification <EN>	➔	Error notification	Not used

*1 *X* represents a tag, parameters represents parameters and *error-code* represents the error code.

3.2.2.3.1. Command with Handshake

A control request <R> handshakes with a positive acknowledgement <PA>/negative acknowledgement <NA> while a query <Q> handshakes with a notification <N>. Receiving a handshake command from CBB, the host computer is notified that a processing (an action) is completed.

■Sequence

Host		CBB
s1. Control Request	**<R>** →	Starts request processing
		Processing the request
Positive acknowledgement	← **<PA>**	Terminates request processing
s2. Control Request	**<R>** →	Starts request processing
		Processing the request
Negative acknowledgement	← **<NA>**	Terminates request processing
s3. Query	**<Q>** →	Starts querying
		On querying
Notification	← **<N>**	Terminates querying

3.2.2.3.2. Commands without Handshake

Handshake is not required with the active notification <AN> and error notification <EN>.

■Sequence

Host		CBB
s4. Active notification	*ENABLE***<R>** →	Active notification is enabled
Positive acknowledgement	← *ENABLE***<PA>**	Starts the active notification
Active notification	← **<AN>**	
Active notification	← **<AN>**	
s5. Error notification	← **<EN>**	Error occurs

*) *ENABLE* represents a command that enables active notification.

3.2.2.3.3. The Nest of Handshaking

The commands can be nested to a handshake command. The commands sent by the host computer are not received in the order they sent. An acknowledgement is sent to the host computer when a corresponding process is completed. However, some <R> commands cannot be nested.

■Sequence

Host				CBB
s6. Control request	*FOO*<**R**>	➔		Starts the processing requested by *FOO*.
	BAR<**R**>	➔		Starts the processing requested by *BAR*.
		←	*FOO*<**PA**>	Completes the processing requested by *FOO*.
		←	*BAR*<**PA**>	Completes the processing requested by *BAR*.
s7. Control request/Query	*FOO*<**R**>	➔		Starts the processing requested by *FOO*.
	BAR<**R**>	➔		Starts the processing requested by *BAR*.
	FOO2<**R**>	➔		Starts the processing requested by *FOO2*.
	BAR2<**R**>	➔		Starts the processing requested by *BAR2*.
	FOO<**Q**>	➔		Starts the query.
		←	*FOO*<**PA**>	Starts the processing requested by *FOO*.
		←	*FOO*<**N**>	Completes the query.
		←	*BAR2*<**PA**>	Completes the processing requested by *BAR2*.
		←	*FOO2*<**PA**>	Completes the processing requested by *FOO2*.
		←	*BAR*<**PA**>	Completes the processing requested by *BAR*.
s8. Control request (cannot be nest)	*FOO*<**R**>	➔		
	BAR<**R**>	➔		
		←	*BAR*<**NA**>	
		←	*FOO*<**PA**>	

* *FOOn* and *BARn* represent tags and *error-code* represents an error code.

3.2.3. **Command Reference**

3.2.3.1. **Normal Commands**

3.2.3.1.1. Login LOG LOG?

■Summary

1. Changes the mode among Remote / Local.
2. Gets the current mode.

■Comments

1. This command is for backward compatibility.
2. Only this command supports index 2.
3. The initial value is backed to the following "default" each time the power is turned on.

■Format

Command	Type	Direction	Description
LOG *p1*	**R**	Host ➔ CBB	Changes the mode among Remote / Local.
LOG +	**PA**	Host ← CBB	Succeeded.
LOG !,*error-code*	**NA**	Host ← CBB	Failed.
LOG?	**Q**	Host ➔ CBB	Queries the current mode.
LOG *p1*	**N**	Host ← CBB	Notifying the current mode.

■Parameters

p1	'OUT'	Local
	'IN'	Remote default

■Sequence

Host			CBB
1LOG IN	➔		Logs in (to Remote mode)
	←	1LOG +	
1LOG?	➔		
	←	1LOG IN	Remote mode.
2LOG IN	➔		Logs in (to Remote mode)
	←	2LOG +	
2LOG?	➔		
	←	2LOG IN	Remote mode.

3.2.3.1.2. Unit U? UNIT?

■Summary

1. Returns the unit name.

■Comments

1. The number of the parameters in <N> is the variable number of the strings that have been specified to represent the respective unit.

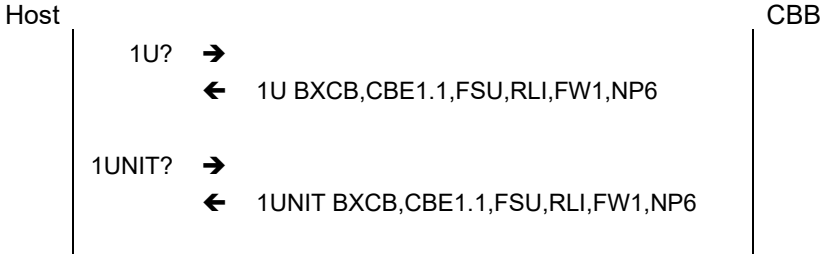
■Format

Command	Type	Direction	Description
U?	**Q**	Host ➔ CBB	Queries the unit name.
U *p1,...*	**N**	Host ➔ CBB	Notifies the unit name.
UNIT?	**Q**	Host ➔ CBB	Queries the unit name.
UNIT *p1,...*	**N**	Host ➔ CBB	Notifies the unit name.

■Parameters

	ID	Description
p1	BXCB	System ID
	CBE1.1	Extension control box
	FSU	Focus sensing unit for auto focus
	RLI	Motorized Reflected light illuminator
	FW1	Filter wheel 6 holes
	NP5	Motorized Revolving nosepiece 5 holes
	NP6	Motorized Revolving nosepiece 6 holes

■Sequence



3.2.3.1.3. Version V

■Summary

1. Gets the firmware version.

■Comments

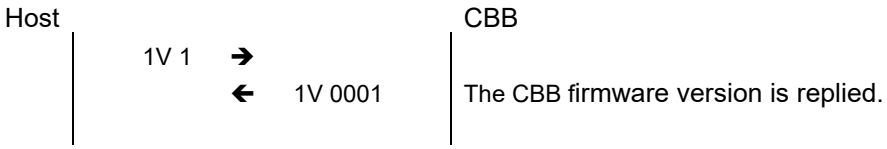
■Format

Command	Type	Direction	Description
V *p1*	**R**	Host ➔ CBB	Gets the firmware version.
V *p2*	**pA**	Host ➔ CBB	Firmware version is replied.
V !,*error-code*	**nA**	Host ➔ CBB	Failed.

■Parameters

p1	Strings	Part
	1	CBB
	2	FSU
	3	RLI
	4	CBE1.1
p2	(0001 - 9999)	Firmware version 4[B] fixed length
	----	Unknown 4[B] fixed length

■Sequence



3.2.3.1.5. LED Illumination IL IL?

■Summary

1. Controls the LED illumination intensity.
2. Gets the LED illumination intensity.

■Comments

1. Dimming with a linear curve.
2. It lights up when the intensity value is 200 or more. There is no guarantee that the light will light up when the intensity value is less than 200.
3. The last set value will be the initial value at the next startup.
4. When RLI is not connected, the response of the <Q> command returns 'X'.

