

LJ-V7000 Series Communication Library

Reference Manual

Please read this manual before use.

After reading this manual, store it in a safe place where it can be used at any time.

Contents

1	Software License Agreement.....	4
2	Introduction	5
3	Operating Environment	5
4	USB Driver	5
5	File Structure.....	5
6	Incorporating the Library	6
6.1	File structure	6
6.2	Linking	6
6.2.1	C++	6
6.2.2	C#/VB.NET.....	6
7	Types.....	6
8	Constant, Structure Definitions.....	7
8.1	Constant definitions	7
8.2	Structure definitions.....	10
8.3	Callback function interface definition	21
9	Functions.....	24
9.1	Function list.....	24
9.1.1	Operations for the DLL	24
9.1.2	Establish/disconnect the communication path with the controller	24
9.1.3	System control.....	24
9.1.4	Measurement control.....	24
9.1.5	Functions related to modifying or reading settings	25
9.1.6	Acquiring measurement results	25
9.1.7	Storage function related	25
9.1.8	High-speed data communication related.....	26
9.1.9	Supplement	26
9.2	Function reference.....	27
9.2.1	Operations for the DLL	27
9.2.2	Establish/disconnect the communication path with the controller	28
9.2.3	System control.....	28
9.2.4	Measurement control.....	30
9.2.5	Functions related to modifying or reading settings	32
9.2.6	Acquiring measurement results	36
9.2.7	Store function related	40
9.2.8	High-speed data communication related.....	44
9.2.9	Supplement	46
10	Common Return Codes.....	55
10.1	Return codes returned by the communication library	55
10.2	Return codes returned from the controller	55
11	Sample Program	56
11.1	User interface specification.....	56
11.2	Save file format.....	57
12	Appendix.....	58
	Revision History.....	88

1 Software License Agreement

NOTICE TO USER: PLEASE READ THIS SOFTWARE LICENSE AGREEMENT (THIS "AGREEMENT") CAREFULLY. BY USING ALL OR ANY PORTION OF THIS "SOFTWARE", YOU ARE AGREEING TO BE BOUND BY ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO ANY TERMS OF THIS AGREEMENT, DO NOT USE THIS SOFTWARE.

1. Grant of License.

Conditioned upon compliance with all of the terms and conditions of this Agreement, KEYENCE grants you a nonexclusive and nontransferable license.

2. Restrictions

Except for installation of updates or new functions provided by KEYENCE, you may not modify or add any function to this Software.

- a) You may not reverse engineer, decompile or disassemble this Software.
- b) You may not create derivative works based on this Software.
- c) Other than expressly stated by KEYENCE, you may not resell, retransfer, rent or otherwise redistribute this Software to any third parties. However, you may redistribute this Software with the application that you developed using this Software.

3. Intellectual Property Rights.

Except as expressly stated herein, KEYENCE reserves all right, title and interest in this Software, and all associated copyrights, trademarks, and other intellectual property rights therein.

4. Disclaimer.

Keyence is licensing this Software to you "AS IS" and without any warranty of any kind. In no event will KEYENCE or its suppliers be liable to you for any damages, claims, costs or any lost profits caused by using this Software.

5. Support

KEYENCE shall not provide technical support in accordance with this Software including the use of this Software.

6. Termination.

- 6.1 Your license under this Agreement will terminate automatically if you destroy this Software and the copy of this Software in your possession or voluntarily return this Software to us.
- 6.2 Your license under this Agreement will terminate automatically without any notice from KEYENCE if you fail to comply with any of the terms and conditions of this Agreement. Promptly upon termination, you shall cease all use of this Software and destroy all copies, full or partial, of this Software in your possession or control.
- 6.3 You will compensate KEYENCE for costs or any lost profits caused by your violation or breach of any term of this Agreement.

7. Governing Law.

This Agreement will be governed by and construed in accordance with the substantive laws of Japan without regards to the principles of conflicts of law.

2 Introduction

The LJ-V7000 Series communication library provides a communication interface for controlling the LJ-V7000 Series from a user application (Win32 DLL). For specific ways to use the communication library, refer to the sample program.

3 Operating Environment

OS	Windows 7 (Home Premium/Professional/Ultimate) Windows Vista (Home Basic/Home Premium/Business/Ultimate) Windows XP (SP2 or later) (Home Edition/Professional Edition)
CPU	Core i3 2.3 GHz or faster (Core2 Duo 2.8 GHz or faster)
Memory	2 GB or more
Secondary cache memory	2 MB or more
Free drive space	10 GB or more
Interface	A PC equipped with either of the interfaces below. USB 2.0/1.1 ^{*1} , Ethernet 1000BASE-T/100BASE-TX ^{*2}

*1 Operation is not guaranteed with connections via a USB hub

*2 Operation is not guaranteed with connections to a LAN or via a router

3.1 Execution environment

This section describes the necessary environment to execute applications that use the LJ-V7000 Series communication library.

3.1.1 Microsoft C runtime library

The Microsoft C runtime library is required for the DLL to operate.

Run vcredist_x86.exe included on the installation media to install the library.

3.1.2 Microsoft .NET Framework

The Microsoft .NET Framework is required to run the sample application.

Run NetFx20SP2_x86.exe included on the installation media to install the library.

4 USB Driver

Install and use LJ-Navigator2 for the USB driver.

5 File Structure

LJV7_IF.dll	The DLL.
LJV7_IF.lib	The import library for LJV7_IF.dll.
LJV7_ErrorCode.h	The header file that defines the error codes.
LJV7_IF.h	The header file that defines the LJV7_IF.dll interface.
Source	The folder for the sample source code. The source code for the sample program created in C#.

