



# DA1000/DA1100 Alignment Controller Software Reference manual

# Contents

---

1. Before Use .....	1
1.1. Introduction .....	1
1.2. Related Documents .....	1
1.3. Precautions .....	1
2. Setup .....	2
2.1. External PC Connection Setup .....	2
2.1.1. External PC Communication Runtime Setup .....	2
2.1.2. DLL Installation .....	2
3. Command List .....	3
3.1. System Class .....	3
3.2. AxisComponents Class .....	3
3.3. Axis2D Class .....	4
3.4. Axis3D Class .....	5
3.5. Alignment Class .....	6
3.6. Profile Class .....	7
3.7. AngleAdjustment Class .....	7
3.8. IO Class .....	8
3.9. PowerMeter Class .....	8
4. Command Details .....	9
4.1. Machine-dependent parameters .....	9
4.2. System Class .....	10
4.2.1. Methods .....	10
4.2.1.1. GetInstance .....	10
4.2.1.2. SetAddress .....	10
4.2.1.3. Connected(C#) IsConnected(C++) .....	10
4.2.1.4. IsError .....	10
4.2.1.5. GetErrorCode .....	10
4.2.1.6. IsEmergencyAsserted .....	10
4.3. AxisComponents Class .....	11
4.3.1. Methods .....	11
4.3.1.1. AxisComponents .....	11
4.3.1.2. MoveAbsolute .....	11
4.3.1.3. MoveRelative .....	11
4.3.1.4. ReturnOrigin .....	11
4.3.1.5. Stop .....	11
4.3.1.6. TurnOnServo .....	12
4.3.1.7. TurnOffServo .....	12

4.3.1.8. ResetError .....	12
4.3.1.9. GetTargetPosition.....	12
4.3.1.10. GetActualPosition.....	12
4.3.1.11. IsServoOn .....	12
4.3.1.12. IsServoAlarm .....	13
4.3.1.13. IsMoving.....	13
4.3.1.14. IsOriginReturnSucceeded .....	13
4.3.1.15. IsOriginSignalOn.....	13
4.3.1.16. IsCwLimitSignalOn.....	13
4.3.1.17. IsCcwLimitSignalOn .....	13
4.3.1.18. IsCwTorqueLimitOn.....	14
4.3.1.19. IsCcwTorqueLimitOn .....	14
4.3.1.20. SetMaxSpeed.....	14
4.3.1.21. SetCWTorqueLimit.....	14
4.3.1.22. SetCCWTorqueLimit .....	14
4.3.1.23. SetAccelRate .....	15
4.3.1.24. SetDecelRate .....	15
4.3.1.25. SetSineMotion.....	15
4.3.1.26. SetFulcrumSpan .....	15
4.3.1.27. SetEncoderStepLength.....	16
4.3.1.28. GetErrorCode.....	16
4.3.1.29. GetStatus .....	16
4.3.2. Enumeration .....	16
4.3.2.1. ErrorCode .....	16
4.3.2.2. Status.....	17
4.4. Axis2D Class.....	18
4.4.1. Methods.....	18
4.4.1.1. Axis2D.....	18
4.4.1.2. GetAxisNumber .....	18
4.4.1.3. MoveAbsolute.....	18
4.4.1.4. MoveRelative.....	18
4.4.1.5. MoveAbsoluteAngle .....	19
4.4.1.6. MoveRelativeAngle .....	19
4.4.1.7. Stop .....	19
4.4.1.8. GetTargetPosition.....	19
4.4.1.9. GetActualPosition.....	20
4.4.1.10. IsMoving.....	20
4.4.1.11. SetSpeed .....	20

4.4.1.12. SetAccelRate .....	20
4.4.1.13. SetDecelRate .....	20
4.4.1.14. SetAxisNumber .....	21
4.4.1.15. GetErrorCode .....	21
4.4.1.16. GetStatus .....	21
4.4.1. Structure .....	21
4.4.1.1. Point.....	21
4.4.2. Enumeration .....	21
4.4.2.1. ErrorCode .....	21
4.4.2.2. Status.....	21
4.5. Axis3D Class.....	22
4.5.1. Methods.....	22
4.5.1.1. Axis3D.....	22
4.5.1.2. GetAxisNumber .....	22
4.5.1.3. MoveAbsolute.....	22
4.5.1.4. MoveRelative.....	24
4.5.1.5. Stop .....	25
4.5.1.6. GetTargetPosition .....	25
4.5.1.7. GetActualPosition .....	25
4.5.1.8. IsMoving.....	26
4.5.1.9. SetSpeed .....	26
4.5.1.10. SetAccelRate .....	26
4.5.1.11. SetDecelRate .....	26
4.5.1.12. SetStageNumber.....	26
4.5.1.13. SetSubStageNumber .....	27
4.5.1.14. SetRotationCenter .....	27
4.5.1.15. GetErrorCode.....	27
4.5.1.16. GetStatus .....	28
4.5.2. Structure .....	28
4.5.2.1. Point3D.....	28
4.5.2.2. RotationCenter (C#).....	28
4.5.3. Enumeration .....	28
4.5.3.1. ErrorCode .....	28
4.5.3.2. Status.....	29
4.6. Alignment Class .....	30
4.6.1. Methods.....	30
4.6.1.1. Alignment .....	30
4.6.1.2. StartSingle .....	30

4.6.1.3. StartFlat.....	30
4.6.1.4. StartFocus .....	30
4.6.1.5. StartTwoChRotation.....	30
4.6.1.6. StartMMFRotation .....	30
4.6.1.7. StartLoopBack .....	31
4.6.1.8. Stop .....	31
4.6.1.9. GetStatus.....	31
4.6.1.10. GetErrorAxisID.....	31
4.6.1.11. GetVoltage .....	31
4.6.1.12. GetPower .....	32
4.6.1.13. GetMeasurementWaveLength.....	32
4.6.1.14. GetAligningStatus .....	32
4.6.1.15. GetProfilePacketSumNumber .....	32
4.6.1.16. RequestProfileData.....	33
4.6.1.17. SetSingle.....	33
4.6.1.18. SetFlat .....	34
4.6.1.19. SetFocus .....	34
4.6.1.20. SetTwoChRotation.....	35
4.6.1.21. SetMMFRotation .....	35
4.6.1.22. SetLoopBack .....	36
4.6.1.23. SetMeasurementWaveLength.....	36
4.6.2. Structure .....	37
4.6.2.1. SingleParameter .....	37
4.6.2.2. FlatParameter.....	38
4.6.2.3. FocusParameter.....	39
4.6.2.4. TwoChRotationParameter .....	40
4.6.2.5. MMFRotationParameter.....	41
4.6.2.6. LooopBackParameter .....	42
4.6.2.7. ProfileData .....	43
4.6.2.8. RotationCenter.....	43
4.6.3. Enumeration .....	44
4.6.3.1. ErrorCode .....	44
4.6.3.2. Status.....	44
4.6.3.3. Zmode .....	44
4.6.3.4. AligningStatusCode .....	45
4.6.3.5. ProfileDataType .....	45
4.7. Profile Class.....	46
4.7.1. Methods.....	46

4.7.1.1. Profile .....	46
4.7.1.2. SetProfile .....	46
4.7.1.3. Start .....	46
4.7.1.4. Stop .....	46
4.7.1.5. GetProfileStatus .....	47
4.7.1.6. GetProfilePacketSumIndex.....	47
4.7.1.7. RequestProfileData.....	47
4.7.2. Structure .....	48
4.7.2.1. ProfileParameter .....	48
4.7.2.2. ProfileData .....	48
4.7.2.3. RotationCenter.....	48
4.7.3. Enumeration .....	49
4.7.3.1. ErrorCode .....	49
4.7.3.2. Status.....	49
4.8. AngleAdjustment Class .....	50
4.8.1. Methods.....	50
4.8.1.1. AngleAdjustment .....	50
4.8.1.2. Start.....	50
4.8.1.3. Stop .....	50
4.8.1.4. GetStatus.....	50
4.8.1.5. GetErrorAxisID.....	50
4.8.1.6. GetAdjustingStatus .....	51
4.8.1.7. GetProfilePacketSumIndex.....	51
4.8.1.8. RequestProfileData.....	51
4.8.1.9. SetParameter .....	51
4.8.2. Structure .....	53
4.8.2.1. AngleAdjustmentParameter .....	53
4.8.2.2. ProfileData .....	54
4.8.2.3. RotationCenter.....	54
4.8.3. Enumeration .....	55
4.8.3.1. ErrorCode .....	55
4.8.3.2. Status.....	55
4.8.3.3. AdjustingStatus .....	55
4.8.3.4. ProfileDataType .....	56
4.9. IO Class.....	57
4.9.1. Methods.....	57
4.9.1.1. IO .....	57
4.9.1.2. GetPortState .....	57

4.9.1.3. SetPortState.....	57
4.9.1.4. SetPulseOutput.....	57
4.9.1.5. GetAnalogValue .....	58
4.9.1.6. SetAnalogValue.....	58
4.9.2. Enumeration.....	58
4.9.2.1. DigitalIOType.....	58
4.9.2.2. AnalogIOType.....	58
4.10. PowerMeter Class .....	59
4.10.1. Methods.....	59
4.10.1.1. PowerMeter .....	59
4.10.1.2. GetPower .....	59
4.10.1.3. GetRange.....	59
4.10.1.4. GetWavelength.....	59
4.10.1.5. SetRange .....	60
4.10.1.6. SetWavelength .....	60
4.10.1.7. PauseCommunication.....	60
4.10.2. Enumeration.....	61
4.10.2.1. ErrorCode.....	61
4.10.2.2. AllowedRange.....	61

## 1. Before Use

---

### 1.1. Introduction

---

This document explains the external control function of the alignment controller DA1000/DA1100.  
The content of this document is compatible with software version Ver.1.3.4 and later.

### 1.2. Related Documents

---

DA1000 User Manual, DA1100 User Manual

DA1000 External PC Communication Runtime Setup Manual

### 1.3. Precautions

---

- Unauthorized reproduction or duplication of any part or all of this document is strictly prohibited.
- For product improvement, the contents of this document (specifications, software versions, etc.) are subject to change without notice.
- Regardless of the above, our company cannot be held responsible for the results of operation via external control function.
- The product names and company names mentioned in this manual are registered trademarks or trademarks of their respective companies.



## 2. Setup

---

### 2.1. External PC Connection Setup

---

#### 2.1.1. External PC Communication Runtime Setup

Before using the DLL, set up an execution environment on the control external PC for communication with the controller. Please refer to the separate 'External PC Communication Runtime Setup Manual' for the setup procedures.

#### 2.1.2. DLL Installation

It is available to make a control program using the DLL (for C#, for C++) provided by Suruga Seiki.

Item	C#	C++
DLL	srgmc.dll	srgmc.dll srgmc.lib
Header File	N/A	System.h AxisComponent.h Axis2D.h Axis3D.h Alignment.h Profile.h AngleAdjustment.h IO.h PowerMeter.h
Required Environment	.NET 6.0 or .NET Framework4.8	-
Third-Party Package	Beckhoff .NET ADS Communication Package Beckhoff.TwinCAT.Ads (In Visual Studio, install version 6.0.356 or later from NuGet.)	-
Third-Party DLL	-	Beckhoff C++ ADS Communication DLL TcAdsDll.dll TcAdsDll.lib
Namespace	SurugaSeiki.Motion	SurugaSeiki.Motion

Please refer to the manual of the development environment being used for actual usage of the DLL in each language.

### 3. Command List

---

#### 3.1. System Class

---

Method	Description
GetInstance	Retrieves an instance of the class object.
SetAddress	Sets the address of the connection target.
Connected (C#), IsConnected (C++)	Checks the connection status with the connection target.
IsError	Returns whether there is an error.
GetErrorCode	Retrieves the error code related to the entire system.
IsEmergencyAsserted	Retrieves the input state of the TQ-OFF terminal.