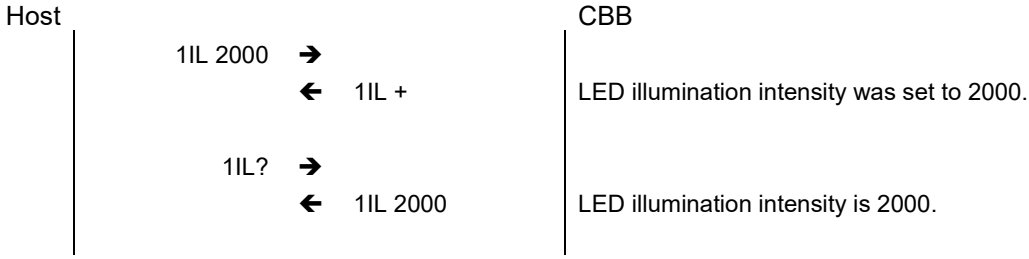
■Format

Command	Type	Direction	Description
IL *p1*	**R**	Host ➔ CBB	Controls the LED illumination intensity.
IL +	**PA**	Host ← CBB	Succeeded.
IL !,*error-code*	**NA**	Host ← CBB	Failed.
IL?	**Q**	Host ➔ CBB	Queries the LED illumination intensity value.
IL *p2*	**N**	Host ← CBB	Notifies the LED illumination intensity value.

■Parameters

p1	(0 - 65535)	Illumination intensity value
p2	(0 - 65535)	Illumination intensity value.
	X	Unknown

■Sequence



3.2.3.1.6. LED Illumination on / off Switch ILSW ILSW?

■Summary

1. Turn on / off the LED illumination.
2. Gets the LED illumination status.

■Comments

1. When RLI is not connected, the response of the <Q> command returns 'X'.
2. The last set value will be the initial value at the next startup.

■Format

Command	Type	Direction	Description
ILSW *p1*	**R**	Host ➔ CBB	Turn on / off the LED illumination.
ILSW +	**PA**	Host ← CBB	Succeeded.
ILSW !,*error-code*	**NA**	Host ← CBB	Failed.
ILSW?	**Q**	Host ➔ CBB	Queries the LED illumination status.
ILSW *p2*	**N**	Host ← CBB	Notifies the LED illumination status.

■Parameters

p1	0	off
	1	on
p2	0	off
	1	on
	X	Unknown

■Sequence

Host			CBB
1IL 2000	➔		
	←	1IL+	LED illumination intensity was set to 2000.
1ILSW 0	➔		
	←	1ILSW +	LED illumination was turned off.
1IL?	➔		
	←	1IL 2000	LED illumination intensity is 2000.
1ILSW 1	➔		
	←	1ILSW +	LED illumination was turned on. The intensity is 2000.

3.2.3.1.7. Aperture Stop Switching AS AS?

■Summary

1. Specifies the diameter of AS.
2. Gets the diameter of AS.

■Comments

1. When RLI is not connected, the response of the <Q> command returns 'X'.
2. The last set position will be the initial position at the next startup.

■Format

Command	Type	Direction	Description
AS *p1*	**R**	Host ➔ CBB	Specifies the diameter of AS.
AS +	**PA**	Host ← CBB	Succeeded.
AS !,*error-code*	**NA**	Host ← CBB	Failed.
AS?	**Q**	Host ➔ CBB	Queries the diameter of AS.
AS *p2*	**N**	Host ← CBB	Notifies the diameter of AS.

■Parameters

p1	(0 - 3113)	The driving address of AS
p2	(0 - 3113)	The driving address of AS
	X	Unknown

■Sequence

Host			CBB
1AS 1000	➔		
	←	1AS +	AS value was set to 1000.
1AS?	➔		
	←	1AS 1000	AS value is 1000.

3.2.3.1.8. Mirror Unit Switching MU MU?

■Summary

1. Controls the MU position.
2. Gets the MU position.

■Comments

1. When RLI is not connected, the response of the <Q> command returns 'X'.
2. The last set position will be the initial position at the next startup.

■Format

Command	Type	Direction	Description
MU *p1*	**R**	Host ➔ CBB	Controls the MU position.
MU +	**PA**	Host ← CBB	Succeeded.
MU !,*error-code*	**NA**	Host ← CBB	Failed.
MU?	**Q**	Host ➔ CBB	Queries the MU position.
MU *p2*	**N**	Host ← CBB	Notifies the MU position.

■Parameters

p1	(1 - 2)	MU position (1:BF, 2:DF)
p2	(1 - 2)	MU position (1:BF, 2:DF)
	X	Unknown

■Sequence

Host			CBB
1MU 1	➔		
	←	1MU +	Changed the position of MU to BF.
1MU?	➔		
	←	1MU 1	The position of MU is BF.

3.2.3.1.9. Filter Wheel 1 Switching FW1 FW1?

■Summary

1. Engages the filter wheel position specified into the light axis.
2. Gets the status of the filter wheel.

■Comments

1. When CBE1.1 is not connected, the response of the <Q> command returns 'X'.
2. The last set position will be the initial position at the next startup.

■Format

Command	Type	Direction	Description
FW1 *p1*	**R**	Host ➔ CBB	Engages the filter wheel position specified into the light axis.
FW1 +	**PA**	Host ← CBB	Succeeded.
FW1 !,*error-code*	**NA**	Host ← CBB	Failed.
FW1?	**Q**	Host ➔ CBB	Queries the filter wheel position engaged into the light axis.
FW1 *p2*	**N**	Host ← CBB	Notifies the filter wheel position engaged into the light axis.

■Parameters

p1	(1 - 6)	The filter wheel position.
p2	(1 - 6)	The filter wheel position.
	X	Unknown

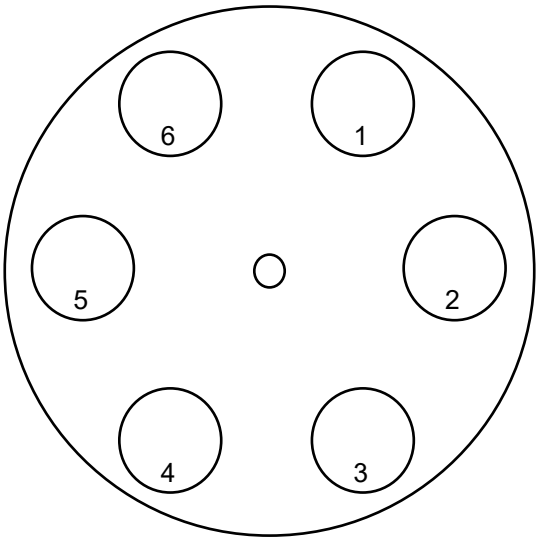


Fig 2 The filter wheel position (The filter side view)

■Sequence

Host			CBB
1FW1 6	➔		
	←	1FW1 +	Changed the position of FW to 6.
1FW1?	➔		
	←	1FW1 6	The position of FW is 6.

3.2.3.1.11. Objective Switching OB OB?

■Summary

1. Inserts the specified position of the revolving nosepiece into the light axis.
2. Gets the revolving nosepiece position.

■Comments

1. When the OB position is switched while the OB position is off the optical axis, the OB position is switched to the optical axis on the side where the OB position is smaller (CW direction), and then the OB position moves to the specified position.