6 Incorporating the Library

6.1 File structure

The files required at execution are listed below.

Place these folders/files in the same folder as the executable file.

- LJV7_IF.dll

6.2 Linking

6.2.1 C++

6.2.1.1 Linking

The library can be linked implicitly or explicitly.

To implicitly link the library, link with "LJV7_IF.lib".

* "LJV7_IF.lib" was built with Visual C++ 2008 SP1.

6.2.1.2 Include files

Include the following header files in the necessary source files.

- LJV7_IF.h
- LJV7_ErrorCode.h

6.2.2 C#/VB.NET

Call each interface using the DllImport attribute.

When passing a structure as an interface argument, specify the StructLayout attribute and pass a structure of the same memory structure as the DLL.

For details, refer to the NativeMethods class (NativeMethods.cs) in the sample.

The processing to call each function has been implemented.

7 Types

In this document, variable types are described according to the following definitions.

CHAR	Signed 8-bit integer
BYTE	Unsigned 8-bit integer
SHORT	Signed 16-bit integer
WORD	Unsigned 16-bit integer
LONG	Signed 32-bit integer
DWORD	Unsigned 32-bit integer
FLOAT	Single precision floating point number (32 bits)
DOUBLE	Double precision floating point number (64 bits)

8 Constant, Structure Definitions

8.1 Constant definitions

Name	Setting value storage level designation
Definition	<pre>typedef enum { LJV7IF_SETTING_DEPTH_WRITE = 0x00, // Write settings area LJV7IF_SETTING_DEPTH_RUNNING = 0x01, // Running settings area LJV7IF_SETTING_DEPTH_SAVE = 0x02 // Save area } LJV7IF_SETTING_DEPTH;</pre>
Description	This enumeration designates the operation target level in functions that modify or read settings. For details on the setting value storage level, refer to the LJ-V7000 Series User's Manual.
Comment	<p>The controller retains three sets of settings data. Those levels are used in the applications below.</p> <p>Write settings area Settings that do not affect operation. In order to not allow an error in controller operations from inconsistencies in settings that occur temporarily when changing multiple settings, the operation of the controller can be changed without causing an error by reflecting the settings from this area to the running settings area after writing the settings to this area.</p> <p>Running settings area The settings the controller is using in its operation. When the controller starts, this area is initialized with the settings in the save area.</p> <p>Save area The settings that are saved even when the controller's power is turned off.</p>

Name	Initialization target setting item designation
Definition	<pre>typedef enum { LJV7IF_INIT_SETTING_TARGET_PRG0 = 0x00, // Program 0 LJV7IF_INIT_SETTING_TARGET_PRG1 = 0x01, // Program 1 LJV7IF_INIT_SETTING_TARGET_PRG2 = 0x02, // Program 2 LJV7IF_INIT_SETTING_TARGET_PRG3 = 0x03, // Program 3 LJV7IF_INIT_SETTING_TARGET_PRG4 = 0x04, // Program 4 LJV7IF_INIT_SETTING_TARGET_PRG5 = 0x05, // Program 5 LJV7IF_INIT_SETTING_TARGET_PRG6 = 0x06, // Program 6 LJV7IF_INIT_SETTING_TARGET_PRG7 = 0x07, // Program 7 LJV7IF_INIT_SETTING_TARGET_PRG8 = 0x08, // Program 8 LJV7IF_INIT_SETTING_TARGET_PRG9 = 0x09, // Program 9 LJV7IF_INIT_SETTING_TARGET_PRG10 = 0x0A, // Program 10 LJV7IF_INIT_SETTING_TARGET_PRG11 = 0x0B, // Program 11 LJV7IF_INIT_SETTING_TARGET_PRG12 = 0x0C, // Program 12 LJV7IF_INIT_SETTING_TARGET_PRG13 = 0x0D, // Program 13 LJV7IF_INIT_SETTING_TARGET_PRG14 = 0x0E, // Program 14 LJV7IF_INIT_SETTING_TARGET_PRG15 = 0x0F, // Program 15 } LJV7IF_INIT_SETTING_TARGET;</pre>
Description	This enumeration designates which settings to initialize in settings initialization function.
Comment	-

Name	Definition that indicates the validity of a measurement value
Definition	<pre> Typedef enum { LJV7IF_MEASURE_DATA_INFO_VALID = 0x00, // Normal measurement data LJV7IF_MEASURE_DATA_INFO_ALARM = 0x01, // Measurement alarm data LJV7IF_MEASURE_DATA_INFO_WAIT = 0x02 // Judgment wait data } LJV7IF_MEASURE_DATA_INFO; </pre>
Description	This enumeration indicates the validity or invalidity of the measurement value.
Comment	-

Name	Definition that indicates the tolerance judgment result of the measurement value
Definition	<pre> Typedef enum { LJV7IF_JUDGE_RESULT_HI = 0x01, // HI LJV7IF_JUDGE_RESULT_GO = 0x02, // GO LJV7IF_JUDGE_RESULT_LO = 0x04 // LO } LJV7IF_JUDGE_RESULT; </pre>
Description	This enumeration indicates the tolerance judgment result for the measurement value in bit units.
Comment	If the measurement value is measurement alarm data, the judgment result is 0x05 (both HI and LO bits are 1).