#### 3.2. AxisComponents Class

---

Method	Description
AxisComponents	Constructor
MoveAbsolute	Absolute position movement
MoveRelative	Relative position movement
ReturnOrigin	Start of return to origin
Stop	Axis stop
TurnOnServo	Servo ON
TurnOffServo	Servo OFF
ResetError	Error reset
GetTargetPosition	Retrieves target position
GetActualPosition	Retrieves current axis position
IsServoOn	Retrieves servo state
IsServoAlarm	Retrieves servo alarm state
IsMoving	Retrieves stage drive state
IsOriginReturnSucceeded	Retrieves return to origin state
IsOriginSignalOn	Retrieves the state of the origin sensor
IsCwLimitSignalOn	Retrieves the state of the CW limit sensor
IsCcwLimitSignalOn	Retrieves the state of the CCW limit sensor
IsCwTorqueLimitOn	Retrieves the state of the CW torque limit sensor
IsCcwTorqueLimitOn	Retrieves the state of the CCW torque limit sensor
SetMaxSpeed	Sets maximum speed
SetCWTorqueLimit	Sets CW torque limit
SetCCWTorqueLimit	Sets CCW torque limit
SetAccelRate	Sets acceleration

SetDecelRate	Sets deceleration
SetSineMotion	Sets enable/disable of sine motion
SetFulcrumSpan	Sets fulcrum span
SetEncoderStepLength	Sets encoder step length
GetErrorCode	Retrieves error code related to axis operation
GetStatus	Retrieves axis status
Enumeration	Description
ErrorCode	Axis control error code
Status	Axis status

### 3.3. Axis2D Class

Method	Description
Axis2D	Constructor
GetAxisNumber	Retrieves axis number
MoveAbsolute	Absolute position 2-axis movement
MoveRelative	Relative position 2-axis movement
MoveAbsoluteAngle	Absolute position movement with angle offset
MoveRelativeAngle	Relative position movement with angle offset
Stop	Stop 2-axis movement
GetTargetPosition	Retrieves target position
GetActualPosition	Retrieves current position
IsMoving	Retrieves 2-axis stage drive state
SetSpeed	Sets 2-axis combined speed
SetAccelRate	Sets 2-axis acceleration
SetDecelRate	Sets 2-axis deceleration
SetAxisNumber	Sets axis number
GetErrorCode	Retrieves error code related to 2-axis operation
GetStatus	Retrieves axis status
Structure	Description
Point	2D coordinates
Enumeration	Description
ErrorCode	Axis control error code
Status	Axis status.

### 3.4. Axis3D Class

Method	Description
Axis3D	Constructor
GetAxisNumber	Retrieves axis number
MoveAbsolute	Absolute position 3-axis movement
MoveRelative	Relative position 3-axis movement
Stop	Stop 3-axis movement
GetTargetPosition	Retrieves target position
GetActualPosition	Retrieves current position
IsMoving	Retrieves 3-axis stage drive state
SetSpeed	Sets 3-axis combined speed
SetAccelRate	Sets 3-axis acceleration
SetDecelRate	Sets 3-axis deceleration
SetStageNumber	Sets the axis number of the main axis to be used
SetSubStageNumber	Sets the axis number of the sub-axis to be used
SetRotationCenter	Sets the distance of rotation center shift
GetErrorCode	Retrieves error code related to 3-axis operation
GetStatus	Retrieves axis status
Structure	Description
Point3D	3D Coordinates
RotationCenter (C#)	Parameters for rotation center shift
Enumeration	Description
ErrorCode	Axis control error code
Status	Axis status

### 3.5. Alignment Class

Method	Description
Alignment	Constructor
StartSingle	Start Single alignment
StartFlat	Start Flat alignment
StartFocus	Start Focus alignment
StartTwoChRotation	Start Two-Ch rotation alignment
StartMMFRotation	Start MMF rotation alignment
StartLoopBack	Start Loop back alignment
Stop	Stop the alignment in progress
GetStatus	Retrieves alignment status
GetErrorAxisID	Retrieves error Axis ID
GetVoltage	Retrieves voltage value
GetPower	Retrieves power value
GetMeasurementWaveLength	Retrieves measurement wavelength
GetAligningStatus	Retrieves search status during alignment
GetProfilePacketSumIndex	Retrieves the number of alignment profile packets
RequestProfileData	Retrieves alignment profile
SetSingle	Sets Single alignment parameter
SetFlat	Sets Flat alignment parameter
SetFocus	Sets Focus alignment parameter
SetTwoChRotation	Sets Two-Ch rotation alignment parameter
SetMMFRotation	Sets MMF rotation alignment parameter
SetLoopBack	Sets Loop back alignment parameter
SetMeasurementWaveLength	Sets measurement wavelength
Structure	Description
SingleParameter	Single alignment parameter
FlatParameter	Flat alignment parameter
FocusParameter	Focus alignment parameter
TwoChRotationParameter	Two-Ch rotation alignment parameter
MMFRotationParameter	MMF rotation alignment parameter
LoopBackParameter	Loop back alignment parameter
ProfileData	Profile data
RotationCenter (C++)	Parameters for rotation center shift
Enumeration	Description
ErrorCode	Alignment error code
Status	Alignment status

Zmode	Field search Z mode
AligningStatusCode	Search status during execution
ProfileDataType	Type of alignment profile

### 3.6. Profile Class

Method	Description
Profile	Constructor
SetProfile	Sets profile parameter
Start	Start profile measurement
Stop	Stop profile measurement
GetProfileStatus	Retrieves profile measurement status
GetProfilePacketSumIndex	Retrieves the number of profile result packets
RequestProfileData	Retrieves profile result
Structure	Description
ProfileParameter	Profile measurement parameter
ProfileData	Profile data
RotationCenter (C++)	Parameters for rotation center shift
Enumeration	Description
ErrorCode	Error code
Status	Profile Measurement Status

### 3.7. AngleAdjustment Class

Method	Description
AngleAdjustment	Constructor
Start	Start angle adjustment
Stop	Stop angle adjustment
GetStatus	Retrieves angle adjustment status
GetErrorAxisID	Retrieves error Axis ID
GetAdjustingStatus	Retrieves search status during angle adjustment
GetProfilePacketSumIndex	Retrieves the number of angle adjustment profile packets
RequestProfileData	Retrieves angle adjustment profile
SetParameter	Sets angle adjustment parameter
Structure	Description
AngleAdjustmentParameter	Angle adjustment parameter
ProfileData	Profile data
Enumeration	Description

ErrorCode	Angle adjustment error code
Status	Angle adjustment status
AdjustingStatus	Adjustment status during execution
ProfileDataType	Type of alignment profile

### 3.8. IO Class

Method	Description
IO	Constructor
GetPortState	Retrieves digital input/output On/Off
SetPortState	Sets digital output On/Off
SetPulseOutput	Specify output port On/Off time
GetAnalogValue	Retrieves analog input and output values
SetAnalogValue	Not implemented (reserved)
Enumeration	Description
DigitalIOType	Types of digital input and output
AnalogIOType	Types of analog input and output

### 3.9. PowerMeter Class

Method	Description
PowrMeter	Constructor
GetPower	Gets the power value
GetRange	Gets the current range
GetWavelength	Gets the currently set wavelength
SetRange	Sets the range
SetWavelength	Sets the wavelength
PauseCommunication	Temporarily pauses communication with DA1000/1100
Enumeration	Description
ErrorCode	Error code
AllowedRange	Available range setting

## 4. Command Details

---

### 4.1. Machine-dependent parameters

---

The maximum values for the number of axes, digital input/output, analog input/output, and power meter channels vary depending on the actual connected machine.

In the subsequent sections detailing the commands, parameters dependent on the connected machine will be indicated using the following symbols. Please set the values within the supported range for each actual connected machine.

Item	Symbol
Number of axes	*a
Number of power meter channels	*b
Number of digital inputs	*c
Number of digital outputs	*d
Number of analog inputs	*e
Number of analog outputs	*f



## 4.2. System Class

### 4.2.1. Methods

#### 4.2.1.1. GetInstance

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns an instance of the System class	System	System	-	-

This class is a Singleton, so get the class instance with this method.

#### 4.2.1.2. SetAddress

Argument	Description	C# Type	C++ Type	L Limit	U Limit
addr	Sets the AMS Net ID	string	unsigned char[6]	-	-

Sets the AMS Net ID of the connection destination. In C#, specify it in the form of "123.456.789.012.1.1" as a string type. In C++, pass each numerical value of the address as an array of unsigned char type with 6 elements.

#### 4.2.1.3. Connected(C#) IsConnected(C++)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: connected, false: not connected	bool	bool	-	-

Checks whether a connection has been established with the target.

#### 4.2.1.4. IsError

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: an error occurred, false: no error	bool	bool	-	-

Returns whether an error has occurred on the target device.

#### 4.2.1.5. GetErrorCode

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns the type of the last error that occurred.	ErrorCode	ErrorCode	-	-

Returns the type of error currently occurring or the last error that occurred.

#### 4.2.1.6. IsEmergencyAsserted

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: torque off, false: in normal operation	bool	bool	-	-

Returns whether the torque-off input is enabled.

## 4.3. AxisComponents Class

### 4.3.1. Methods

#### 4.3.1.1. AxisComponents

A constructor. Specifies the axis number of the controller as an argument and creates an instance.

AxisComponents(stageNumber)

Argument	Description	C# Type	C++ Type	L Limit	U Limit
stageNumber	Axis number	uint	unsigned long	1	*a

#### 4.3.1.2. MoveAbsolute

Performs absolute position movement. The argument is a target position with selected position unit for the axis.

ErrorCode MoveAbsolute(targetPosition)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetPosition	Target position [ $\mu$ m or deg]	double	double	-	-

#### 4.3.1.3. MoveRelative

Performs relative position movement. The argument is a target position with selected position unit for the axis.

ErrorCode MoveRelative(distance)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
distance	Target position [ $\mu$ m or deg]	double	double	-	-

#### 4.3.1.4. ReturnOrigin

Returns to the origin.

ErrorCode ReturnOrigin()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.3.1.5. Stop

Stops the axis movement.

ErrorCode Stop()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.3.1.6. TurnOnServo

Turns the servo on.

ErrorCode TurnOnServo()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.3.1.7. TurnOffServo

Turns the servo off.

ErrorCode TurnOffServo()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.3.1.8. ResetError

Resets the servo alarm.

ErrorCode ResetError()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.3.1.9. GetTargetPosition

Retrieves the target position of the axis. The unit of the return value is the selected position unit for the axis.

double GetTargetPosition()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Target position [ $\mu$ m or deg]	double	double	-	-

#### 4.3.1.10. GetActualPosition

Retrieves the current position of the axis. The unit of the return value is the selected position unit for the axis.

double GetActualPosition()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Current position [ $\mu$ m or deg]	double	double	-	-

#### 4.3.1.11. IsServoOn

Retrieves the servo state.

bool IsServoOn()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: ON, false: OFF	bool	bool	false	true

#### 4.3.1.12. IsServoAlarm

Retrieves the servo alarm state.

bool IsServoAlarm()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: Alarm, false: Normal	bool	bool	false	true

#### 4.3.1.13. IsMoving

Retrieves the stage drive state.

bool IsMoving()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: Moving, false: Stopped	bool	bool	false	true

#### 4.3.1.14. IsOriginReturnSucceeded

Retrieves the return to origin state.

bool IsOriginReturnSucceeded()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: Return to origin succeeded, false: Return to origin failed or not performed	bool	bool	false	true

#### 4.3.1.15. IsOriginSignalOn

Retrieves the state of the origin sensor.

bool IsOriginSignalOn()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: ON, false: OFF	bool	bool	false	true

#### 4.3.1.16. IsCwLimitSignalOn

Retrieves the state of the CW limit sensor.

bool IsCwLimitSignalOn()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: ON, false: OFF	bool	bool	false	true

#### 4.3.1.17. IsCcwLimitSignalOn

Retrieves the state of the CCW limit sensor.

bool IsCcwLimitSignalOn()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: ON, false: OFF	bool	bool	false	true

#### 4.3.1.18. IsCwTorqueLimitOn

Retrieves the state of the CW torque limit detection.

bool IsCwTorqueLimitOn ()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: ON, false: OFF	bool	bool	false	true

#### 4.3.1.19. IsCcwTorqueLimitOn

Retrieves the state of the CCW torque limit detection.

bool IsCcwTorqueLimitOn ()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: ON, false: OFF	bool	bool	false	true

#### 4.3.1.20. SetMaxSpeed

Sets the maximum speed. The unit of the return value is the selected speed unit for the axis.