■Format

Command	Type	Direction	Description
OB *p1*	**R**	Host ➔ CBB	Inserts the specified position of the revolving nosepiece into the light axis.
OB +	**PA**	Host ← CBB	Succeeded.
OB !,*error-code*	**NA**	Host ← CBB	Failed.
OB?	**Q**	Host ➔ CBB	Queries the revolving nosepiece position.
OB *p2*	**N**	Host ← CBB	Notifies the revolving nosepiece position.

■Parameters

p1	(1 - n)	The revolving nosepiece position; 1 - n n := (5, 6) **It depends on the flexibility of the revolving nosepiece equipped.**
p2	(1 - n)	The revolving nosepiece position; 1 - n n := (5, 6) **It depends on the flexibility of the revolving nosepiece equipped.**
	X	Unknown

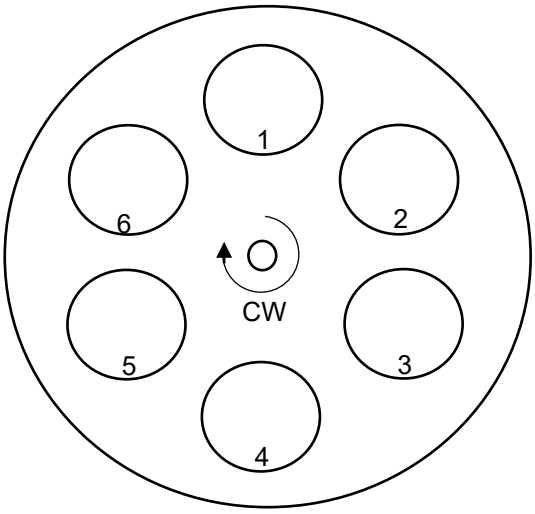


Fig 3 Revolving Nosepiece position (top view)

■Sequence

Host			CBB
	1OB 2 ➔		
	←	1OB +	Changed The revolving nosepiece position to 2.
	1OB? ➔		
	←	1OB 2	The revolving nosepiece position is 2.

3.2.3.1.12. OB Refresh　OBREF

■Summary

1. Make one rotation from the current revolving nosepiece position in the specified direction (CW or CCW).

■Comments

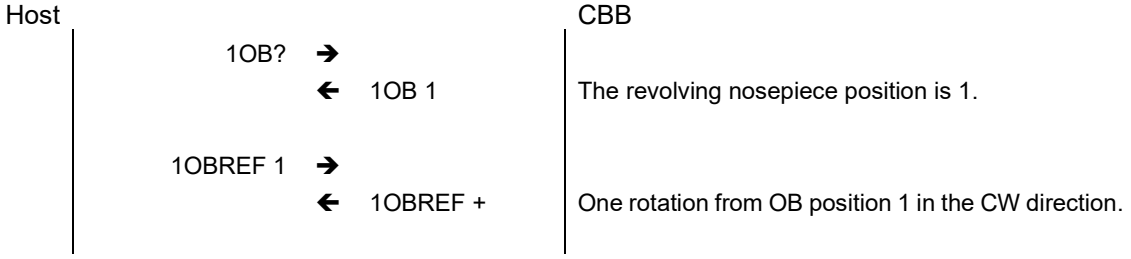
■Format

コマンド	種類	方向	意味
OBREF *p1*	**R**	Host ➔ CBB	Make one rotation from the current position in the specified direction.
OBREF +	**PA**	Host ← CBB	Succeeded.
OBREF !,*error-code*	**NA**	Host ← CBB	Failed.

■Parameters

p1	1	CW
	2	CCW

■Sequence



3.2.3.1.23. AF Parameter Table Selection AFTBL AFTBL?

■Summary

1. Selects the AF parameter table.
2. Gets the AF parameter table.

■Comments

1. The parameter table differs with respect to each observation method and objective. Whenever the observation method or objective is changed, this command is used to specify the corresponding parameter table number.
2. By setting the AF parameter table, Focus Status Judgment Signal（In-Focus, In-Capture）are output based on the parameters preset inside the CPU.
3. When this command is not used or when a parameter table number incompliant with the following list is specified, Focus Status Judgment Signal output becomes unstable.
4. The initial value is backed to the following "default" each time the power is turned on.
5. AFTBL can't be set during Focus Error Signal output.

■Format

Command	Type	Direction	Description
AFTBL *p1*	**R**	Host ➔ CBB	Selects the AF parameter table.
AFTBL +	**PA**	Host ← CBB	Succeeded.
AFTBL !,*error-code*	**NA**	Host ← CBB	Failed.
AFTBL?	**Q**	Host ➔ CBB	Queries the AF parameter table number.
AFTBL *p2*	**N**	Host ← CBB	Notifies the AF parameter table number.

■Parameters

p1	(0 - 109)	AF parameter table number. default: 0
p2	(0 - 109)	AF parameter table number.
	X	Unknown

Table 6 AF parameter table No. classification

Area	No.	Note
Fixed	0	Invalid AF parameter
	1 - 20	Reserved
	21 - 30	LM series BF: Various magnifications
	31 - 40	LM series DF: Various magnifications
	41 - 50	UM series BF: Various magnifications
	51 - 60	UM series DF: Various magnifications
	61 - 70	Other BF: Various magnifications
	71 - 79	Other DF: Various magnifications
	80 - 99	Reserved
User	100 - 109	User Setting

■Sequence

Host				CBB
Changing AF Table to 23. (LMPLFL10X)	1AFTBL 23	➔		
		←	1AFTBL +	
	1AFTBL?	➔		
		←	1AFTBL 23	
	1ESOUT 1	➔		Starts Focus Error Signal output.
		←	1ESOUT +	
Changing AF Table to 25. (LMPLFL50X)	1AFTBL 28	➔		
		←	1AFTBL !,*error-code*	

3.2.3.1.24. AF Parameter AFP GAFP

■Summary

1. Sets the AF control parameter.
2. Gets the AF control parameter.

■Comments

1. The parameter set with this command is canceled and returns to the preset parameter when the 'AFTBL' command changes the parameter table.
2. AFP can't be set during Focus Error Signal output.

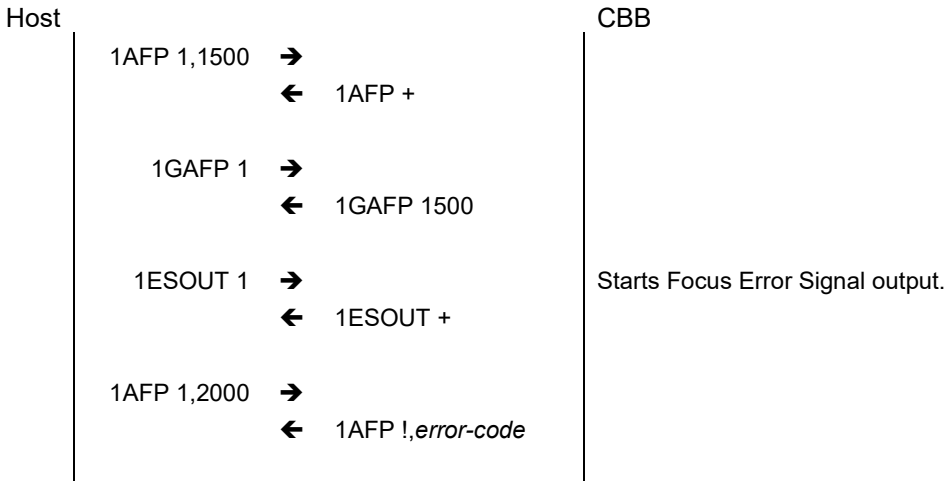
■Format

Command	Type	Direction	Description
AFP *p1,p2*	**R**	Host ➔ CBB	Sets the AF control parameter.
AFP +	**PA**	Host ← CBB	Succeeded.
AFP !,*error-code*	**NA**	Host ← CBB	Failed.
GAFP *p1*	**R**	Host ➔ CBB	Gets the AF control parameter.
GAFP *p2*	**PA**	Host ← CBB	AF control parameter is replied.
GAFP !,*error-code*	**NA**	Host ← CBB	Failed.