Name	Get profile target buffer designation
Definition	<pre> Typedef enum { LJV7IF_PROFILE_BANK_ACTIVE = 0x00, // Active surface LJV7IF_PROFILE_BANK_INACTIVE = 0x01 // Inactive surface } LJV7IF_PROFILE_BANK; </pre>
Description	When the memory allocation is "double buffer" in the get profile command, this enumeration designates which surface to get the profiles from.
Comment	-

Name	Get profile position specification method designation (batch measurement: off)
Definition	<pre> Typedef enum { LJV7IF_PROFILE_POS_CURRENT = 0x00, // From current LJV7IF_PROFILE_POS_OLDEST = 0x01, // From oldest LJV7IF_PROFILE_POS_SPEC = 0x02, // Specify position } LJV7IF_PROFILE_POS; </pre>
Description	<p>In the get profile command, this enumeration indicates the specification method for the profiles to get out of the profile data retained in the controller. In get profile, the profiles are stored from oldest to newest.</p> <p>From current Gets the current profiles. The end of the acquired profiles becomes the current profile.</p> <p>From oldest Gets the oldest profile. The head of the acquired profiles becomes the oldest profile.</p> <p>Specify position Gets the specified number of profiles from the specified profile position. The head of the acquired profiles becomes the profiles at the specified position.</p>
Comment	For the specified number of profiles, refer to the individual structure definitions.

Name	Get profile batch data position specification method designation (batch measurement: on)
Definition	<pre> Typedef enum { LJV7IF_BATCH_POS_CURRENT = 0x00, // From current LJV7IF_BATCH_POS_SPEC = 0x02, // Specify position LJV7IF_BATCH_POS_COMMITTED = 0x03, // From current after batch // commitment LJV7IF_BATCH_POS_CURRENT_ONLY = 0x04 // Current only } LJV7IF_BATCH_POS; </pre>
Description	<p>In the get batch profile command, this enumeration indicates the specification method for the profiles to get in what batch out of the batch data retained in the controller. In get profile, the profiles are stored from oldest to newest.</p> <p>From current Gets the profiles in the current batch data.</p> <p>Specify position Gets the profiles in the batch data with the specified number.</p> <p>From current after batch commitment Gets the profiles in the current batch data after commitment.</p> <p>Current only Gets one current profile in the current batch data.</p>
Comment	For the specified number of profiles, refer to the individual structure definitions.

Name	Number of OUT settings
Definition	Const static LONG LJV7IF_OUT_COUNT = 16;
Description	This constant indicates the number of OUT settings.
Comment	-

Name	Number of simultaneously connectable controllers
Definition	Const static LJV7IF_DEVICE_COUNT = 6;
Description	This constant is the upper limit for the number of controllers that can simultaneously communicate.
Comment	-

8.2 Structure definitions

Name	Ethernet settings structure
Definition	Typedef struct { BYTE abyIpAddress[4]; WORD wPortNo; BYTE reserve[2]; } LJV7IF_ETHERNET_CONFIG;
Description	<p>This structure contains the settings passed during an Ethernet communication connection.</p> <p>abyIpAddress The IP address of the controller to connect to. For 192.168.0.1: Set abyIpAddress[0]=192, abyIpAddress[1]=168, and so on.</p> <p>wPortNo(in) The port number of the controller to connect to.</p>
Comment	-

Name	Date/time structure
Definition	Typedef struct { BYTE byYear; BYTE byMonth; BYTE byDay; BYTE byHour; BYTE byMinute; BYTE bySecond; BYTE reserve[2]; } LJV7IF_TIME;
Description	<p>The date/time for the controller.</p> <p>byYear Year. Set from 0 to 99, which means 2000 to 2099. byMonth Month.1 to 12. byDay Day.1 to 31. byHour Hour.0 to 23. byMinute Minute.0 to 59. bySecond Second.0 to 59.</p>
Comment	-

Name	Setting item designation structure
Definition	<pre> Typedef struct { BYTE byType; BYTE byCategory; BYTE byItem; BYTE reserve; BYTE byTarget1; BYTE byTarget2; BYTE byTarget3; BYTE byTarget4; } LJV7IF_TARGET_SETTING; </pre>
Description	<p>byType, byCategory, byItem When modifying or reading a setting, these variables are used to specify the target setting item.</p> <p>byTarget1, byTarget2, byTarget3, byTarget4 These variables are used when specifying further details for the setting item. For example, when configuring OUT measurement mode, these are used to specify the OUT number.</p>
Comment	For details, see the appendix.

Name	Measurement results structure
Definition	<pre> Typedef struct { BYTE byDataInfo; BYTE byJudge; BYTE reserve[2]; FLOAT fValue; } LJV7IF_MEASURE_DATA; </pre>
Description	<p>Measurement value and judgment results.</p> <p>byDataInfo This variable indicates whether or not the measurement value (fValue) is valid, and if it is not a valid value, what kind of data it is. See LJV7IF_MEASURE_DATA_INFO.</p> <p>byJudge Tolerance judgment result. See LJV7IF_JUDGE_RESULT.</p> <p>fValue Measurement value. When not a valid value, a large negative value is stored (-10^{10}).</p>
Comment	-

Name	Profile information structure
Definition	<pre> Typedef struct { BYTE byProfileCnt; BYTE byEnvelope; BYTE reserve[2]; WORD wProfDataCnt; BYTE reserve2[2]; LONG IXStart; LONG IXPitch; } LJV7IF_PROFILE_INFO; </pre>
Description	<p>Information related to the profile.</p> <p>byProfileCnt Whether dicates the amount of profile data stored. (When 2 head/combine (wide) is off, 2 profile data units is stored, otherwise 1 profile data unit is stored.)</p> <p>byEnvelope Whether profile compression (time axis) is on. 0: off, 1: on.</p> <p>wProfDataCnt Profile data count (initial setting: 800).</p> <p>IXStart 1st point X coordinate.</p> <p>IXPitch Profile data X direction interval.</p>
Comment	IXStart and IXPitch are stored in 0.01 μm units.