ErrorCode SetMaxSpeed(speed)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
speed	Max speed [um/s or deg/s]	double	double	>0	-

#### 4.3.1.21. SetCwTorqueLimit

Sets the CW torque limit. The unit is %.

ErrorCode SetCwTorqueLimit(cWlimit)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
cWlimit	Torque limit value [%]	double	double	>= 0	<= 100

#### 4.3.1.22. SetCCWTorqueLimit

Sets the CCW torque limit. The unit is %.

ErrorCode SetCCWTorqueLimit(cCWlimit)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
cCWlimit	Torque limit value [%]	double	double	>= 0	<= 100

#### 4.3.1.23. SetAccelRate

Sets the acceleration. The unit of the return value is the selected acceleration unit for the axis.

ErrorCode SetAccelRate(rate)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
rate	Acceleration [um/s2 or deg/s2]	double	double	>0	-

#### 4.3.1.24. SetDecelRate

Sets the deceleration. The unit of the return value is the selected deceleration unit for the axis.

ErrorCode SetDecelRate(rate)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
rate	Deceleration [um/s2 or deg/s2]	double	double	>0	-

#### 4.3.1.25. SetSineMotion

Sets the sine motion to be effective. When enabled, it calculates coordinates as a sine motion goniometric axis.

ErrorCode SetSineMotion(sineMotion)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
sineMotion	true: Sine motion enabled, false: Sine motion disabled	bool	bool	false	true

#### 4.3.1.26. SetFulcrumSpan

Sets the distance between the fulcrums of the sine motion goniometer. It is used when sine motion is enabled.

ErrorCode SetFulcrumSpan(span)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
span	Fulcrums [μm] Effective when SetSineMotion is enabled	double	double	>0	-

#### 4.3.1.27. SetEncoderStepLength

Sets the travel amount per encoder step. Set the value obtained by dividing the stage lead length by the number of steps per rotation according to the motor attached to the selected axis.

When sine motion is enabled, set the travel distance for fulcrum movement. In the case of a Suruga Seiki stage, the internal ball screw lead is 1mm, so use 1mm as the lead value for calculations.

ErrorCode SetEncoderStepLength(length)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
length	Encoder step length [ $\mu$ m or deg]	double	double	>0	-

#### 4.3.1.28. GetErrorCode

Retrieves the error occurrence state.

ErrorCode GetErrorCode()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.3.1.29. GetStatus

Retrieves the current axis state.

Status GetStatus

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns the axis status	Status	Status	-	-

### 4.3.2. Enumeration

#### 4.3.2.1. ErrorCode

Axis Error Code

ErrorCode

Error	Value	Description
None	0	Normal
ServoAlarm	1	Servo alarm ongoing
ServoOff	2	No movement due to servo off
Moving	3	Command rejection due to movement
NotConnected	4	Stage not connected
Limit	5	Stopped by detecting limit sensor
Origin	6	Failed to return to origin
Parameter	7	Inappropriate parameter
Torque	8	Stopped by torque limit

#### 4.3.2.2. Status

Axis status

Status

Status	Value	Description
ServoAlarm	0	Servo alarm state
ServoOff	1	Servo off state
Stopping	2	Under control stop
InPosition	3	Stopped within target position
Moving	4	Moving
NotConnected	5	Stage not connected
CwLimit	6	CW limit detected
CcwLimit	7	CCW limit detected
CwTorqueLimit	8	CW torque limit detected
CcwTorqueLimit	0	CCW torque limit detected



## 4.4. Axis2D Class

### 4.4.1. Methods

#### 4.4.1.1. Axis2D

Constructor. Specifies the axis numbers of the main and sub-stages as arguments.

Axis2D(mainStageNumber, subStageNumber)

Argument	Description	C# Type	C++ Type	L Limit	U Limit
mainStageNumber	Main axis number	ushort	unsigned long	1	*a
subStageNumber	Sub axis number	ushort	unsigned long	1	*a

#### 4.4.1.2. GetAxisNumber

Retrieves axis number.

ushort[] GetAxisNumber()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	軸番号の配列	ushort[]	unsigned long*	-	-

#### 4.4.1.3. MoveAbsolute

Performs absolute position 2-axis interpolation movement.

ErrorCode MoveAbsolute(mainTargetPosition, subTargetPosition)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
mainTargetPosition	Main axis target position [μm or deg]	double	double	-	-
subTargetPosition	Sub axis target position [μm or deg]	double	double	-	-

ErrorCode MoveAbsolute(targetPosition)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	false	true
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetPosition	2D Target position [um or deg], [um or deg]	Point	Point	-	-

#### 4.4.1.4. MoveRelative

Performs relative position 2-axis interpolation movement.

ErrorCode MoveRelative(mainDistance, subDistance)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
mainDistance	Main axis travel distance [μm or deg]	double	double	-	-
subDistance	Sub axis travel distance [μm or deg]	double	double	-	-

#### ErrorCode MoveRelative(distance)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
Targetdistance	2D travel distance [um or deg], [um or deg]	Point	Point	-	-

#### 4.4.1.5. MoveAbsoluteAngle

Moves along a straight line rotated by the angle set with the interpolation angle with respect to the specified absolute position of the main axis.

#### ErrorCode MoveAbsoluteAngle(mainTargetPosition, subAngle)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
mainTargetPosition	Main axis target position [um or deg]	double	double	-	-
subAngle	Interpolation angle [deg]	double	double	-45	45

#### 4.4.1.6. MoveRelativeAngle

Moves along a straight line rotated by the angle set with the interpolation angle with respect to the specified relative position of the main axis.

#### ErrorCode MoveRelativeAngle(mainDistance, subAngle)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
mainDistance	Main axis travel distance [um or deg]	double	double	-	-
subAngle	Interpolation angle [deg]	double	double	-45	45

#### 4.4.1.7. Stop

Stops the 2-axis movement.

#### ErrorCode Stop()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.4.1.8. GetTargetPosition

Retrieves the target position.

#### Point GetTargetPosition()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Target position [um or deg], [um or deg]	Point	Point	-	-

#### 4.4.1.9. GetActualPosition

Retrieves the current position.

Point GetActualPosition()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Current position [ $\mu\text{m}$ or deg], [ $\mu\text{m}$ or deg]	Point	Point	-	-

#### 4.4.1.10. IsMoving

Retrieves the drive status of the 2-axis stage.

bool IsMoving()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: Moving, false: Stopped	bool	bool	false	true

#### 4.4.1.11. SetSpeed

Sets the linear speed of the 2-axis interpolation movement.

ErrorCode SetSpeed(speed)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
speed	Combined 2-axis speed [ $\mu\text{m/s}$ or deg/s]	double	double	>0	-

#### 4.4.1.12. SetAccelRate

Sets the acceleration of the 2-axis interpolation movement.

ErrorCode SetAccelRate(rate)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
rate	Acceleration [ $\mu\text{m/s}^2$ or deg/s <sup>2</sup> ]	double	double	>0	-

#### 4.4.1.13. SetDecelRate

Sets the deceleration of the 2-axis interpolation movement.

ErrorCode SetDecelRate(rate)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
rate	Deceleration [ $\mu\text{m/s}^2$ or deg/s <sup>2</sup> ]	double	double	>0	-

#### 4.4.1.14. SetAxisNumber

Sets the axis number.

ErrorCode SetAxisNumber(mainStageNumber, subStageNumber)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
mainStageNumber	Main axis number	ushort	unsigned long	1	*a
subStageNumber	Sub axis number	ushort	unsigned long	1	*a

#### 4.4.1.15. GetErrorCode

Retrieves the error state.

ErrorCode GetErrorCode()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.4.1.16. GetStatus

Retrieves the current axis state.

Status GetStatus

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns the axis status	Status	Status	-	-

### 4.4.1. Structure

#### 4.4.1.1. Point

2D coordinate value

Point

Axis	Description	C# Type	C++ Type	L Limit	U Limit
X	Main axis (X-axis) value	double	double	-	-
Y	Sub axis (Y-axis) value	double	double	-	-

### 4.4.2. Enumeration

#### 4.4.2.1. ErrorCode

Axis error code

ErrorCode

Common with AxisComponents Class (4.3.2.1. ErrorCode)

#### 4.4.2.2. Status

Axis status

Status

Common with AxisComponents class (4.3.2.2. Status).

## 4.5. Axis3D Class

### 4.5.1. Methods

#### 4.5.1.1. Axis3D

Constructor. Specifies the axis numbers of the main X,Y,Z axis numbers.

Axis3D (mainStageNumberX, mainStageNumberY, mainStageNumberZ)

Argument	Description	C# Type	C++ Type	L Limit	U Limit
mainStageNumberX	Main axis number for X axis	ushort	unsigned long	1	*a
mainStageNumberY	Main axis number for Y axis	ushort	unsigned long	1	*a
mainStageNumberZ	Main axis number for Z axis	ushort	unsigned long	1	*a

#### 4.5.1.2. GetAxisNumber

Retrieves axis numbers for main axes.

ushort[] GetAxisNumber()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Array of axis numbers	ushort[]	unsigned long*	-	-

#### 4.5.1.3. MoveAbsolute

Performs absolute position 3-axis interpolation movement.

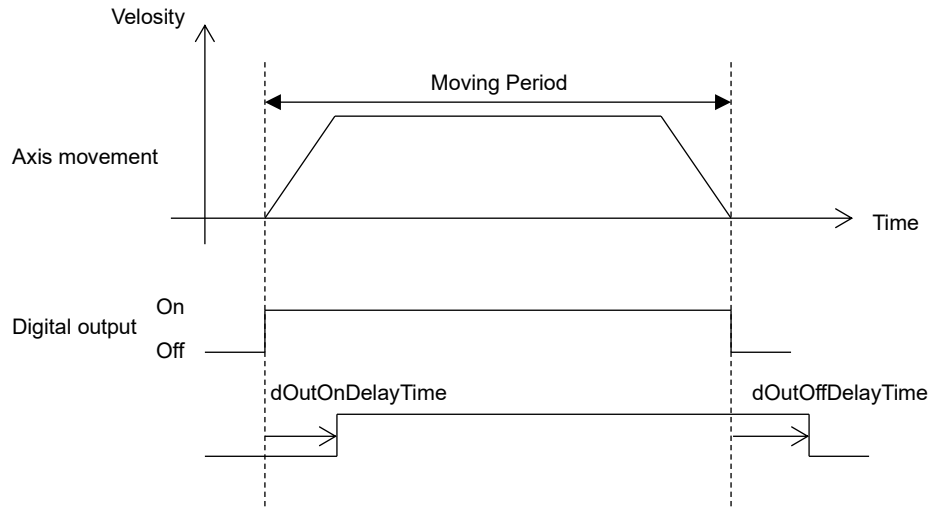
ErrorCode MoveAbsolute(targetPositionX, targetPositionY, targetPositionZ)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetPositionX	X axis target position [μm or deg]	double	double	-	-
targetPositionY	Y axis target position [μm or deg]	double	double	-	-
targetPositionZ	Z axis target position [μm or deg]	double	double	-	-

ErrorCode MoveAbsolute(targetPosition)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetPosition	3D Target position [μm or deg], [μm or deg], [μm or deg]	Point3D	Point3D	-	-

Performs synchronous control of absolute position 3-axis interpolation movement and digital output port.