■Parameters

p1	(0 - 18)	AF control parameter number.
p2		Parameter data. Input range differs depending on *p1*. (See application notes)

■Sequence



3.2.3.1.25. AF Parameter Save User Area AFTBLCPY

■Summary

1. Save the AF parameters in the user area.

■Comments

1. Save the currently deployed AF parameters in the specified user area.

■Format

Command	Type	Direction	Description
AFTBLCPY *p1*	**R**	Host ➔ CBB	Save in the specified user area.
AFTBLCPY +	**PA**	Host ← CBB	Succeeded.
AFTBLCPY !,*error-code*	**NA**	Host ← CBB	Failed.

■Parameters

p1	(100 - 109)	User area number.

■Sequence

Host				CBB
Changing AF Table to 23. (LMPLFL10X)	1AFTBL 23	➔		
		←	1AFTBL +	
Change the AF parameters.	1AFP 1,1500	➔		
		←	1AFP +	
Save in the user area.	1AFTBLCPY 100	➔		
		←	1AFTBLCPY +	Saved in user area number 100.
Changing AF Table to 100. (User area)	1AFTBL 100	➔		
		←	1AFTBL +	
Gets the AF parameters.	1GAFP 1	➔		
		←	1GAFP 1500	

3.2.3.1.26. AF Parameter User Area Reset AFTBLRST

■Summary

1. Sets all user areas in the AF parameter table to factory default settings.

■Comments

■Format

Command	Type	Direction	Description
AFTBLRST	**R**	Host ➔ CBB	Sets user area to factory default.
AFTBLRST +	**PA**	Host ← CBB	Succeeded.
AFTBLRST !,*error-code*	**NA**	Host ← CBB	Failed.

■Sequence

Host				CBB
Changing AF Table to 23. (LMPLFL10X)	1AFTBL 23	➔		
		←	1AFTBL +	
Change the AF parameters.	1AFP 1,1500	➔		
		←	1AFP +	
Save in the user area.	1AFTBLCPY 100	➔		
		←	1AFTBLCPY +	Saved in user area number 100.
Changing AF Table to 100. (User area)	1AFTBL 100	➔		
		←	1AFTBL +	The AF parameter with user area number 100 has been deployed.
Gets the AF parameters.	1GAFP 1	➔		
		←	1GAFP 1500	
User area reset	1AFTBLRST	➔		
		←	1AFTBLRST +	
Gets the AF parameters.	1GAFP 1	➔		
		←	1GAFP 1500	(No change at this point)
Changing AF Table to 100. (User area)	1AFTBL 100	➔		
		←	1AFTBL +	The AF parameter with user area number 100 has been deployed.
Gets the AF parameters.	1GAFP 1	➔		
		←	1GAFP 1000	Gets factory default values.

3.2.3.1.33. Aberration Lens Absolute Movement ABG

■Summary

1. Moves the aberration lens. (absolute position)

■Comments

1. The last set position will be the initial position at the next startup.

■Format

Command	Type	Direction	Description
ABG *p1*	**R**	Host ➔ CBB	Moves the aberration lens. (absolute position)
ABG +	**PA**	Host ← CBB	Succeeded.
ABG !,*error-code*	**NA**	Host ← CBB	Failed.

■Parameters

p1	(0 - 13120)	Aberration lens position [p] Aberration lens movement resolution: 1.5625 um/pulse

■Sequence

Host			CBB
1ABG 1000	➔		Starts moving the aberration lens.
	←	1ABG +	The movement has been completed.
1ABP?	➔		
	←	1ABP 1000	Notify the position.
1ESOUT 1	➔		Starts Focus Error Signal output.
	←	1ESOUT +	
1ABG 2000	➔		Starts moving the aberration lens.
	←	1ABG +	The movement has been completed.

3.2.3.1.34. Aberration Lens Relative Movement ABM

■Summary

1. Moves the aberration lens. (relative position)

■Comments

■Format

Command	Type	Direction	Description
ABM *p1*	**R**	Host ➔ CBB	Moves the aberration lens. (relative position)
ABM +	**PA**	Host ← CBB	Succeeded.
ABM !,*error-code*	**NA**	Host ← CBB	Failed.

■Parameters

p1	(-13120 - 13120)	The moving distance (Step amount) [p]

■Sequence

Host			CBB
1ABP?	➔		
	←	1ABP 1000	Notify the position.
1ABM -1000	➔		Starts moving the aberration lens. (Moves the aberration correction lens toward the driving motor.)
	←	1ABM +	The movement has been completed.
1ABP?	➔		
	←	1ABP 0	Notify the position.
1ABM 1000	➔		Starts moving the aberration lens. (Moves the aberration correction lens in the opposite direction of the driving motor.)
	←	1ABM +	The movement has been completed.
1ABP?	➔		
	←	1ABP 1000	Notify the position.
1ESOUT 1	➔		Starts Focus Error Signal output.
	←	1ESOUT +	
1ABM 1000	➔		Starts moving the aberration lens.
	←	1ABM +	The movement has been completed.

3.2.3.1.35. Aberration Lens Position ABP?

■Summary

1. Gets the aberration lens position.

■Comments

1. When FSU is not connected, the response of the <Q> command returns ‘X’.

■Format

Command	Type	Direction	Description
ABP?	**Q**	Host ➔ CBB	Queries the aberration lens position.
ABP *p1*	**N**	Host ➔ CBB	Notifies the aberration lens position.

■Parameters

p1	(0 - 13120)	Aberration lens position [p]
	X	Unknown

■Sequence

Host			CBB
1ABG 1000	➔		Starts moving the aberration lens.
	➔	1ABG +	The movement has been completed.
1ABP?	➔		
	➔	1ABP 1000	Notify the position.

3.2.3.1.36. Safety Lock Status SL?

■Summary

1. Gets the status of the safety lock.

■Comments

1. ‘1’ is returned when the safety lock terminal is opened (unlocked) and ‘0’ is returned when it is short-circuited (locked).
2. When the safety lock terminal is opened, the laser is not irradiated, a Focus Error Signal is not output.
3. If the safety lock terminal is opened during the Focus Error Signal output, the laser is not irradiated, the Focus Error Signal output value does not change.