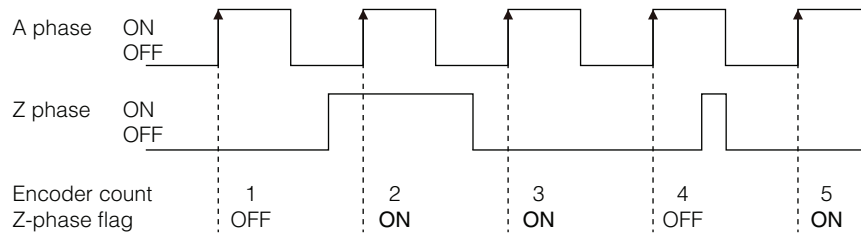
Name	Profile header information structure
Definition	<pre> Typedef struct { DWORD reserve; DWORD dwTriggerCnt; DWORD dwEncoderCnt; DWORD reserve2[3]; } LJV7IF_PROFILE_HEADER; </pre>
Description	<p>The header information added to the profile.</p> <p>reserve 7th bit: Indicates whether the encoder's Z phase has been entered. (*)</p> <p>dwTriggerCnt Indicates which number trigger from the start of measurements this profile is. (Trigger counter)</p> <p>dwEncoderCnt The encoder count when the trigger was issued. (Encoder counter)</p>
Comment	<p>Other than when settings are modified or the program is switched, the trigger counter and the encoder counter are reset at the following times.</p> <ul style="list-style-type: none"> • When the memory is cleared in high-speed mode (profile only) • When laser emission stops and is restarted with the LASER_OFF terminal • When laser emission is allowed after it was prohibited with the REMOTE terminal

*: About the Z-phase flag

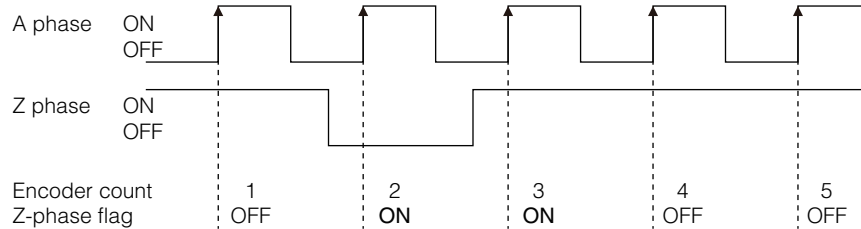
This flag can be used when the controller is version 3.0 or later.

This flag is turned ON when Z-phase ON input is received during the period between the previous trigger input (or the start of measurement if there was no previous trigger input) and the current trigger input.

Example: Single phase 1x multiplier encoder trigger with no skipping



Note: When the Z-phase input uses a negative logic encoder, set the TRG minimum input time, which is a common measurement setting, to 7 μ s. With negative logic, the Z-phase flag turns ON as shown in the following figure.



Name	Profile footer information structure
Definition	<pre> typedef struct { DWORD reserve; } LJV7IF_PROFILE_FOOTER; </pre>
Description	The footer information added to the profile. None (reserved only).
Comment	-
Name	High-speed mode get profile request structure (batch measurement: off)
Definition	<pre> typedef struct { BYTE byTargetBank; BYTE byPosMode; BYTE reserve[2]; DWORD dwGetProfNo; BYTE byGetProfCnt; BYTE byErase; BYTE reserve[2]; } LJV7IF_GET_PROFILE_REQ; </pre>
Description	<p>The get profile designation information when the operation mode is "high-speed (profile only)" and batch measurements are off in the get profile command.</p> <p>byTargetBank Specifies whether to get the profiles from the active surface or whether to get the profiles from the inactive surface. See LJV7IF_PROFILE_BANK.</p> <p>byPosMode Specifies the get profile position specification method. See LJV7IF_PROFILE_POS.</p> <p>dwGetProfNo When byPosMode is LJV7IF_PROFILE_POS_SPEC, specifies the profile number for the profile to get.</p> <p>byGetProfCnt The number of profiles to read.</p> <p>byErase Specifies whether or not to erase the profile data that was read and the profile data older than that. 0: Do not erase, 1: erase</p>
Comment	If the communication buffer is insufficient, the number of profiles specified by byGetProfCnt may not be acquired. In this situation, the maximum number of profiles that can be acquired is returned.

Name	High-speed mode get profile request structure (batch measurement: on)
Definition	<pre> Typedef struct { BYTE byTargetBank; BYTE byPosMode; BYTE reserve[2]; DWORD dwGetBatchNo; DWORD dwGetProfNo; BYTE byGetProfCnt; BYTE byErase; BYTE reserve[2]; } LJV7IF_GET_BATCH_PROFILE_REQ; </pre>
Description	<p>The get profile designation information when the operation mode is "high-speed (profile only)" and batch measurements are on in the get profile command.</p> <p>byTargetBank Specifies whether to get the profiles from the active surface or whether to get the profiles from the inactive surface. See LJV7IF_PROFILE_BANK.</p> <p>byPosMode Specifies the get profile position specification method. See LJV7IF_BATCH_POS.</p> <p>dwGetBatchNo When byPosMode is LJV7IF_BATCH_POS_SPEC, specifies the batch number for the profile to get.</p> <p>dwGetProfNo Specifies the profile number to start getting profiles from in the specified batch number.</p> <p>byGetProfCnt The number of profiles to read.</p> <p>byErase Specifies whether or not to erase the batch data that was read and the batch data older than that. 0: Do not erase, 1: erase</p>
Comment	If the communication buffer is insufficient, the number of profiles specified by byGetProfCnt may not be acquired. In this situation, the maximum number of profiles that can be acquired is returned.