`ErrorCode MoveAbsolute(targetPositionX, targetPositionY, targetPositionZ, dOutId, dOutOnDelayTime, dOutOffDelayTime)`

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetPositionX	X axis target position [ $\mu\text{m}$ or deg]	double	double	-	-
targetPositionY	Y axis target position [ $\mu\text{m}$ or deg]	double	double	-	-
targetPositionZ	Z axis target position [ $\mu\text{m}$ or deg]	double	double	-	-
dOutId	Digital output port ID	ushort	unsigned short	1	*d
dOutOnDelayTime	Delay time of digital output turned on	short	short	-	-
dOutOffDelayTime	Delay time of digital output turned off	shrot	short	-	-

`ErrorCode MoveAbsolute(targetPosition, dOutId, dOutOnDelayTime, dOutOffDelayTime)`

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetPosition	3D Target position [ $\mu\text{m}$ or deg], [ $\mu\text{m}$ or deg], [ $\mu\text{m}$ or deg]	Point3D	Point3D	-	-
dOutId	Digital output port ID	ushort	unsigned short	1	*d
dOutOnDelayTime	Delay time of digital output turned on	short	short	-	-
dOutOffDelayTime	Delay time of digital output turned off	shrot	short	-	-

#### 4.5.1.4. MoveRelative

Performs relative position 3-axis interpolation movement.

ErrorCode MoveRelative(targetDistanceX, targetDistanceY, targetDistanceZ)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetDistanceX	X axis travel distance [μm or deg]	double	double	-	-
targetDistanceY	Y axis travel distance [μm or deg]	double	double	-	-
targetDistanceZ	Z axis travel distance [μm or deg]	double	double	-	-

ErrorCode MoveRelative(targetDistance)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetDistance	3D Travel distance [μm or deg], [μm or deg], [μm or deg]	Point3D	Point3D	-	-

Performs synchronous control of relative position 3-axis interpolation movement and digital output port.

ErrorCode MoveRelative(targetDistanceX, targetDistanceY, targetDistanceZ, dOutId, dOutOnDelayTime, dOutOffDelayTime)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetDistanceX	X axis travel distance [μm or deg]	double	double	-	-
targetDistanceY	Y axis travel distance [μm or deg]	double	double	-	-
targetDistanceZ	Z axis travel distance [μm or deg]	double	double	-	-
dOutId	Digital output port ID	ushort	unsigned short	1	*d
dOutOnDelayTime	Delay time of digital output turned on	short	short	-	-
dOutOffDelayTime	Delay time of digital output turned off	shrot	short	-	-

ErrorCode MoveRelative(targetPosition, dOutId, dOutOnDelayTime, dOutOffDelayTime)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
targetPosition	3D Travel distance [μm or deg], [μm or deg], [μm or deg]	Point3D	Point3D	-	-
dOutId	Digital output port ID	ushort	unsigned short	1	*d
dOutOnDelayTime	Delay time of digital output turned on	short	short	-	-
dOutOffDelayTime	Delay time of digital output turned off	shrot	short	-	-

#### 4.5.1.5. Stop

Stops the 3-axis movement.

ErrorCode Stop()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.5.1.6. GetTargetPosition

Retrieves the target position.

Point3D GetTargetPosition()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	3D target position [um or deg], [um or deg] , [um or deg]	Point3D	Point3D	-	-

#### 4.5.1.7. GetActualPosition

Retrieves the current position.

Point3D GetActualPosition()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	3D current position [um or deg], [um or deg] , [um or deg]	Point3D	Point3D	-	-



#### 4.5.1.8. IsMoving

Retrieves the drive status of the 3-axis stage.

bool IsMoving()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	true: Moving, false: Stopped	bool	bool	false	true

#### 4.5.1.9. SetSpeed

Sets the linear speed of the 3-axis interpolation movement.

ErrorCode SetSpeed(speed)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
speed	Combined 3-axis speed [ $\mu\text{m/s}$ or $\text{deg/s}$ ]	double	double	>0	-

#### 4.5.1.10. SetAccelRate

Sets the acceleration of the 3-axis interpolation movement.

ErrorCode SetAccelRate(accelRate)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
accelRate	Acceleration [ $\mu\text{m/s}^2$ or $\text{deg/s}^2$ ]	double	double	>0	-

#### 4.5.1.11. SetDecelRate

Sets the deceleration of the 3-axis interpolation movement.

ErrorCode SetDecelRate(decelRate)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
decelRate	Deceleration [ $\mu\text{m/s}^2$ or $\text{deg/s}^2$ ]	double	double	>0	-

#### 4.5.1.12. SetStageNumber

Sets the stage number.

ErrorCode SetStageNumber(mainStageNumberX, mainStageNumberY, mainStageNumberZ)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
mainStageNumberX	Axis number for X axis	ushort	unsigned long	1	*a
mainStageNumberY	Axis number for Y axis	ushort	unsigned long	1	*a
mainStageNumberZ	Axis number for Z axis	ushort	unsigned long	1	*a

#### 4.5.1.13. SetSubStageNumber

Sets the sub-axis number to be used when setting the goniometric axis to the main axis and shifting the center of rotation.

Note: The X, Y, and Z axes within this method may differ from the device coordinate system as they are in the right-handed coordinate system.

Example: When the left-right direction of the actual device is the Z-axis and the axis ID is X=1, Y=2, Z=3, set the Z-axis to subStageNumberY like SetSubStageNumber (1, 3, 2).

ErrorCode SetSubStageNumber(subStageNumberX, subStageNumberY, subStageNumberZ)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
subStageNumberX	X axis number, 0: None	ushort	unsigned long	0	*a
subStageNumberY	Y axis number, 0: None	ushort	unsigned long	0	*a
subStageNumberZ	Z axis number, 0: None	ushort	unsigned long	0	*a

#### 4.5.1.14. SetRotationCenter

Sets the amount of rotation center shift when shifting the rotation center.

Note: The X, Y, and Z axes within this method may differ from the device coordinate system as they are in the right-handed coordinate system.

ErrorCode SetRotationCenter(rotationOffsetX, rotationOffsetY, rotationOffsetZ);

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
rotationOffsetX	Shift amount from X-axis mechanical rotation center [μm]	double	double	-	-
rotationOffsetY	Shift amount from Y-axis mechanical rotation center [μm]	double	double	-	-
rotationOffsetZ	Shift amount from Z-axis mechanical rotation center [μm]	double	double	-	-

The SubStage cannot shift beyond the stroke limit set for the linear axis.

It will stop when the end of the stroke is reached.

#### 4.5.1.15. GetErrorCode

Retrieves the error state.

ErrorCode GetErrorCode()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.5.1.16. GetStatus

Retrieves the axis state.

Status GetStatus

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns axis status	Status	Status	-	-

#### 4.5.2. Structure

##### 4.5.2.1. Point3D

3D coordinate values.

Point3D

Axis	Description	C# Type	C++ Type	L Limit	U Limit
X	X-axis value	double	double	-	-
Y	Y-axis value	double	double	-	-
Z	Z-axis value	double	double	-	-

##### 4.5.2.2. RotationCenter (C#)

Parameters of rotation center shift. Only for C# DLL.

Use this parameter for the rotation center shift function in Alignment or Profile or AngleAdjustment functionality.

RotationCenter

Parameter	Description	C# Type	C++ Type	L Limit	U Limit
enabled	Enables rotation center shift settings true: Enabled, false: Disabled	bool	Bool	false	true
mainStageNumberX	Main axis (rotation axis) X-axis number for rotation center shift	ushort	unsigned long	1	*a
mainStageNumberY	Main axis (rotation axis) Y-axis number for rotation center shift	ushort	unsigned long	1	*a
mainStageNumberZ	Main axis (rotation axis) Z-axis number for rotation center shift	ushort	unsigned long	1	*a
subStageNumberX	Sub-axis (linear axis) X-axis number for rotation center shift, 0: None	ushort	unsigned long	0	*a
subStageNumberY	Sub-axis (linear axis) Y-axis number for rotation center shift, 0: None	ushort	unsigned long	0	*a
subStageNumberZ	Sub-axis (linear axis) Z-axis number for rotation center shift, 0: None	ushort	unsigned long	0	*a
offsetX	Shift amount from X-axis mechanical rotation center [μm]	double	double	-	-
offsetY	Shift amount from Y-axis mechanical rotation center [μm]	double	double	-	-
offsetZ	Shift amount from Z-axis mechanical rotation center [μm]	double	double	-	-

#### 4.5.3. Enumeration

##### 4.5.3.1. ErrorCode

Axis error code

ErrorCode

Common with AxisComponents class (4.3.2.1. ErrorCode).

#### 4.5.3.2. Status

Axis status

Status

Common with AxisComponents class (4.3.2.2. Status).

## 4.6. Alignment Class

---

### 4.6.1. Methods

#### 4.6.1.1. Alignment

Constructor.

Alignment(id)

Argument	Description	C# Type	C++ Type	L Limit	U Limit
id	Instance Id of Alignment	ushort	unsigned long	1	2

Alignment()

Instance ID is incremented by 1 each time an instance is created. The upper and lower limits of Id are the same as Alignment(id).

#### 4.6.1.2. StartSingle

Starts a single-axis alignment. Set the necessary parameters beforehand with SetSingle method.

ErrorCode StartSingle()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.6.1.3. StartFlat

Starts a Flat alignment. Set the necessary parameters beforehand with the SetFlat method.

ErrorCode StartFlat()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.6.1.4. StartFocus

Starts a Focus alignment. Set the necessary parameters beforehand with the SetFocus method.

ErrorCode StartFocus()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.6.1.5. StartTwoChRotation

Starts a 2-Ch rotation alignment. Set the necessary parameters beforehand with the SetTwoChRotation method.

ErrorCode StartTwoChRotation()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.6.1.6. StartMMFRotation

Starts a MMF rotation alignment. Set the necessary parameters beforehand with the SetMMFRotation method.

ErrorCode StartMMFRotation()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.6.1.7. StartLoopBack

Starts a Loop back alignment. Set the necessary parameters beforehand with the SetLoopBack method.

ErrorCode StartLoopBack()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.6.1.8. Stop

Stops the alignment in progress.

ErrorCode Stop()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.6.1.9. GetStatus

Retrieves the alignment status.

Status GetStatus()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Alignment status	Status	Status	-	-

#### 4.6.1.10. GetErrorAxisID

Retrieves the error axis ID during alignment execution.

uint GetErrorAxisID

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Error axis ID number	uint	unsigned long	-	*a

#### 4.6.1.11. GetVoltage

Retrieves the voltage value of the analog input.

double GetVoltage(ch)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Voltage value [V]	double	double	0	10
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Channel of analog input	uint	unsigned long	1	*e

#### 4.6.1.12. GetPower

Retrieves the power value measured by the power meter. Valid only when power meter control connection is established.

double GetPower(ch)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Optical power value [dBm]	double	double	-99.99	99.99
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Power meter Ch, 1: Ch1, 2: Ch2	uint	unsigned long	1	*b

#### 4.6.1.13. GetMeasurementWaveLength

Retrieves the measurement wavelength set on the power meter. Valid only when power meter control connection is established.

double GetMeasurementWaveLength(ch)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Measurement wavelength [nm]	double	double	380.00	1800.00
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Power meter Ch, 1: Ch1, 2: Ch2	uint	unsigned long	1	*b

#### 4.6.1.14. GetAligningStatus

Retrieves the search status during alignment.