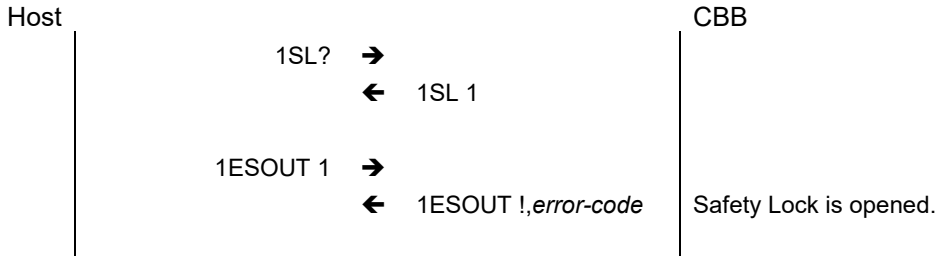
■Format

Command	Type	Direction	Description
SL?	**Q**	Host ➔ CBB	Queries about the status of the safety lock.
SL *p1*	**N**	Host ➔ CBB	Notifies of the status of the safety lock.

■Parameters

p1	0	Locked
	1	Unlocked
	X	Unknown

■Sequence



3.2.3.1.37. AF Operation Type AFTYPE AFTYPE?

■Summary

1. Sets the AF operation type.
2. Gets the AF operation type.

■Comments

1. '2' should be set to the AF type before executing 'ESOUT' (sparing '1').
2. The initial value is backed to the following "default" each time the power is turned on.
3. AFTYPE can't be set during Focus Error Signal output.

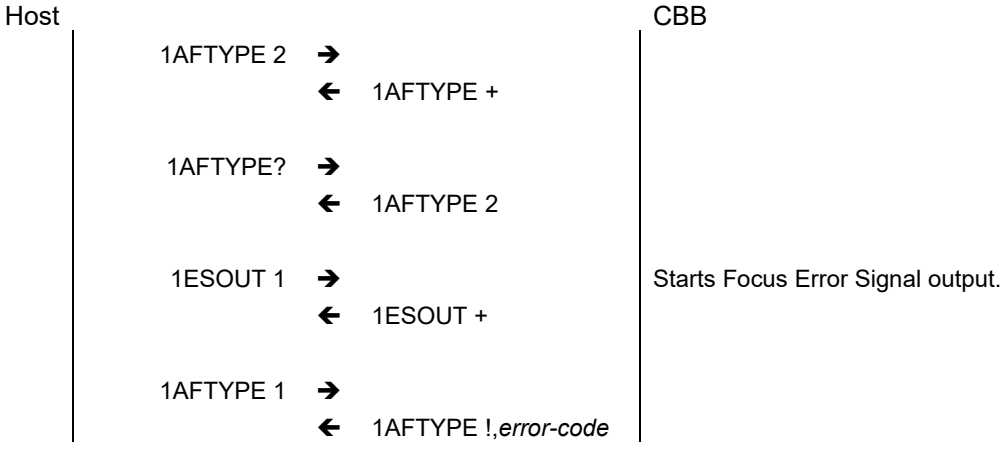
■Format

Command	Type	Direction	Description
AFTYPE *p1*	**R**	Host ➔ CBB	Sets the AF operation type.
AFTYPE +	**PA**	Host ← CBB	Succeeded.
AFTYPE !,*error-code*	**NA**	Host ← CBB	Failed.
AFTYPE?	**Q**	Host ➔ CBB	Queries the AF operation type.
AFTYPE *p2*	**N**	Host ← CBB	Notifies the AF operation type.

■Parameters

p1	1	Reserved default
	2	Focus Error Signal
p2	1	Reserved
	2	Focus Error Signal
	X	Unknown

■Sequence



3.2.3.1.38. Focus Error Signal Output ESOUT

■Summary

1. Focus Error Signal Controls the output operation.

■Comments

1. Set AFTYPE to 2 before starting Focus Error Signal output.
2. The initial value is backed to the following "default" each time the power is turned on.

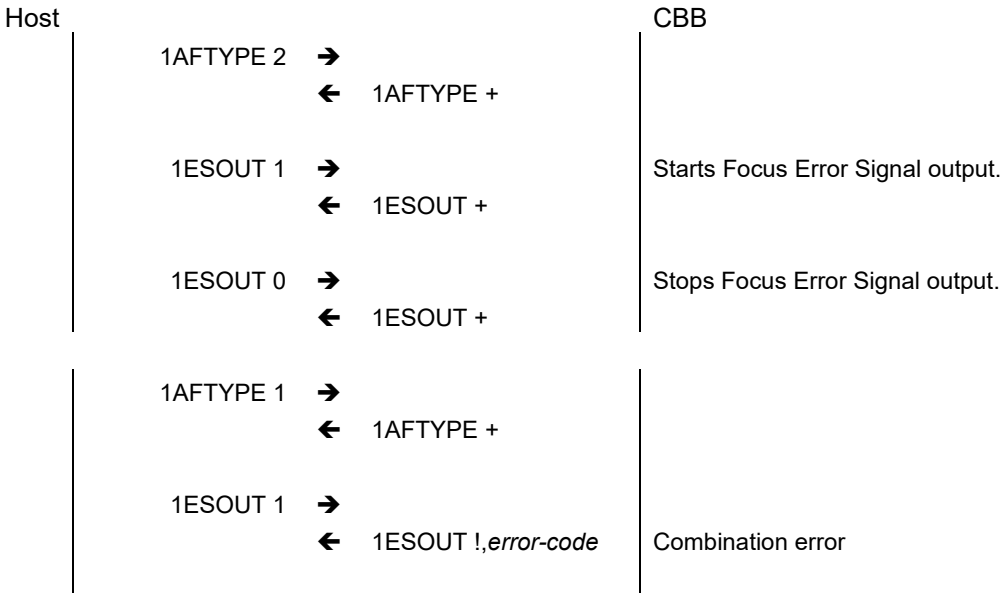
■Format

Command	Type	Direction	Description
ESOUT *p1*	**R**	Host ➔ CBB	Starts / stops Focus Error Signal output.
ESOUT +	**PA**	Host ➔ CBB	Succeeded.
ESOUT !,*error-code*	**NA**	Host ➔ CBB	Failed.

■Parameters

p1	0	Stops Focus Error Signal output. default
	1	Starts Focus Error Signal output.

■Sequence



3.2.3.1.39. Focus Error Signal Polarity ESPOL ESPOL?

■Summary

1. Sets the polarity of the Focus Error Signal.
2. Gets the polarity of the Focus Error Signal.

■Comments

1. Set the polarity of the Focus Error Signal output from D/A to standard or reverse.
2. ESPOL can't be set during Focus Error Signal output.
3. The initial value is backed to the following "default" each time the power is turned on.

■Format

Command	Type	Direction	Description
ESPOL *p1*	**R**	**Host** → CBB	Sets the polarity of the Focus Error Signal.
ESPOL +	**PA**	**Host** ← CBB	Succeeded.
ESPOL !,*error-code*	**NA**	**Host** ← CBB	Failed.
ESPOL?	**Q**	**Host** → CBB	Queries about the polarity of the Focus Error Signal.
ESPOL *p2*	**N**	**Host** ← CBB	Notifies of the polarity of the Focus Error Signal.