Name	Advanced mode get profile request structure (batch measurement: on)
Definition	<pre> Typedef struct { BYTE byPosMode; BYTE reserve[3]; DWORD dwGetBatchNo; DWORD dwGetProfNo; BYTE byGetProfCnt; BYTE reserve[3]; } LJV7IF_GET_BATCH_PROFILE_ADVANCE_REQ; </pre>
Description	<p>The get profile designation information when the operation mode is "advanced (with OUT measurement)" and batch measurements are on in the get batch profile command.</p> <p>byPosMode Specifies the get profile position specification method. See LJV7IF_BATCH_POS.</p> <p>dwGetBatchNo When byPosMode is LJV7IF_BATCH_POS_SPEC, specifies the batch number for the profiles to get.</p> <p>dwGetProfNo Specifies the profile number for the profiles to get.</p> <p>byGetProfCnt The number of profiles to read.</p>
Comment	If the communication buffer is insufficient, the number of profiles specified by byGetProfCnt may not be acquired. In this situation, the maximum number of profiles that can be acquired is returned.

Name	High-speed mode get profile response structure (batch measurement: off)
Definition	<pre> Typedef struct { DWORD dwCurrentProfNo; DWORD dwOldestProfNo; DWORD dwGetTopProfNo; BYTE byGetProfCnt; BYTE reserve[3]; } LJV7IF_GET_PROFILE_RSP; </pre>
Description	<p>The profile information returned for the get profiles command when the operation mode is "high-speed (profile only)" and batch measurements are off.</p> <p>dwCurrentProfNo The profile number at the current point in time.</p> <p>dwOldestProfNo The profile number for the oldest profile held by the controller.</p> <p>dwGetTopProfNo The profile number for the oldest profile out of those that were read this time.</p> <p>byGetProfCnt The number of profiles that were read this time.</p>
Comment	-

Name	High-speed mode get profile response structure (batch measurement: on)
Definition	<pre> Typedef struct { DWORD dwCurrentBatchNo; DWORD dwCurrentBatchProfCnt; DWORD dwOldestBatchNo; DWORD dwOldestBatchProfCnt; DWORD dwGetBatchNo; DWORD dwGetBatchProfCnt; DWORD dwGetBatchTopProfNo; BYTE byGetProfCnt; BYTE byCurrentBatchCommitted; BYTE reserve[2]; } LJ7IF_GET_BATCH_PROFILE_RSP; </pre>
Description	<p>The profile information returned for the get profiles command when the operation mode is "high-speed (profile only)" and batch measurements are on.</p> <p>dwCurrentBatchNo The batch number at the current point in time.</p> <p>dwCurrentBatchProfCnt The number of profiles in the newest batch.</p> <p>dwOldestBatchNo The batch number for the oldest batch held by the controller.</p> <p>dwOldestBatchProfCnt The number of profiles in the oldest batch held by the controller.</p> <p>dwGetBatchNo The batch number that was read this time.</p> <p>dwGetBatchProfCnt The number of profiles in the batch that was read this time.</p> <p>dwGetBatchTopProfNo Indicates what number profile in the batch is the oldest profile out of the profiles that were read this time.</p> <p>byGetProfCnt The number of profiles that were read this time.</p> <p>byCurrentBatchCommitted Indicates if the batch measurements for the newest batch number has finished. 0: Not finished, 1: finished</p>
Comment	-

Name	Advanced mode get profile response structure (batch measurement: on)
Definition	<pre> Typedef struct { DWORD dwGetBatchNo; DWORD dwGetBatchProfCnt; DWORD dwGetBatchTopProfNo; BYTE byGetProfCnt; BYTE reserve[3]; } LJV7IF_GET_BATCH_PROFILE_ADVANCE_RSP; </pre>
Description	<p>The profile information returned for the get profiles command when the operation mode is "advanced mode (with OUT measurement)" and batch measurements are on.</p> <p>dwGetBatchNo The batch number that was read this time.</p> <p>dwGetBatchProfCnt The number of profiles in the batch that was read this time.</p> <p>dwGetBatchTopProfNo Indicates what number profile in the batch is the oldest profile out of the profiles that were read this time.</p> <p>byReadProfCnt The number of profiles that were read this time.</p>
Comment	-

Name	Get storage status request structure
Definition	<pre> Typedef struct { DWORD dwReadArea; } LJV7IF_GET_STORAGE_STATUS_REQ; </pre>
Description	<p>Get target designation information in the get storage status command.</p> <p>dwReadArea The target surface to read. 0: Active surface 1: When the memory allocation setting is "double buffer", surface A When the memory allocation setting is "entire area (overwrite)", fixed as 1 2: When the memory allocation setting is "double buffer", surface B</p>
Comment	-