AligningStatusCode GetAligningStatus()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Search status during alignment	AligningStatusCode	AligningStatusCode	-	-

#### 4.6.1.15. GetProfilePacketSumNumber

Retrieves the total number of profile packets.

uint GetProfilePacketSumNumber(profileDataType)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Total number of packets, 0: No data	uint	unsigned long	0	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
profileDataType	Profile data type	ProfileDataType	ProfileDataType	-	-

#### 4.6.1.16. RequestProfileData

Retrieves the Profile data of the number entered in index. (Profile data is 1000 data per packet).

ProfileData RequestProfileData(profileDataType ,index)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Profile data	ProfileData	ProfileData	0	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
profileDataType	Profile data type	ProfileDataType	ProfileDataType	-	-
index	Profile request packet number	uint	unsigned long	1	-

Set the necessary profile data type to the argument profileDataType.

It is possible to retrieve whole profile data by retrieving each packet from 1 to max number of profile data packet.

The max number of the packet can be retrieved with GetProfilePacketSumIndex method.

ProfileData RequestProfileData(profileDataType ,index, isRotationCenterShift)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Profile data	ProfileData	ProfileData	0	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
profileDataType	Profile data type	ProfileDataType	ProfileDataType	-	-
index	Profile request packet number	uint	unsigned long	1	-
isRotationCenterShift	Unused. This value has no effect on settings.	bool	bool	false	true

Set the necessary profile data type to the argument profileDataType.

It is possible to retrieve whole profile data by retrieving each packet from 1 to max number of profile data packet.

The max number of the packet can be retrieved with GetProfilePacketSumIndex method.

#### 4.6.1.17. SetSingle

Sets the single-axis alignment parameter.

ErrorCode SetSingle(parameter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Single-axis alignment parameter	SingleParameter	SingleParameter	-	-



#### ErrorCode SetSingle (parameter, rotationCenter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Single-axis alignment parameter	SingleParameter	SingleParameter	-	-
rotationCenter	Rotation center shift parameter	Axis3D.RotationCenter	Alignment.RotationCenter	-	-

When performing alignment using the rotation center shift function, the second argument sets the RotationCenter parameter of the Axis3D class in C# and the RotationCenter parameter of the Alignment class in C++.

#### 4.6.1.18. SetFlat

Sets the Flat alignment parameter.

##### ErrorCode SetFlat(parameter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Flat alignment parameter	FlatParameter	FlatParameter	-	-

##### ErrorCode SetFlat(parameter, rotationCenter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Flat alignment parameter	FlatParameter	FlatParameter	-	-
rotationCenter	Rotation center shift parameter	Axis3D.RotationCenter	Alignment.RotationCenter	-	-

When performing alignment using the rotation center shift function, the second argument sets the RotationCenter parameter of the Axis3D class in C# and the RotationCenter parameter of the Alignment class in C++.

#### 4.6.1.19. SetFocus

Sets the Focus alignment parameter.

##### ErrorCode SetFocus(parameter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Focus alignment parameter	FocusParameter	FocusParameter	-	-

#### ErrorCode SetFocus (parameter, rotationCenter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Focus alignment parameter	FocusParameter	FocusParameter	-	-
rotationCenter	Rotation center shift parameter	Axis3D.RotationCenter	Alignment.RotationCenter	-	-

When performing alignment using the rotation center shift function, the second argument sets the RotationCenter parameter of the Axis3D class in C# and the RotationCenter parameter of the Alignment class in C++.

#### 4.6.1.20. SetTwoChRotation

Sets the Two-Ch rotation alignment parameter.

##### ErrorCode SetTwoChRotation(parameter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Two-Ch rotation alignment parameter	TwoChRotationParameter	TwoChRotationParameter	-	-

#### ErrorCode SetTwoChRotation (parameter, rotationCenter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Two-Ch rotation alignment parameter	TwoChRotationParameter	TwoChRotationParameter	-	-
rotationCenter	Rotation center shift parameter	Axis3D.RotationCenter	Alignment.RotationCenter	-	-

When performing alignment using the rotation center shift function, the second argument sets the RotationCenter parameter of the Axis3D class in C# and the RotationCenter parameter of the Alignment class in C++.

#### 4.6.1.21. SetMMFRotation

Sets the MMF Rotation alignment parameter.

##### ErrorCode SetMMFRotation(parameter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	MMF Rotation alignment parameter	MMFRotationParameter	MMFRotationParameter	-	-

#### ErrorCode SetMMFRotation (parameter, rotationCenter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	MMF Rotation alignment parameter	MMFRotationParameter	MMFRotationParameter	-	-
rotationCenter	Rotation center shift parameter	Axis3D.RotationCenter	Alignment.RotationCenter	-	-

When performing alignment using the rotation center shift function, the second argument sets the RotationCenter parameter of the Axis3D class in C# and the RotationCenter parameter of the Alignment class in C++.

#### 4.6.1.22. SetLoopBack

Sets the Loop back alignment parameter.

#### ErrorCode SetLoopBack(parameter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Loop back alignment parameter	LoopBackParameter	LoopBackParameter	-	-

#### ErrorCode SetLoopBack(parameter, rotationCenter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	LoopBack alignment parameter	LoopBackParameter	LoopBackParameter	-	-
rotationCenter	Rotation center shift parameter	Axis3D.RotationCenter	Alignment.RotationCenter	-	-

When performing alignment using the rotation center shift function, the second argument sets the RotationCenter parameter of the Axis3D class in C# and the RotationCenter parameter of the Alignment class in C++.

#### 4.6.1.23. SetMeasurementWaveLength

Retrieves the measurement wavelength set in the power meter. Only valid when the power meter control is connected.

#### ErrorCode SetMeasurementWaveLength(ch, waveLength)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Power meter Ch	uint	unsigned long	1	*b
waveLength	Wavelength	double	double	380.00	1800.00

## 4.6.2. Structure

### 4.6.2.1. SingleParameter

Single-axis alignment parameter structure.

SingleParameter

Parameter	Description	C# Type	C++ Type	L Limit	U Limit
zMode	Specifies the Z-axis operation for field search	uint	unsigned long	0	2
mainStageNumberX	X-axis alignment axis number	uint	unsigned long	1	*a
subStageNumberX	X-axis interpolation axis number, 0: None Setting the same value as mainStageNumberX is not allowed.	uint	unsigned long	0	*a
subAngleX	X-axis interpolation angle [deg]	double	double	-45	45
pmCh	Power meter Ch number	uint	unsigned long	1	*b
analogCh	Analog input Ch number for alignment	uint	unsigned long	1	*e
pmAutoRangeUpOn	Power meter auto range up, True: On, False: Off	bool	bool	false	true
pmInitRangeSettingOn	Power meter initial range setting, True: On, False: Off	bool	bool	false	true
pmInitRange	Power meter initial range [dBm]: (20,10,0,-10,-20,-30,-40,-50,-60)	int	long	-60	20
fieldSearchThreshold	Field search threshold [V]	double	double	0	10
peakSearchThreshold	Peak search threshold [%]	double	double	0	99.99
searchRangeX	X-axis search range [ $\mu$ m or deg]	double	double	> 0	-
fieldSearchSpeedX	X-axis field search speed [ $\mu$ m/s or deg/s]	double	double	> 0	-
peakSearchSpeedX	X-axis peak search speed [ $\mu$ m/s or deg/s]	double	double	> 0	-
smoothingRangeX	X-axis smoothing range [sample]	uint	unsigned long	0	1000
centroidThresholdX	X-axis centroid alignment threshold [%]	double	double	0	99.99
convergentRangeX	X-axis convergence range [ $\mu$ m or deg]	double	double	0	-
comparisonCount	Number of comparisons	uint	unsigned long	1	99
maxRepeatCount	Max number of alignment repetitions	uint	unsigned long	1	99

#### 4.6.2.2. FlatParameter

Flat alignment parameter structure.

FlatParameter

Parameter	Description	C# Type	C++ Type	L Limit	U Limit
mainStageNumberX	X-axis alignment axis number	uint	unsigned long	1	*a
mainStageNumberY	Y-axis alignment axis number	uint	unsigned long	1	*a
subStageNumberXY	XY-axis interpolation axis number, 0: None Setting the same value as mainStageNumberX and mainStageNumberY is not allowed.	uint	unsigned long	0	*a
subAngleX	X-axis interpolation angle [deg]	double	double	-45	45
subAngleY	Y-axis interpolation angle [deg]	double	double	-45	45
pmCh	Power meter Ch number	uint	unsigned long	1	*b
analogCh	Analog input Ch number for alignment	uint	unsigned long	1	*e
pmAutoRangeUpOn	Power meter auto range up, True: On, False: Off	bool	bool	false	true
pmlnitRangeSettingOn	Power meter initial range setting, True: On, False: Off	bool	bool	false	true
pmlnitRange	Power meter initial range [dBm]: (20,10,0,-10,-20,-30,-40,-50,-60)	int	long	-60	20
fieldSearchThreshold	Field search threshold [V]	double	double	0	10
peakSearchThreshold	Peak search threshold [%]	double	double	0	99.99
searchRangeX	X-axis search range [μm or deg]	double	double	> 0	-
searchRangeY	Y-axis search range [μm or deg]	double	double	> 0	-
fieldSearchPitchX	X-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchPitchY	Y-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchFirstPitchX	X-axis field search initial move amount [μm or deg]	double	double	0	-
fieldSearchSpeedX	X-axis field search speed [μm/s or deg/s]	double	double	> 0	-
fieldSearchSpeedY	Y-axis field search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedX	X-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedY	Y-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
smoothingRangeX	X-axis smoothing range [sample]	uint	unsigned long	0	1000
smoothingRangeY	Y-axis smoothing range [sample]	uint	unsigned long	0	1000
centroidThresholdX	X-axis centroid alignment threshold [%]	double	double	0	99.99
centroidThresholdY	Y-axis centroid alignment threshold [%]	double	double	0	99.99
convergentRangeX	X-axis convergence range [μm or deg]	double	double	0	-
convergentRangeY	Y-axis convergence range [μm or deg]	double	double	0	-
comparisonCount	Number of comparisons	uint	unsigned long	1	99
maxRepeatCount	Max number of alignment repetitions	uint	unsigned long	1	99

#### 4.6.2.3. FocusParameter

Focus alignment parameter structure.

FocusParameter

Parameter	Description	C# Type	C++ Type	L Limit	U Limit
zMode	Specifies the Z-axis operation for field search	uint	unsigned long	0	2
mainStageNumberX	X-axis alignment axis number	uint	unsigned long	1	*a
mainStageNumberY	Y-axis alignment axis number	uint	unsigned long	1	*a
mainStageNumberZ	Z-axis alignment axis number	uint	unsigned long	1	*a
subStageNumberXY	XY-axis interpolation axis number, 0: None Setting the same value as mainStageNumberX and mainStageNumberY is not allowed.	uint	unsigned long	0	*a
subAngleX	X-axis interpolation angle [deg]	double	double	-45	45
subAngleY	Y-axis interpolation angle [deg]	double	double	-45	45
pmCh	Power meter Ch number	uint	unsigned long	1	*b
analogCh	Analog input Ch number for alignment	uint	unsigned long	1	*e
pmAutoRangeUpOn	Power meter auto range up, True: On, False: Off	bool	bool	false	true
pmInitRangeSettingOn	Power meter initial range setting, True: On, False: Off	bool	bool	false	true
pmInitRange	Power meter initial range [dBm]: (20,10,0,-10,-20,-30,-40,-50,-60)	int	long	-60	20
fieldSearchThreshold	Field search threshold [V]	double	double	0	10
peakSearchThresholdXY	XY-axis peak search threshold [%]	double	double	0	99.99
peakSearchThresholdZ	Z-axis peak search threshold [%]	double	double	0	99.99
searchRangeX	X-axis search range [μm or deg]	double	double	> 0	-
searchRangeY	Y-axis search range [μm or deg]	double	double	> 0	-
searchRangeZ	Z-axis search range [μm or deg]	double	double	> 0	-
fieldSearchPitchX	X-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchPitchY	Y-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchPitchZ	Z-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchFirstPitchX	X-axis field search initial move amount [μm or deg]	double	double	0	-
fieldSearchSpeedX	X-axis field search speed [μm/s or deg/s]	double	double	> 0	-
fieldSearchSpeedY	Y-axis field search speed [μm/s or deg/s]	double	double	> 0	-
fieldSearchSpeedZ	Z-axis field search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedX	X-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedY	Y-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedZ	Z-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
smoothingRangeX	X-axis smoothing range [sample]	uint	unsigned long	0	1000
smoothingRangeY	Y-axis smoothing range [sample]	uint	unsigned long	0	1000
smoothingRangeZ	Z-axis smoothing range [sample]	uint	unsigned long	0	1000
centroidThresholdX	X-axis centroid alignment threshold [%]	double	double	0	99.99
centroidThresholdY	Y-axis centroid alignment threshold [%]	double	double	0	99.99
centroidThresholdZ	Z-axis centroid alignment threshold [%]	double	double	0	99.99
convergentRangeX	X-axis convergence range [μm or deg]	double	double	0	-
convergentRangeY	Y-axis convergence range [μm or deg]	double	double	0	-
convergentRangeZ	Z-axis convergence range [μm or deg]	double	double	0	-
comparisonCount	Number of comparisons	uint	unsigned long	1	99
maxRepeatCount	Max number of alignment repetitions	uint	unsigned long	1	99

#### 4.6.2.4. TwoChRotationParameter

Two-Ch Rotation alignment parameter structure.