■Parameters

p1	0	The polarity of the Focus Error Signal is standard. default
	1	The polarity of the Focus Error Signal is reversed.
p2	0	The polarity of the Focus Error Signal is standard.
	1	The polarity of the Focus Error Signal is reversed.
	X	Unknown

■Sequence

Host		CBB
1ESPOL 1 →		
	← 1ESPOL +	
1ESPOL? →		
	← 1ESPOL 1	
1ESOUT 1 →		Starts Focus Error Signal output.
	← 1ESOUT +	
1ESPOL 0 →		
	← 1ESPOL !,*error-code*	
1ESPOL? →		
	← 1ESPOL 1	
1ESOUT 0 →		Stops Focus Error Signal output.
	← 1ESOUT +	
1ESPOL 0 →		
	← 1ESPOL +	

3.2.3.1.40. Focus Error Signal Range ESRNG ESRNG?

■Summary

1. Sets the range of the voltage of Focus Error Signal.
2. Gets the range of the voltage of Focus Error Signal.

■Comments

1. ESRNG can't be set during Focus Error Signal output.
2. The voltage level exceeding the setting range is not output. (See the following chart.)
3. The initial value is backed to the following "default" each time the power is turned on.

■Format

Command	Type	Direction	Description
ESRNG *p1*	**R**	Host ➔ CBB	Sets the range of the voltage of Focus Error Signal.
ESRNG +	**PA**	Host ← CBB	Succeeded.
ESRNG !,*error-code*	**NA**	Host ← CBB	Failed.
ESRNG?	**Q**	Host ➔ CBB	Queries about the range of the voltage of Focus Error Signal.
ESRNG *p2*	**N**	Host ← CBB	Notifies of the range of the voltage of Focus Error Signal.

■Parameters

p1	(0 - 100)	The range of the voltage of Focus Error Signal (±0.1 [V]). default: 100
p2	(0 - 100)	The range of the voltage of Focus Error Signal (±0.1 [V]).
	X	Unknown

■Sequence

Host			CBB
	1ESRNG 45	➔	Sets to output ±4.5V.
		← 1ESRNG +	
	1ESRNG?	➔	
		← 1ESRNG 45	
	1ESOUT 1	➔	Starts Focus Error Signal output.
		← 1ESOUT +	
	1ESRNG 40	➔	
		← 1ESRNG !,*error-code*	
	1ESRNG?	➔	
		← 1ESRNG 45	
	1ESOUT 0	➔	Stops Focus Error Signal output.
		← 1ESOUT +	
	1ESRNG 100	➔	Sets to output ±10V.
		← 1ESRNG +	

3.2.3.1.41. Focus Error Signal Gain ESGAIN ESGAIN?

■Summary

1. Sets the gain of Focus Error Signal.
2. Gets the gain of Focus Error Signal.

■Comments

1. ESGAIN can't be set during Focus Error Signal output.
2. The voltage level exceeding the setting range (set with 'ESRNG') is not output. (See the following chart.)
3. The initial value is backed to the following "default" each time the power is turned on.

■Format

Command	Type	Direction	Description
ESGAIN *p1*	**R**	Host ➔ CBB	Sets the gain of Focus Error Signal.
ESGAIN +	**PA**	Host ← CBB	Succeeded.
ESGAIN !,*error-code*	**NA**	Host ← CBB	Failed.
ESGAIN?	**Q**	Host ➔ CBB	Queries about the gain of Focus Error Signal.
ESGAIN *p2*	**N**	Host ← CBB	Notifies of the gain of Focus Error Signal.

■Parameters

p1	(0 - 100)	Gain (0.1x) default: 10
p2	(0 - 100)	Gain (0.1x)
	X	Unknown

■Sequence

Host			CBB
1ESGAIN 15	➔		Sets 1.5x of gain.
	←	1ESGAIN +	
1ESGAIN?	➔		
	←	1ESGAIN 15	
1ESOUT 1	➔		Starts Focus Error Signal output.
	←	1ESOUT +	
1ESGAIN 20	➔		
	←	1ESGAIN !,*error-code*	
1ESGAIN?	➔		
	←	1ESGAIN 15	
1ESOUT 0	➔		Stops Focus Error Signal output.
	←	1ESOUT +	
1ESGAIN 5	➔		Sets 0.5x of gain.
	←	1ESGAIN +	

3.2.3.1.42. Focus Error Signal Offset ESOFST ESOFST?

■Summary

1. Sets the offset value of Focus Error Signal.
2. Gets the offset value of Focus Error Signal.

■Comments

1. ESOFST can't be set during Focus Error Signal output.
2. The voltage level exceeding the setting range (set with 'ESRNG') is not output.
3. The initial value is backed to the following "default" each time the power is turned on.

■Format

Command	Type	Direction	Description
ESOFST *p1*	**R**	Host ➔ CBB	Sets the offset value of Focus Error Signal.
ESOFST +	**PA**	Host ← CBB	Succeeded.
ESOFST !,*error-code*	**NA**	Host ← CBB	Failed.
ESOFST?	**Q**	Host ➔ CBB	Queries about the offset value of Focus Error Signal.
ESOFST *p2*	**N**	Host ← CBB	Notifies of the offset value of Focus Error Signal.

■Parameters

p1	(-100 - 100)	Offset value (0.1 [V]) default: 0
p2	(-100 - 100)	Offset value (0.1 [V])
	X	Unknown

■Sequence

Host			CBB
1ESOFST 15	➔		Sets 1.5V of offset value.
	←	1ESOFST +	
1ESOFST?	➔		
	←	1ESOFST 15	
1ESOUT 1	➔		Starts Focus Error Signal output.
	←	1ESOUT +	
1ESOFST 20	➔		
	←	1ESOFST !,*error-code*	
1ESOFST?	➔		
	←	1ESOFST 15	
1ESOUT 0	➔		Stops Focus Error Signal output.
	←	1ESOUT +	
1ESOFST 5	➔		Sets 0.5V of offset value.
	←	1ESOFST +	

3.2.3.1.43. Focus Status Judgement Signal Polarity LSPOL LSPOL?

■Summary

1. Sets the polarity of the Focus Status Judgement Signal (In-Focus, In-Capture).
2. Gets the polarity of the Focus Status Judgement Signal.

■Comments

1. LSPOL can't be set during Focus Error Signal output.
2. The initial value is backed to the following "default" each time the power is turned on.