Name	Get storage status response structure
Definition	<pre> Typedef struct { DWORD dwSurfaceCnt; DWORD dwActiveSurface; } LJV7IF_GET_STORAGE_STATUS_RSP; </pre>
Description	<p>The storage status information returned for the get storage status command.</p> <p>dwSurfaceCnt Storage surface count</p> <p>dwActiveSurface The active storage surface. When the active program has storage off, 0.</p>
Comment	-

Name	Storage information structure
Definition	<pre> Typedef struct { BYTE byStatus; BYTE byProgramNo; BYTE byTarget; BYTE reserve[5]; DWORD dwStorageCnt; } LJV7IF_STORAGE_INFO; </pre>
Description	<p>Information related to the storage status.</p> <p>byStatus Storage status. 0: Empty (Takes on this value when the target surface has not operated even once in a program with storage on) 1: Storing (only the active storage surface can be 1) 2: Storage complete</p> <p>byProgramNo The program number for the relevant storage surface.</p> <p>byTarget Storage target. 0: Data storage, 2: profile storage, 3: batch profile storage. However, when batch measurements are on and profile compression (time axis) is on, 2: profile storage is stored.</p> <p>dwStorageCnt Storage count (batch count when batch is on)</p>
Comment	-

Name	Get storage data request structure
Definition	<pre> Typedef struct { BYTE reserve[4]; DWORD dwSurface; DWORD dwStartNo; DWORD dwDataCnt; } LJV7IF_GET_STORAGE_REQ; </pre>
Description	<p>The get data designation information in the get data storage data command and the get profile storage data command.</p> <p>dwSurface Storage surface to read.</p> <p>dwStartNo The data number to start reading.</p> <p>dwDataCnt The number of items to read.</p>
Comment	-

Name	Get batch profile storage request structure
Definition	<pre> Typedef struct { BYTE reserve[4]; DWORD dwSurface; DWORD dwGetBatchNo; DWORD dwGetBatchTopProfNo; BYTE byGetProfCnt; BYTE reserved[3]; } LJV7IF_GET_BATCH_PROFILE_STORAGE_REQ; </pre>
Description	<p>Get data designation information in the get batch storage data command.</p> <p>dwSurface Storage surface to read.</p> <p>dwGetBatchNo Batch number to read.</p> <p>dwGetBatchTopProfNo Specifies from what profile number in the batch to get the data.</p> <p>byGetProfCnt The number of profiles to read.</p>
Comment	-

Name	Get storage data response structure
Definition	<pre> Typedef struct { DWORD dwStartNo; DWORD dwDataCnt; LJV7IF_TIME stBaseTime; } LJV7IF_GET_STORAGE_RSP; </pre>
Description	<p>The get data information returned for the get storage data command and the get profile storage command.</p> <p>dwStartNo The storage number to start reading.</p> <p>dwDataCnt The number of items to read.</p> <p>stBaseTime Base time.</p>
Comment	-

Name	Get batch profile storage response structure
Definition	<pre> Typedef struct { DWORD dwGetBatchNo; DWORD dwGetBatchProfCnt; DWORD dwGetBatchTopProfNo; BYTE byGetProfCnt; BYTE reserve[3]; LJV7IF_TIME stBaseTime; } LJV7IF_GET_BATCH_PROFILE_STORAGE_RSP; </pre>
Description	<p>The get data information returned for the get batch profile storage command.</p> <p>dwGetBatchNo The batch number that was read this time.</p> <p>dwGetBatchProfCnt The number of profiles in the batch that was read this time.</p> <p>dwGetBatchTopProfNo Indicates what number profile in the batch is the oldest profile out of the profiles that were read this time.</p> <p>byGetProfCnt The number of profiles that were read this time.</p> <p>stBaseTime Base time.</p>
Comment	-

Name	High-speed communication prep start request structure
Definition	<pre> Typedef struct { BYTE bySendPos; BYTE reserve[3]; } LJV7IF_HIGH_SPEED_PRE_START_REQ; </pre>
Description	<p>High-speed communication start preparation request command</p> <p>bySendPos Send start position. 0: From previous send complete position (from oldest data if 1st time), 1: From oldest data (reacquire), 2: From next data</p>
Comment	-

8.3 Callback function interface definition

Format	<pre>void (*pCallBack)(const BYTE* pBuffer, const DWORD dwSize, const DWORD dwCount, const DWORD dwNotify, const DWORD dwUser);</pre>
Parameters	<p>pBuffer(in) A pointer to the buffer that stores the profile data. The profile data is stored in this buffer with "LJV7IF_PROFILE_HEADER - signed 32-bit profile data - LJV7IF_PROFILE_FOOTER" as a single unit of profile data, and only the number of profiles that could be acquired (dwCount) are returned.</p> <p>dwSize(in) The size in BYTES per single unit of the profile "LJV7IF_PROFILE_HEADER - signed 32-bit profile data - LJV7IF_PROFILE_FOOTER" contained in pBuffer.</p> <p>dwCount(in) The number of profiles stored in pBuffer.</p> <p>dwNotify(in) Notification of an interruption in high-speed communication or a break in batch measurements. For details, see "8.3.1 Supplement".</p> <p>dwUser(in) User information set when high-speed communication was initialized.</p>
Return value	None
Explanation	<p>When using the high-speed communication function, this callback function is called when data is received and when there is a change in the communication state.</p> <p>This callback function is called from a thread other than the main thread.</p> <p>Take care to implement the callback function as thread safe as required.</p> <p>Profile data is stored in 0.01 μm units.</p>

8.3.1 Supplement

8.3.1.1 dwNotify parameter

This section describes the dwNotify parameter used in the callback function.