TwoChRotationParameter

Parameter	Description	C# Type	C++ Type	L Limit	U Limit
mainStageNumberX	X-axis alignment axis number	uint	unsigned long	1	*a
mainStageNumberY	Y-axis alignment axis number	uint	unsigned long	1	*a
subStageNumberXY	XY-axis interpolation axis number, 0: None Setting the same value as mainStageNumberX and mainStageNumberY is not allowed.	uint	unsigned long	0	*a
subAngleX	X-axis interpolation angle [deg]	double	double	-45	45
subAngleY	Y-axis interpolation angle [deg]	double	double	-45	45
rotateStageNumber	Axis number for rotation alignment	uint	unsigned long	1	*a
rotateSpeed	Rotation alignment speed [deg/s]	double	double	> 0	-
rotateDirection	Rotation alignment direction: -1, 1	int	long	-1	1
coreDistance	Core distance between core 1 and 2 [μm]	double	double	> 0	-
pmCh1	Power meter Ch number	uint	unsigned long	1	*b
pmCh2	Power meter Ch number	uint	unsigned long	1	*b
analogCh1	Analog input Ch number for alignment	uint	unsigned long	1	*e
analogCh2	Analog input Ch number for alignment	uint	unsigned long	1	*e
pmAutoRangeUpOn	Power meter auto range up, True: On, False: Off	bool	bool	false	true
pmlnitRangeSettingOn	Power meter initial range setting, True: On, False: Off	bool	bool	false	true
pmlnitRangeCh1	Power meter initial range [dBm]: (20,10,0,- 10,-20,-30,-40,-50,-60)	int	long	-60	20
pmlnitRangeCh2	Power meter initial range [dBm]: (20,10,0,- 10,-20,-30,-40,-50,-60)	int	long	-60	20
fieldSearchThresholdCh1	Field search threshold [V]	double	double	0	10
fieldSearchThresholdCh2	Field search threshold [V]	double	double	0	10
peakSearchThresholdCh1	Peak search threshold [%]	double	double	0	99.99
peakSearchThresholdCh2	Peak search threshold [%]	double	double	0	99.99
searchRangeX	X-axis search range [μm or deg]	double	double	> 0	-
searchRangeY	Y-axis search range [μm or deg]	double	double	> 0	-
fieldSearchPitchX	X-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchPitchY	Y-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchFirstPitchX	X-axis field search initial move amount [μm or deg]	double	double	0	-
fieldSearchSpeedX	X-axis field search speed [μm/s or deg/s]	double	double	> 0	-
fieldSearchSpeedY	Y-axis field search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedX	X-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedY	Y-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
smoothingRangeX	X-axis smoothing range [sample]	uint	unsigned long	0	1000
smoothingRangeY	Y-axis smoothing range [sample]	uint	unsigned long	0	1000
centroidThresholdX	X-axis centroid alignment threshold [%]	double	double	0	99.99
centroidThresholdY	Y-axis centroid alignment threshold [%]	double	double	0	99.99
convergentRangeX	X-axis convergence range [μm or deg]	double	double	0	-
convergentRangeY	Y-axis convergence range [μm or deg]	double	double	0	-
comparisonCount	Number of comparisons	uint	unsigned long	1	99
maxRepeatCount	Max number of alignment repetitions	uint	unsigned long	1	99

#### 4.6.2.5. MMFRotationParameter

MMF Rotation alignment parameter structure.

MMFRotationParameter

Parameter	Description	C# Type	C++ Type	L Limit	U Limit
mainStageNumberX	X-axis alignment axis number	uint	unsigned long	1	*a
mainStageNumberY	Y-axis alignment axis number	uint	unsigned long	1	*a
subStageNumberXY	XY-axis interpolation axis number, 0: None Setting the same value as mainStageNumberX and mainStageNumberY is not allowed.	uint	unsigned long	0	*a
subAngleX	X-axis interpolation angle [deg]	double	double	-45	45
subAngleY	Y-axis interpolation angle [deg]	double	double	-45	45
rotateStageNumber	Axis number for rotation alignment	uint	unsigned long	1	*a
rotateSpeed	Rotation alignment speed [deg/s]	double	double	> 0	-
rotateDirection	Rotation alignment direction: -1, 1	int	long	-1	1
coreDistance	Core distance between core 1 and 2 [μm]	double	double	> 0	-
mmfStageNumber	Axis number for MMF	uint	unsigned long	1	*a
mmfSpeed	Axis speed for MMF [μm/s or deg/s]	double	double	> 0	-
mmfDistance	MMF moving distance [um or deg]	double	double	-	-
pmCh1	Power meter Ch number	uint	unsigned long	1	*b
pmCh2	Power meter Ch number	uint	unsigned long	1	*b
analogCh1	Analog input Ch number for alignment	uint	unsigned long	1	*e
analogCh2	Analog input Ch number for alignment	uint	unsigned long	1	*e
pmAutoRangeUpOn	Power meter auto range up, True: On, False: Off	bool	bool	false	true
pmlnitRangeSettingOn	Power meter initial range setting, True: On, False: Off	bool	bool	false	true
pmlnitRangeCh1	Power meter initial range [dBm]: (20,10,0,-10,-20,-30,-40,-50,-60)	int	long	-60	20
pmlnitRangeCh2	Power meter initial range [dBm]: (20,10,0,-10,-20,-30,-40,-50,-60)	int	long	-60	20
fieldSearchThresholdCh1	Field search threshold [V]	double	double	0	10
fieldSearchThresholdCh2	Field search threshold [V]	double	double	0	10
peakSearchThresholdCh1	Peak search threshold [%]	double	double	0	99.99
peakSearchThresholdCh2	Peak search threshold [%]	double	double	0	99.99
searchRangeX	X-axis search range [μm or deg]	double	double	> 0	-
searchRangeY	Y-axis search range [μm or deg]	double	double	> 0	-
fieldSearchPitchX	X-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchPitchY	Y-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchFirstPitchX	X-axis field search initial move amount [μm or deg]	double	double	0	-
fieldSearchSpeedX	X-axis field search speed [μm/s or deg/s]	double	double	> 0	-
fieldSearchSpeedY	Y-axis field search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedX	X-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedY	Y-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
smoothingRangeX	X-axis smoothing range [sample]	uint	unsigned long	0	1000
smoothingRangeY	Y-axis smoothing range [sample]	uint	unsigned long	0	1000
centroidThresholdX	X-axis centroid alignment threshold [%]	double	double	0	99.99
centroidThresholdY	Y-axis centroid alignment threshold [%]	double	double	0	99.99
convergentRangeX	X-axis convergence range [μm or deg]	double	double	0	-
convergentRangeY	Y-axis convergence range [μm or deg]	double	double	0	-
comparisonCount	Number of comparisons	uint	unsigned long	1	99
maxRepeatCount	Max number of alignment repetitions	uint	unsigned long	1	99



#### 4.6.2.6. LoopBackParameter

Loop back alignment parameter structure.

LoopBackParameter

Parameter	Description	C# Type	C++ Type	L Limit	U Limit
mainStageNumberX	X-axis alignment axis number	uint	unsigned long	1	*a
mainStageNumberY	Y-axis alignment axis number	uint	unsigned long	1	*a
mainStageNumberZ	Tz-axis alignment axis number	uint	unsigned long	1	*a
subStageNumberXY	XY-axis interpolation axis number, 0: None Setting the same value as mainStageNumberX and mainStageNumberY is not allowed.	uint	unsigned long	0	*a
subAngleX	X-axis interpolation angle [deg]	double	double	-45	45
subAngleY	Y-axis interpolation angle [deg]	double	double	-45	45
pmCh	Power meter Ch number	uint	unsigned long	1	*b
analogCh	Analog input Ch number for alignment	uint	unsigned long	1	*e
pmAutoRangeUpOn	Power meter auto range up, True: On, False: Off	bool	bool	false	true
pmlnitRangeSettingOn	Power meter initial range setting, True: On, False: Off	bool	bool	false	true
pmlnitRange	Power meter initial range [dBm]: (20,10,0,-10,-20,-30,-40,-50,-60)	int	long	-60	20
fieldSearchThreshold	Field search threshold [V]	double	double	0	10
peakSearchThresholdXY	XY Peak search threshold [%]	double	double	0	99.99
peakSearchThresholdZ	Tz Peak search threshold [%]	double	double	0	99.99
searchRangeX	X-axis search range [μm or deg]	double	double	> 0	-
searchRangeY	Y-axis search range [μm or deg]	double	double	> 0	-
searchRangeZ	Tz-axis search range [μm or deg]	double	double	> 0	-
fieldSearchPitchX	X-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchPitchY	Y-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchPitchZ	Tz-axis field search pitch [μm or deg]	double	double	> 0	-
fieldSearchFirstPitchX	X-axis field search initial move amount [μm or deg]	double	double	0	-
fieldSearchSpeedX	X-axis field search speed [μm/s or deg/s]	double	double	> 0	-
fieldSearchSpeedY	Y-axis field search speed [μm/s or deg/s]	double	double	> 0	-
fieldSearchSpeedZ	Tz-axis field search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedX	X-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedY	Y-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
peakSearchSpeedZ	Tz-axis peak search speed [μm/s or deg/s]	double	double	> 0	-
smoothingRangeX	X-axis smoothing range [sample]	uint	unsigned long	0	1000
smoothingRangeY	Y-axis smoothing range [sample]	uint	unsigned long	0	1000
smoothingRangeZ	Tz-axis smoothing range [sample]	uint	unsigned long	0	1000
centroidThresholdX	X-axis centroid alignment threshold [%]	double	double	0	99.99
centroidThresholdY	Y-axis centroid alignment threshold [%]	double	double	0	99.99
centroidThresholdZ	Tz-axis centroid alignment threshold [%]	double	double	0	99.99
convergentRangeX	X-axis convergence range [μm or deg]	double	double	0	-
convergentRangeY	Y-axis convergence range [μm or deg]	double	double	0	-
convergentRangeZ	Tz-axis convergence range [μm or deg]	double	double	0	-
comparisonCount	Number of comparisons	uint	unsigned long	1	99
maxRepeatCount	Max number of alignment repetitions	uint	unsigned long	1	99

#### 4.6.2.7. ProfileData

Profile data structure.