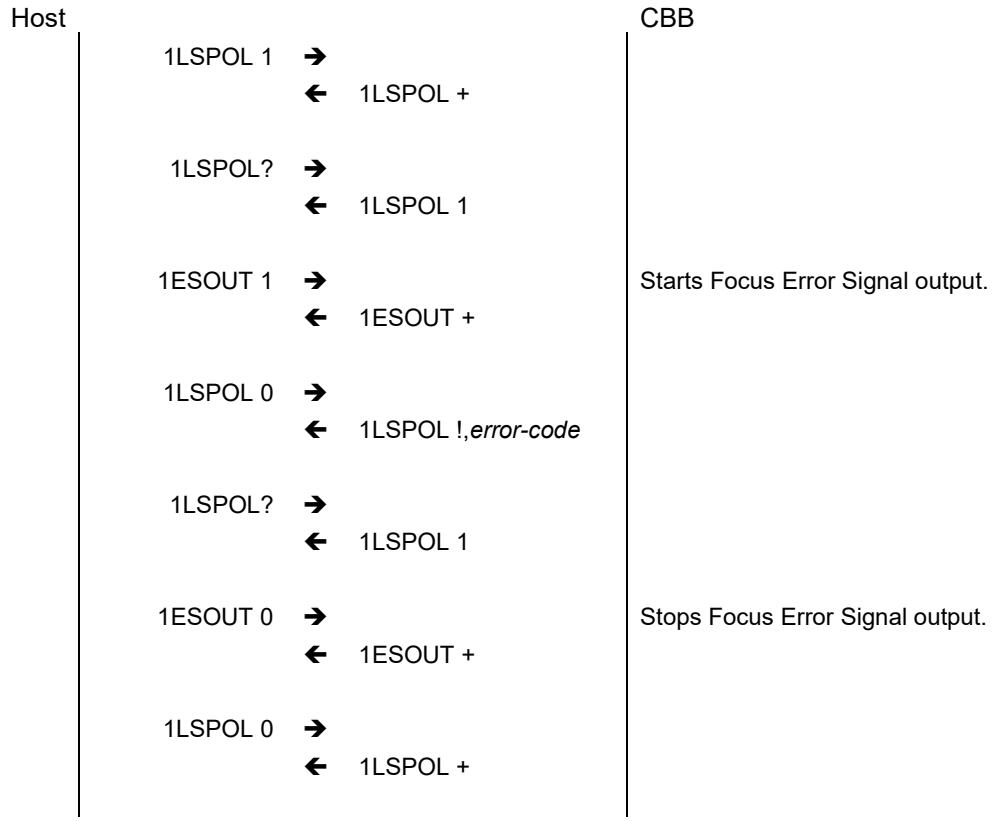
■Format

Command	Type	Direction	Description
LSPOL *p1*	**R**	Host ➔ CBB	Sets the polarity of the Focus Status Judgement Signal.
LSPOL +	**PA**	Host ← CBB	Succeeded.
LSPOL !,*error-code*	**NA**	Host ← CBB	Failed.
LSPOL?	**Q**	Host ➔ CBB	Queries the current Focus Status Judgement Signal polarity.
LSPOL *p2*	**N**	Host ← CBB	Notifies the current Focus Status Judgement Signal polarity.

■Parameters

p1	0	Negative logic (Active "L") default
	1	Positive logic (Active "H")
p2	0	Negative logic (Active "L")
	1	Positive logic (Active "H")
	X	Unknown

■Sequence



3.2.3.1.44. Error ER ER?

■Summary

1. Notifies of an error.
2. Gets the last errors.

■Comments

1. When an error occurs, the error notification <EN> is sent. This <EN> can neither be disabled nor controlled.
2. The parameter of the notification <N> is up to 4. And after gets the last errors, the errors are all cleared.

■Format

Command	Type	Direction	Description
ER *error-code*	**EN**	Host ← CBB	Notifies of an error.
ER?	**Q**	Host → CBB	Gets the error.
ER *error-code,…*	**N**	Host ← CBB	Notifies of an error.

■Parameters

error-code	(E00000000 - EFFFFFFFF)	9[B] fixed length (cf. Error code list)

■Sequence

Host			CBB
1ER?	→		
	←	1ER E00000000	No error
1OB 1	→		
	←	1OB !,*error-code(OB timeout)*	OB timeout
1ER?	→		
	←	1ER *error-code(OB timeout)*	Notifying the last error-code. The errors are all cleared.
1ER?	→		
	←	1ER E00000000	No error
1OB 1	→		
	←	1OB !,*error-code(OB timeout)*	OB timeout
	←	1ER *error-code(OB* disconnected*)*	OB disconnected
1ER?	→		
	←	1ER *error-code(OB timeout)*,*error-code(OB* disconnected*)*	Notifying the last error-code. The errors are all cleared.
1ER?	→		
	←	1ER E00000000	No error

3.2.3.1.45. DIP-SW1 Read DSW1?

■Summary

1. Gets the CBB DIP-SW settings.

■Comments

1. Returns the DIP-SW settings read when the power is turned on.
2. The value is constructed with 8 bit image.

0: Switch off, 1: Switch on

BIT0: SW1
BIT1: SW2
BIT2: SW3
BIT3: SW4
BIT4: SW5
BIT5: SW6
BIT6: SW7
BIT7: SW8

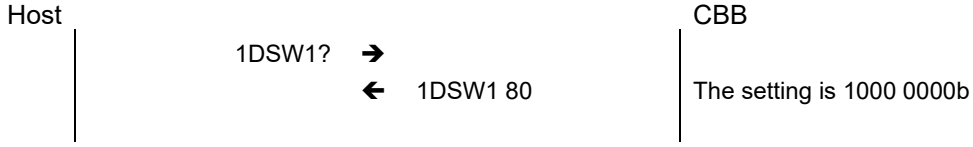
■Format

Command	Type	Direction	Description
DSW1?	**Q**	Host ➔ CBB	Queries the DIP-SW settings.
DSW1 *p1*	**N**	Host ← CBB	Notifies the DIP-SW settings.

■Parameters

p1	(0 - FF)	DIP-SW settings (hexadecimal) Variable length, A to F are uppercase.

■Sequence



3.2.3.1.46. DIP-SW2 Read DSW2?

■Summary

1. Gets the CBE1.1 DIP-SW settings.

■Comments

1. Returns the DIP-SW settings read when the power is turned on.
2. The value is constructed with 8 bit image.

0: Switch off, 1: Switch on

BIT0: SW1
BIT1: SW2
BIT2: SW3
BIT3: SW4
BIT4: SW5
BIT5: SW6
BIT6: SW7
BIT7: SW8

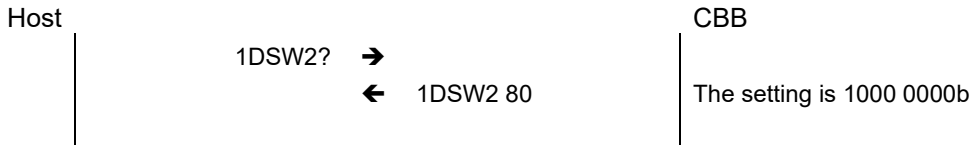
■Format

Command	Type	Direction	Description
DSW2?	**Q**	Host ➔ CBB	Queries the DIP-SW settings.
DSW2 *p1*	**N**	Host ← CBB	Notifies the DIP-SW settings.