In high-speed communication, the callback function is called when any number of events occur, in addition to when profile data is received. These events can be checked with the dwNotify parameter.

dwNotify = 0: Indicates profile data was received.

In all other situations, the bits are set according to the following table.

○: May be returned.

×: Will not be returned.

			Batch off	Batch on
LSB	0	Continuous send was stopped (stop by command)	○	○
	1	Continuous send was stopped (automatic stop) ^{*1}	○	○
	2	Continuous send was stopped (automatic stop) ^{*2}	○	○
	3	Continuous send was stopped (automatic stop) ^{*3}	○	○
	4	Reserved		
	5	Reserved		
	6	Reserved		
	7	Reserved		
	8	Send interrupted by clear memory	○	○
	9	Reserved		
	10	Reserved		
	11	Reserved		
	12	Reserved		
	13	Reserved		
	14	Reserved		
	15	Reserved		
	16	Finished sending the batch measurement amount of data ^{*4}	○	○
	17	Reserved		
	18	Reserved		
	19	Reserved		
	20	Reserved		
	21	Reserved		
	22	Reserved		
	23	Reserved		
	24	Reserved		
	25	Reserved		
	26	Reserved		
	27	Reserved		
	28	Reserved		
	29	Reserved		
	30	Reserved		
MSB	31	Finished sending the batch measurement amount of data ^{*4}	×	○

*1 The setting was modified

*2 The program was switched

*3 The profile was not updated for 30 seconds from the start of transmission (timeout)

*4 When this bit is on, the active profile data count in that batch is stored in bits 0 to 30

Bit 0 to 3 and bit 8 indicate that continuous send was stopped.

To restart continuous send, terminate high-speed data communication (LJV7IF_HighSpeedDataCommunicationFinalize), and then redo the send from the initialization of high-speed data communication for each path (LJV7IF_HighSpeedDataUsbCommunicationInitalize/LJV7IF_HighSpeedDataEthernetCommunicationInitalize).

Bit 16 is only valid when batch measurements are on.

When batch measurements are on, the batch measurements can be ended even when the configured batch count is not fulfilled. Therefore, the callback function is notified with this bit on in order to determine the break in batch data.

9 Functions

9.1 Function list

For the notes indicated by "*" in the tables, see "9.1.9 Supplement".

9.1.1 Operations for the DLL

These functions are processed normally even when the controller is in the system error state.

Function name	Overview
LJV7IF_Initialize	Initializes the DLL
LJV7IF_Finalize	Performs the termination processing for the DLL
LJV7IF_GetVersion	Gets the DLL version

9.1.2 Establish/disconnect the communication path with the controller

These functions are processed normally even when the controller is in the system error state.

Function name	Overview
LJV7IF_UsbOpen	Establishes a USB connection
LJV7IF_EthernetOpen	Establishes an Ethernet connection
LJV7IF_CommClose	Disconnects the connection (both USB and Ethernet)

9.1.3 System control

Excluding LJV7IF_ReRunToFactorySetting, these functions are processed normally even when the controller is in the system error state. LJV7IF_ReRunToFactorySetting may fail in the system error state (when a head is not connected, etc.).

Function name	Overview
LJV7IF_RebootController	Reboots the controller
LJV7IF_ReRunToFactorySetting	Returns the controller to the factory settings
LJV7IF_GetError	Gets the controller system error information
LJV7IF_ClearError	Clears the controller system error

9.1.4 Measurement control

Processing for these functions fails when the controller is in the system error state.

Function name	Overview
LJV7IF_Trigger	Issues a trigger
LJV7IF_StartMeasure	Starts measurements
LJV7IF_StopMeasure	Stops measurements
LJV7IF_AutoZero	Issues auto zero
LJV7IF_Timing	Issues timing
LJV7IF_Reset	Issues a reset
LJV7IF_ClearMemory	Clears the internal memory

9.1.5 Functions related to modifying or reading settings

Processing for these functions fails when the controller is in the system error state.

Function name	Overview
LJV7IF_SetSetting	Sends a setting to the controller
LJV7IF_GetSetting	Gets a setting from the controller
LJV7IF_InitializeSetting	Initializes a controller setting
LJV7IF_ReflectSetting	Reflects the contents of the write settings area in the running settings area and the save area
LJV7IF_RewriteTemporarySetting	Overwrites the contents of the write settings area with the settings in the running settings area and the save area
LJV7IF_CheckMemoryAccess	Checks whether or not settings are being saved to the save area
LJV7IF_SetTime	Sets the date/time for the controller
LJV7IF_GetTime	Gets the date/time for the controller
LJV7IF_ChangeActiveProgram	Changes the active program number
LJV7IF_GetActiveProgram	Gets the active program number

9.1.6 Acquiring measurement results

Processing for these functions fails when the controller is in the system error state.

Function name	Overview
LJV7IF_GetMeasurementValue	Gets measurement values
LJV7IF_GetProfile	Gets profiles when the operation mode is "high-speed (profile only)" (*1)
LJV7IF_GetBatchProfile	Gets profiles when the operation mode is "high-speed (profile only)" (*1)
LJV7IF_GetProfileAdvance	Gets profiles when the operation mode is "advanced (with OUT measurement)" (*2)
LJV7IF_GetBatchProfileAdvance	Gets profiles when the operation mode is "advanced (with OUT measurement)" (*2)

9.1.7 Storage function related

Processing for these functions fails when the controller is in the system error state.