ProfileData

Data	Description	C# Type	C++ Type	L Limit	U Limit
mainAxisNumber	Main axis number	ushort	unsigned short	1	*a
sub1AxisNumber	Sub-axis1 number, 0: None	ushort	unsigned short	0	*a
sub2AxisNumber	Sub-axis2 number, 0: None	ushort	unsigned short	0	*a
signalCh1Number	Signal Channel1 axis number	ushort	unsigned short	1	*e
signalCh2Number	Signal Channel2 axis number. 0: None	ushort	unsigned short	0	*e
packetIndex	Index specified when retrieving ProfileData	uint	unsigned long	1	-
dataCount	Number of data	uint	unsigned long	0	1000
mainAxisPositionList	Main axis position data list [μm or deg]	double[1000]	double[1000]	-	-
subAxisPositionList	Sub axis position data list [μm or deg]	double[1000]	double[1000]	-	-
sub2AxisPositionList	Sub axis2 position data list [μm or deg]	double[1000]	double[1000]	-	-
signalCh1List	Signal value [V]	double[1000]	double[1000]	-	-
signalCh2List	Signal value [V]	double[1000]	double[1000]	-	-

#### 4.6.2.8. RotationCenter

Parameters for rotation center shift. Only for C++ DLL.

Same content as 4.5.2.2. RotationCenter in Axis3D class.

### 4.6.3. Enumeration

#### 4.6.3.1. ErrorCode

Alignment error code

Error	Value	Description
None	0	Normal
Axis	1	Error due to axis
Aligning	2	Alignment in progress
Parameter	3	Inappropriate parameters

#### 4.6.3.2. Status

Alignment status

Status

Status	Value	Description
Stopping	0	Stopped
Success	1	Normal termination
Aligning	2	Aligning
FieldSearchRangeOver	3	Field search range over
ProfileDataOver	4	Profile data exceeded
PeakSearchCountOver	5	Did not converge even after executing the maximum number of alignments
PeakSearchRangeOver	6	Peak search range over
InvalidParameter	7	Invalid alignment parameter
ServosNotReady	8	Servo is not in Ready state
ServosAlarm	9	Servo alarm
StageOnLimit	10	Stage reached at limit sensor
VoltageLimit	11	Signal voltage reached at max limit
PMRangeLimit	12	Could not range up due to PM range limit
PMInitRangeChangeFail	13	Power meter initial range setting failed
PMDisconnected	14	Power meter is not connected
RotationAdjustmentFail	15	Rotation adjustment failed
InPositionFail	16	Not reached to in-position state.
TorqueLimit	17	Stopped by torque limit

#### 4.6.3.3. Zmode

Moving behavior for optical axis direction for alignment.

Zmode

Mode	Value	Description
Round	0	Field search while moving forward and backward
ForwardPlus	1	Field search with plus direction
ForwardMinus	2	Field search with minus direction

#### 4.6.3.4. AligningStatusCode

Alignment status code.

AligningStatusCode

Status	Value	Description
NotAligning	0	Not aligning
Initializing	1	Initializing
FieldSearching	2	Field searching
PeakSearchingX	3	X-axis peak searching
PeakSearchingY	4	Y-axis peak searching
PeakSearchingZ	5	Z-axis peak searching
PeakSearchXCh2	6	Ch2 X-axis peak searching Used only in Two-Ch rotation alignment and MMF rotation alignment
PeakSearchYCh2	7	Ch2 Y-axis peak searching Used only in Two-Ch rotation alignment and MMF rotation alignment

#### 4.6.3.5. ProfileDataType

Profile data type.

ProfileDataType

Data type	Value	Description
FieldSearch	0	Field search profile data
PeakSearchX	1	Peak search X-axis profile data
PeakSearchY	2	Peak search Y-axis profile data
PeakSearchZ	3	Peak search Z-axis profile data Used only in Focus alignment
PeakSearchXCh2	4	Peak search X-axis Ch2 side profile data Used only in Two-Ch rotation alignment and MMF rotation alignment
PeakSearchYCh2	5	Peak search X-axis Ch2 side profile data Used only in Two-Ch rotation alignment and MMF rotation alignment

## 4.7. Profile Class

### 4.7.1. Methods

#### 4.7.1.1. Profile

Constructor.

Profile()

#### 4.7.1.2. SetProfile

Sets profile measurement parameter.

ErrorCode SetProfile (parameter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Profile measurement parameter	ProfileParameter	ProfileParameter	-	-

ErrorCode SetProfile (parameter, rotationCenter)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Profile measurement parameter	ProfileParameter	ProfileParameter	-	-
rotationCenter	Rotation center shift parameter	Axis3D.RotationCenter	Profile.RotationCenter	-	-

When performing profile measurement using the rotation center shift function, the second argument sets the RotationCenter parameter of the Axis3D class in C# and the RotationCenter parameter of the Alignment class in C++.

#### 4.7.1.3. Start

Starts profile measurement.

Set the necessary parameters beforehand with SetProfile method.

ErrorCode Start()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.7.1.4. Stop

Stops profile measurement in progress.

ErrorCode Stop()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.7.1.5. GetProfileStatus

Retrieves profile measurement status.

Status GetProfileStatus ()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Profile measurement status	Status	Status	-	-

#### 4.7.1.6. GetProfilePacketSumIndex

Retrieves the number of profile data packets.

uint GetProfilePacketSumIndex()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Total number of packets, 0: No data	uint	unsigned long	0	-

#### 4.7.1.7. RequestProfileData

Retrieves profile data packet specified by index. One profile data packet includes 1,000 data.

ProfileData RequestProfileData(index)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Profile data packet	ProfileData	ProfileData	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
index	Profile data packet index	uint	unsigned long	-	-

It is possible to retrieve whole profile data by retrieving each packet from 1 to max number of profile data packet.

The max number of the packet can be retrieved with GetProfilePacketSumIndex method.

ProfileData RequestProfileData(index, isRotationCenterShift)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Profile data packet	ProfileData	ProfileData	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
index	Profile data packet index	uint	unsigned long	-	-
isRotationCenterShift	Unused. This value has no effect on settings.	bool	bool	false	true

It is possible to retrieve whole profile data by retrieving each packet from 1 to max number of profile data packet.

The max number of the packet can be retrieved with GetProfilePacketSumIndex method.

## 4.7.2. Structure

### 4.7.2.1. ProfileParameter

Profile measurement parameter structure.

Parameter	Description	C# Type	C++ Type	L Limit	U Limit
mainAxisNumber	Main axis number	ushort	unsigned short	1	*a
sub1AxisNumber	Sub-axis1 number, 0: None	ushort	unsigned short	0	*a
sub2AxisNumber	Sub-axis2 number, 0: None	ushort	unsigned short	0	*a
signalCh1Number	Ch1 signal axis number	ushort	unsigned short	1	*e
signalCh2Number	Ch2 signal axis number, 0: None	ushort	unsigned short	0	*e
mainRange	Main axis profile measurement range [ $\mu\text{m}$ or deg]	double	double	0	-
sub1Range	Sub-axis1 profile measurement range [ $\mu\text{m}$ or deg]	double	double	0	-
sub2Range	Sub-axis2 profile measurement range [ $\mu\text{m}$ or deg]	double	double	0	-
speed	Axis speed [ $\mu\text{m/s}$ or deg/s]	double	double	> 0	-
accelRate	Axis acceleration [ $\mu\text{m/s}^2$ or deg/s <sup>2</sup> ]	double	double	> 0	-
decelRate	Axis deceleration [ $\mu\text{m/s}^2$ or deg/s <sup>2</sup> ]	double	double	> 0	-
smoothing	Smoothing range [sample]	uint	unsigned long	0	-

### 4.7.2.2. ProfileData

Profile data packet structure.

Data	Description	C# Type	C++ Type	L Limit	U Limit
mainAxisNumber	Main axis number	ushort	unsigned short	1	*a
sub1AxisNumber	Sub-axis1 number, 0: None	ushort	unsigned short	0	*a
sub2AxisNumber	Sub-axis2 number, 0: None	ushort	unsigned short	0	*a
signalCh1Number	Ch1 signal axis number	ushort	unsigned short	1	*e
signalCh2Number	Ch2 signal axis number, 0: None	ushort	unsigned short	0	*e
packetIndex	Index number of profile data packet	uint	unsigned long	1	-
dataCount	Number of data	uint	unsigned long	0	1000
mainAxisPositionList	Main axis position data list [ $\mu\text{m}$ or deg]	double[1000]	double[1000]	-	-
subAxisPositionList	Sub-axis1 position data list [ $\mu\text{m}$ or deg]	double[1000]	double[1000]	-	-
sub2AxisPositionList	Sub-axis2 position data list [ $\mu\text{m}$ or deg]	double[1000]	double[1000]	-	-
signalCh1List	Signal value list [V]	double[1000]	double[1000]	-	-
signalCh2List	Signal value list [V]	double[1000]	double[1000]	-	-

### 4.7.2.3. RotationCenter

Parameters for rotation center shift. Only for C++ DLL.

Same content as 4.5.2.2. RotationCenter in Axis3D class.

### 4.7.3. Enumeration

#### 4.7.3.1. ErrorCode

Profile measurement error code

Error	Value	Description
None	0	Normal
Axis	1	Error due to axis
Profiling	2	Executing profile measurement
Parameter	3	Inappropriate parameters

#### 4.7.3.2. Status

Profile measurement status

Status	Value	Description
Stopping	0	Stopped
Success	1	Normal termination
Profiling	2	Executing profile measurement
ProfileDataOver	3	Exceeded profile data store range
InvalidParameter	4	Invalid profile measurement parameters
ServosNotReady	5	Servo not in Ready state
ServosAlarm	6	Servo alarm
StageOnLimit	7	Stage reached at limit sensor
TorqueLimit	8	Stopped by torque limit



## 4.8. AngleAdjustment Class

---

Supported: DA1100

### 4.8.1. Methods

#### 4.8.1.1. AngleAdjustment

Constructor

AngleAdjustment(id)

Argument	Description	C# Type	C++ Type	L Limit	U Limit
id	AngleAdjustment instance Id	ushort	unsigned long	1	2

#### 4.8.1.2. Start

Starts angle adjustment. It is necessary to set related parameters by SetParameter methods.

ErrorCode Start ()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.8.1.3. Stop

Stops the angle adjustment process which is performed.

ErrorCode Stop()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-

#### 4.8.1.4. GetStatus

Retrieves the angle adjustment status.

Status GetStatus()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns angle adjustment status	Status	Status	-	-

#### 4.8.1.5. GetErrorAxisID

Retrieves the error axis ID during angle adjustment execution.

uint GetErrorAxisID()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns axis number which is cause of error.	uint	unsigned long	-	*a

#### 4.8.1.6. GetAdjustingStatus

Retrieves the angle adjustment status.

AdjustingStatus GetAdjustingStatus ()

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Status of angle adjustment status which is performed.	AdjustingStatus	AdjustingStatus	-	-

#### 4.8.1.7. GetProfilePacketSumIndex

Retrieves total number of profile packet data.

uint GetProfilePacketSumNumber(profileDataType)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Number of total packets. No data if 0 returned.	uint	unsigned long	0	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
profileDataType	Profile data type	ProfileDataType	ProfileDataType	-	-

#### 4.8.1.8. RequestProfileData

Requests profile data.

Retrieves the profile data which is specified by index. (Single packet of profile data includes 1000 data.).

ProfileData RequestProfileData(profileDataType, index)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Profile data	ProfileData	ProfileData	0	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
profileDataType	Profile data type	ProfileDataType	ProfileDataType	-	-
index	Profile packet index which is requested	uint	unsigned long	1	-

Set the necessary profile data type to the argument profileDataType.

It is possible to retrieve whole profile data by retrieving each packet from 1 to max number of profile data packet.

The max number of the packet can be retrieved with GetProfilePacketSumIndex method.

#### 4.8.1.9. SetParameter

Sets parameters of performing angle adjustment.