■Parameters

p1	(0 - FF)	DIP-SW settings (hexadecimal) Variable length, A to F are uppercase.
	X	Unknown

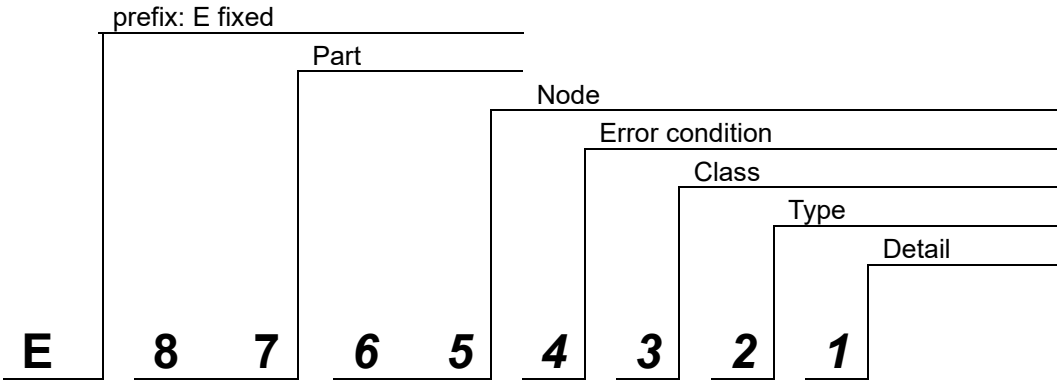
■Sequence



4. Error Code List

Error codes are described as follows. The error codes are created based on the hierarchical structure, and the meaning of the lower digits differs depending on the upper digits.

error-code	E00000000 - EFFFFFFFF	9[B] fixed length with hexadecimal 8 digits following after prefix: E



8th and 7th digit	code	Description	Notes
Guest	01 - 15	Guest ID	Indicates the guest ID of CBB. The guest number of CBB is fixed with "01".

6th and 5th digit	code	Description	Notes
Node	01 - 3F	Node ID	Indicates the node ID of unit.

4th digit	code	Description	Notes
Error condition	0	Warning or nonfatal error	When causes are removed, most of the cases can be restored.
	1	Fatal error	Unable to restore, and the relevant part/unit will be locked.

■ The Class differs depending on the function performed by parts or nodes. All parts or nodes do not necessarily have all classes.

3th digit	code	Description
Class	1	Command
	2	Motorized (including the light source)
	3	AF control
	4	Limit
	5	System
	6	MMI (Man Machine I/F e.g. HS, JOG, JS, etc.)
	7	Nonvolatile memory
	8 - 0	Reserved

■ Further Type (2nd digit) and Detail (1st digit) indicate the different meaning by each Class.

4.1. CBB

Error codes of CBB are described as follows. However, prefix:E is omitted.

Part	Class	Type	error-code*1	Description
All	Command	Nesting error	01xx0110	The command which cannot execute nesting, nested.
		Parameter error	01xx0120	The parameter exceeds the area. Too many parameters or too few parameters.
		Combination error	01xx0130	Unacceptable due to the combination with other status. No target part exists.
			01xx0132	Disable to execute due to the undefined coordinates.
			01xx0133	The operation is interrupted due to the stop command.

*1 xx indicates any Node. It can obtain 01 - 3F.

Part	Class	Type	error-code	Description
0x3F (CBB)	Command	System lock	013F1150	The system locked state due to a fatal error.
		Internal I/F time-out	013F1161	FSU
			013F1162	CBE1.1
			013F1164	RLI
	System	Unit duplication detection	013F1511	FSU
			013F1512	CBE1.1
			013F1514	RLI
		Unit disconnection	013F1521	Unit1
			013F1522	Unit2
			013F1523	Unit3

Part	Class	Type	error-code	Description
0x01 (FSU)	Motorized	Aberration collection lens	01011211	Initialize sensor search error (The expected sensor logic level is not detected after the regular pulse is exported)
			01011212	Initialize sensor search error (Sensor conflict)
			01011213	Initialize sensor escape error (The expected sensor logic level is not detected after the regular pulse is exported)
			01011214	Initialize sensor escape error (Sensor conflict)
			01011215	Initialize error in determining the home position. (The expected sensor logic level is not detected after the regular pulse is exported)
			01011216	Initialize error in determining the home position. (The sensor latching is failed)
			01011217	Initialize timeout in determining the home position. (Initialization has not been completed within the regular timeframe)
	AF control	AF	01011331	Excessive light was detected.
			01010332	Safety Lock is opened.
	Limit	Aberration collection lens	01011411	Minus sensor limit
			01010412	Minus software limit
			01010413	Plus software limit
			01011414	Plus sensor limit
	System	Aberration collection lens	01011510	Driver IC error (Overtemperature)
			01011511	Driver IC error (Overtemperature pre-warning)
			01011512	Driver IC error (Short to GND detected on phase A)
			01011513	Driver IC error (Short to GND detected on phase B)

Part	Class	Type	error-code	Description
0x02 (CBE1.1)	Command	Internal I/F time-out	01021161	FSU
	Motorized	FW	01020211	Initialize failed.
			01020212	Move timeout. (Failed to move within a given time period)
			01021213	Disconnected
	Nonvolatile memory	FRAM	01021711	AF table is invalid.

Part	Class	Type	error-code	Description
0x03 (RLI)	Motorized	Mirror unit	01030210	Move timeout. (Failed to move within a given time period)
			01030211	Over run
			01030213	timeout (Escape of sensor is not detected)
			01030214	timeout (Entering of sensor is not detected)
		AS	01031221	Initialize sensor search error (The expected sensor logic level is not detected after the regular pulse is exported)
			01031222	Initialize sensor search error (Sensor conflict)
			01031223	Initialize sensor escape error (The expected sensor logic level is not detected after the regular pulse is exported)
			01031224	Initialize sensor escape error (Sensor conflict)
			01031225	Initialize error in determining the home position. (The expected sensor logic level is not detected after the regular pulse is exported)
			01031226	Initialize error in determining the home position. (The sensor latching is failed)
			01031227	Initialize timeout in determining the home position. (Initialization has not been completed within the regular timeframe)
		Revolving nosepiece	01030230	timeout (Moving to next click in position)
			01030231	Over run
			01030232	sensor error (Position sensor unsettle)
			01030233	timeout (Escape of Click sensor is not detected)
			01030234	timeout (Entering of Click sensor is not detected)
			01031236	Disconnected
	Limit	AS	01031421	Minus sensor limit
			01030422	Minus software limit
			01030423	Plus software limit
		Revolving nosepiece	01030432	1 side software limit
			01030433	Max side software limit
	System	AS	01031520	Driver IC error (Overtemperature)
			01031521	Driver IC error (Overtemperature pre-warning)
			01031522	Driver IC error (Short to GND detected on phase A)
			01031523	Driver IC error (Short to GND detected on phase B)

Manufactured by

EVIDENT CORPORATION

6666 Inatomi, Tatsuno-machi, Kamiina-gun, Nagano 399-0495, Japan

Distributed by

EVIDENT EUROPE GmbH

Caffamacherreihe 8-10, 20355 Hamburg, Germany

EVIDENT EUROPE GmbH UK Branch

Part 2nd Floor Part A, Endeavour House, Coopers End Road, Stansted CM24 1AL, U.K.

EVIDENT SCIENTIFIC, INC.

48 Woerd Ave Waltham, MA 02453, U.S.A.

EVIDENT AUSTRALIA PTY LTD

97 Waterloo Road, Macquarie Park, NSW 2113, Australia

Life science solutions	Industrial solutions
Service Center	**Service Center**
https://www.olympus-lifescience.com/support/service/	https://www.olympus-ims.com/service-and-support/service-centers/
Official website	**Official website**
https://www.olympus-lifescience.com	https://www.olympus-ims.com

AM5154 02　2023 年 8 月発行