ErrorCode SetParameter(parameter, rotationCenter))

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
parameter	Angle adjustment parameter Refer to AngleAdjustmentParameter about detail of the struct.	AngleAdjustmentParameter	AngleAdjustmentParameter	-	-
rotationCenter	Parameter of rotation center shift	Axis3D.RotationCenter	Alignment.RotationCenter	-	-

Set RotationCenter parameter of Axis3D class for C#.

Set RotationCenter parameter of AngleAdjustment class for C++.

## 4.8.2. Structure

### 4.8.2.1. AngleAdjustmentParameter

Parameter of angle adjustment

AngleAdjustmentParameter

Data	Description	C# Type	C++ Type	L Limit	U Limit
gap	Gap distance after angle adjustment	double	double	1	-
signalChNumber	Ch number of using sensor signal	ushort	unsigned short	1	*e
signalLowerLimit	Lower limit of the signal	double	double	-5	20
unlockDOutChNumber	Digital output ch number for unlock control	ushort	unsigned short	0	*d
unlockDOutControlOn	Polarity of digital output for unlock state True: On for unlocking False: Off for unlockng	bool	bool	false	true
lockUnlockAdjustEnable	True: Adjusting position based on locked signal value False: Adjusting position based on unlocked signal value	bool	bool	false	true
lockUnlockDifference	Manual setting of distance between lock and unlock state. Effective when lockUnlockAdjustEnable is true.	double	double	-	-
contactAxisNumber	Axis number for contact detection	ushort	unsigned short	1	*a
contactSearchRange	Searching range of contact detection [μm]	double	double	1	-
contactSearchSpeed	Axis speed of contact detection [μm/sec]	double	double	1	-
contactSmoothing	Number of samples for moving average while contact sensing	uint	unsigned long	0	1000
contactSensitivity	Sensitivity of contact sensing	uint	unsigned long	0	-
pushDistance	Pushing distance after contact detection	double	double	0	-
angleAxisNumberTx	First axis number for angle adjustment	ushort	unsigned short	0	*a
angleAxisNumberTy	Second axis number for angle adjustment	ushort	unsigned short	0	*a
angleSearchRangeTx	Angle search range for first axis [deg]	double	double	0	-
angleSearchRangeTy	Angle search range for second axis adjustment [deg]	double	double	0	-
angleSearchSpeedTx	Angle search speed for first axis [deg/sec]	double	double	0	-
angleSearchSpeedTy	Angle search speed for second axis [deg/sec]	double	double	0	-
angleSmoothingTx	Moving average sample number for first axis angle adjustment	uint	unsigned long	0	1000
angleSmoothingTy	Moving average sample number for second axis angle adjustment	uint	unsigned long	0	1000
angleSensitivityTx	Sensitivity of angle change for first axis	uint	unsigned long	0	-
angleSensitivityTy	Sensitivity of angle change for second axis	uint	unsigned long	0	-
angleJudgeCountTx	Number of count for judging angle change for first axis	ushort	unsigned short	0	99
angleJudgeCountTy	Number of count for judging angle change for second axis	ushort	unsigned short	0	99
angleConvergentRangeTx	Range of convergence of first axis	double	double	0	-
angleConvergentRangeTy	Range of convergence of second axis	double	double	0	-
angleComparisonCount	Number of judging convergence	ushort	unsigned short	0	99
angleMaxCount	Max number of repeating angle adjustment if convergence judgements are failed.	ushort	unsigned short	0	99

#### 4.8.2.2. ProfileData

Profile data

ProfileData

Data	Description	C# Type	C++ Type	L Limit	U Limit
mainAxisNumber	Number of main axis	ushort	unsigned short	1	*a
sub1AxisNumber	Number of sub axis1, 0: None	ushort	unsigned short	0	*a
sub2AxisNumber	Number of sub axis2, 0: None	ushort	unsigned short	0	*a
signalCh1Number	Number of signal for Ch1	ushort	unsigned short	1	*e
signalCh2Number	Number of signal for Ch2, 0: None	ushort	unsigned short	0	*e
packetIndex	Index number of the Profile data	uint	unsigned long	1	-
dataCount	Number of included data	uint	unsigned long	0	1000
mainAxisPositionList	Main axis position [ $\mu$ m or deg]	double[1000]	double[1000]	-	-
subAxisPositionList	Sub axis position [ $\mu$ m or deg]	double[1000]	double[1000]	-	-
sub2AxisPositionList	Sub2 axis position [ $\mu$ m or deg]	double[1000]	double[1000]	-	-
signalCh1List	Signal value of Ch1 [V or mA]	double[1000]	double[1000]	-	-
signalCh2List	Signal value of Ch2 [V or mA]	double[1000]	double[1000]	-	-

#### 4.8.2.3. RotationCenter

Parameter for rotation center shift for C++.

Same as 4.5.2.2.RotationCenter of Axis3D class.

### 4.8.3. Enumeration

#### 4.8.3.1. ErrorCode

Error code of angle adjustment

ErrorCode

Name	Value	Description
None	0	Normal
Axis	1	Error by axis related
Adjusting	2	Executing angle adjustment
Parameter	3	Invalid parameter

#### 4.8.3.2. Status

Status of angle adjustment

Status

Name	Value	Description
Stopping	0	Angle adjustment is stopping normally.
Success	1	Angle adjustment is succeeded.
Adjusting	2	Executing angle adjustment.
ProfileDataOver	3	Exceeded recording range of profile data.
InvalidParameter	4	Invalid parameter
ServosNotReady	5	Using axis is not turned on.
ServosAlarm	6	Servo alarm is occurring for using axis.
StageOnLimit	7	A position limit of using axis is detected.
SignalLowerLimit	8	Signal is reached to lower limit.
CouldnotContact	9	Failed for no contact detection.
AdjustCountOver	10	Failed for exceeding maximum number of retry count.
AngleAdjustRangeOver	11	Failed for exceeding angle range.
LostContact	12	Failed for lost contact while adjustment.

#### 4.8.3.3. AdjustingStatus

Angle adjustment executing status

AdjustingStatus

Name	Value	Description
NotAdjusting	0	Not adjusting.
Initializing	1	Executing initializing process.
ContactingZ	2	Detecting contact.
AdjustingTx	3	Adjusting axis specified by angleAxisNumberTx.
AdjustingTy	4	Adjusting axis specified by angleAxisNumberTy.

#### 4.8.3.4. ProfileDataType

Profile data type for angle adjustment

ProfileDataType

Name	Value	Description
ContactZ	0	Contact detection Z axis profile data
AdjustmentTx	1	Angle adjustment Tx axis profile data
AdjustmentTy	2	Angle adjustment Ty axis profile data

## 4.9. IO Class

---

### 4.9.1. Methods

#### 4.9.1.1. IO

Constructor.

IO()

#### 4.9.1.2. GetPortState

Supported: DA1100

Retrieves digital I/O port On/Off state.

bool GetPortState (iOType, id)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Returns digital I/O port On/Off state	bool	bool	false	true
Argument	Description	C# Type	C++ Type	L Limit	U Limit
iOType	Digital I/O port type	DigitalIOType	DigitalIOType	-	-
id	Digital I/O port number	ushort	unsigned short	1	*c/*d

#### 4.9.1.3. SetPortState

Supported: DA1100

Sets digital output On/Off state.

SetPortState (id, portState)

Argument	Description	C# Type	C++ Type	L Limit	U Limit
id	Digital output port number	ushort	unsigned short	1	*d
portState	Digital output On/Off state	bool	bool	true	false

#### 4.9.1.4. SetPulseOutput

Supported: DA1100

Sets digital output port On/Off state with time period.

SetPortState (id, delayTime, highWidth, lowWidth, repetitionCount)

Argument	Description	C# Type	C++ Type	L Limit	U Limit
id	Digital output port number	ushort	unsigned short	1	*d
delayTime	Wait time before cycle start	ushort	ushort	0	-
highWith	Width of On period	ushort	ushort	0	-
lowWidth	Width of Off period	ushort	Ushort	0	-
repetitionCount	Number of repetition	short	short	0	-

Set IO.RepetitionCountForever (IO::RepetitionCountForever in case of C++) to repetitionCount argument if continue the cycle forever.



#### 4.9.1.5. GetAnalogValue

Retrieves analog input and output values.

double GetAnalogValue (iOType, id)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Analog value	double	double	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
iOType	Types of analog input and output	AnalogIOType	AnalogIOType	-	-
id	Number of analog input and output	ushort	unsigned short	1	*e/*f

DA1000 and DA1100 support AnalogIOType.Input only.

#### 4.9.1.6. SetAnalogValue

Sets analog output value.

SetAnalogValue(value)

Argument	Description	C# Type	C++ Type	L Limit	U Limit
id	Analog output number	ushort	unsigned short	1	*f
value	Output signal level	double	double	-	-

DA1000 and DA1100 do not support this functionality.

### 4.9.2. Enumeration

#### 4.9.2.1. DigitalIOType

Supported: DA1100

Type of digital I/O port

Type	Value	Description
Input	0	Input
Output	1	Output

#### 4.9.2.2. AnalogIOType

Types of analog input and output.

AnalogIOType

Type	Value	Description
Input	0	Input
Output	1	Output

DA1000 and DA1100 do not support output type.

## 4.10. PowerMeter Class

---

### 4.10.1. Methods

#### 4.10.1.1. PowerMeter

Constructor.

PowerMeter()

#### 4.10.1.2. GetPower

Gets the power value from the power meter.

double GetPower(ch)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Power value	double	double	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Power meter channel number	ushort	unsigned short	1	*b

#### 4.10.1.3. GetRange

Gets the current range setting of the power meter.

AllowedRange GetRange(ch)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Current range	AllowedRange	AllowedRange	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Power meter channel number	ushort	unsigned short	1	*b

#### 4.10.1.4. GetWavelength

Gets the currently set wavelength of the power meter.

double GetWavelength(ch)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Currently set wavelength	double	double	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Power meter channel number	ushort	unsigned short	1	*b

#### 4.10.1.5. SetRange

Sets the range of the power meter.

ErrorCode SetRange(ch, range)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Power meter channel number	ushort	unsigned short	1	*b
range	Range to set	AllowedRange	AllowedRange	Note1	Note1

Note :

1. The minimum and maximum values are determined by the specifications of the target power meter.

#### 4.10.1.6. SetWavelength

Sets the wavelength of the power meter.

ErrorCode SetWavelength(ch, wavelength)

	Description	C# Type	C++ Type	L Limit	U Limit
Return value	Error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ Type	L Limit	U Limit
ch	Power meter channel number	ushort	unsigned short	1	*b
Wavelength	Wavelength to set (nm)	double	double	Note1	Note1

Note :

1. The minimum and maximum values are determined by the specifications of the target power meter.

#### 4.10.1.7. PauseCommunication

Temporarily pauses communication between the power meter and the DA1000/DA1100.

ErrorCode PauseCommunication(ch, isEnabled)

	Description	C# Type	C++ 型	L Limit	U Limit
Return value	Error code	ErrorCode	ErrorCode	-	-
Argument	Description	C# Type	C++ 型	L Limit	U Limit
ch	Power meter channel number	ushort	unsigned short	1	*b
isEnabled	true: pause, false: resume	bool	bool	false	true

If multiple power meters are connected in a frame, setting a pause for any one of them will pause communication with all power meters included in the frame.

#### 4.10.2. Enumeration

##### 4.10.2.1. ErrorCode

メンバー	Value	Description
None	0	Normal (no error)
ChannelOutOfRange	1	Specified channel does not exist
Invalid	2	The range specified for getting or setting is AllowedRange Invalid

##### 4.10.2.2. AllowedRange

メンバー	Value	Description
Minus60	-60	-60dBm
Minus50	-50	-50dBm
Minus40	-40	-40dBm
Minus30	-30	-30dBm
Minus20	-20	-20dBm
Minus10	-10	-10dBm
Zero	0	0dBm
Plus10	10	10dBm
Plus20	20	20dBm
Auto	32767	Auto
Invalid	-32768	Any range not listed above (invalid)