Function name	Overview
LJV7IF_StartStorage	Starts storage
LJV7IF_StopStorage	Stops storage
LJV7IF_GetStorageStatus	Gets the storage status
LJV7IF_GetStorageData	Gets the stored data when the storage target is "OUT value"
LJV7IF_GetStorageProfile	Gets the stored profiles when the storage target is "Profiles" (*3)
LJV7IF_GetStorageBatchProfile	Gets the stored profiles when the storage target is "Profiles" (*3)

9.1.8 High-speed data communication related

Processing for these functions fails when the controller is in the system error state.

Function name	Overview
LJV7IF_HighSpeedDataUSBCommunicationInitalize	Performs the initialization required for high-speed data communication (USB)
LJV7IF_HighSpeedDataEthernetCommunicationInitalize	Performs the initialization required for high-speed data communication (Ethernet)
LJV7IF_PreStartHighSpeedDataCommunication	Requests preparation before starting high-speed data communication
LJV7IF_StartHighSpeedDataCommunication	Starts high-speed data communication
LJV7IF_StopHighSpeedDataCommunication	Stops high-speed data communication
LJV7IF_HighSpeedDataCommunicationFinalize	Performs high-speed data communication termination processing

9.1.9 Supplement

*1: How to use LJV7IF_GetProfile and LJV7IF_GetBatchProfile

The interface to use differs according to the combination of controller settings below.

For details, see the diagram below.

- "Batch measurements"
- "Profile compression (time axis)"

	Compression (time axis): off	Compression (time axis): on
Batch measurements: off	LJV7IF_GetProfile	LJV7IF_GetProfile
Batch measurements: on	LJV7IF_GetBatchProfile	LJV7IF_GetProfile

*2: How to use LJV7IF_GetProfileAdvance and LJV7IF_GetBatchProfileAdvance

The interface to use differs according to the combination of controller settings below.

For details, see the diagram below.

- "Batch measurements"
- "Profile compression (time axis)"

	Compression (time axis): off	Compression (time axis): on
Batch measurements: off	LJV7IF_GetProfileAdvance	LJV7IF_GetProfileAdvance
Batch measurements: on	LJV7IF_GetBatchProfileAdvance	LJV7IF_GetProfileAdvance

*3: How to use LJV7IF_GetStorageProfile and LJV7IF_GetStorageBatchProfile

The interface to use differs according to the combination of controller settings below.

For details, see the diagram below.

- "Batch measurements"
- "Profile compression (time axis)"

	Compression (time axis): off	Compression (time axis): on
Batch measurements: off	LJV7IF_GetStorageProfile	LJV7IF_GetStorageProfile
Batch measurements: on	LJV7IF_GetStorageBatchProfile	LJV7IF_GetStorageProfile

9.2 Function reference

The type of the return value for the functions where there is a possibility of an error occurring is LONG. Normally, 0 (ERR_NONE) is returned, and the return code is expressed in the lower 2 bytes (the upper 2 bytes are reserved).

For the common return codes for functions, see "10 Common Return Codes". For the individual return codes for functions, see the function description in this chapter. The return codes are listed as the lower 2 bytes in hexadecimal (example: 0x0100).

9.2.1 Operations for the DLL

■ Initialize DLL

Format	LONG LJV7IF_Initialize(void);
Parameters	-
Return value	No individual return code
Explanation	This function initializes the DLL. (Always run this function)
Supported version	1.00

■ Finalize DLL

Format	LONG LJV7IF_Finalize(void);
Parameters	-
Return value	No individual return code
Explanation	This function performs the termination processing for the DLL. (Always run this function)
Supported version	1.00

■ Get DLL version

Format	DWORD LJV7IF_GetVersion(void);
Parameters	-
Return value	DLL version
Explanation	<p>This function gets the DLL version. The version is expressed as a hexadecimal number. Viewed as hexadecimal, the 4th digit is the major version, the 3rd digit is the minor version, the 2nd digit is the revision, and the 1st digit is the build. For example, the initial version (1.2.3.4) is expressed as 0x1234.</p> <p>The major version is incremented when the DLL's backward compatibility is lost. The minor revision is incremented when the version is updated with additional functions.</p>
Supported version	1.00

9.2.2 Establish/disconnect the communication path with the controller

For communication devices, see "9.2.9.1 Communication devices".

■ USB communication connection

Format	LONG LJV7IF_UsbOpen(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	No individual return code
Explanation	This function establishes a connection with the controller so that the library can communicate with a USB-connected controller.
Supported version	1.00

■ Ethernet communication connection

Format	LONG LJV7IF_EthernetOpen (LONG IDeviceId, LJV7IF_ETHERNET_CONFIG* pEthernetConfig);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pEthernetConfig(in) Ethernet communication settings. For each member, see "8 Constant, Structure Definitions"
Return value	No individual return code
Explanation	This function establishes a connection with the controller so that the library can communicate with an Ethernet-connected controller.
Supported version	1.00

■ Disconnect communication path

Format	LONG LJV7IF_CommClose(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	No individual return code
Explanation	This function closes the USB or Ethernet connection. Even if this function is called when a connection has not been established, an error does not occur.
Supported version	1.00

9.2.3 System control

For communication devices, see "9.2.9.1 Communication devices".

■ Reboot the controller

Format	LONG LJV7IF_RebootController(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	0x80A0: Accessing the save area
Explanation	This function reboots the controller and connected devices. An error occurs while accessing the save area.
Supported version	1.00

■ Return to factory state

Format	LONG LJV7IF_ReRunToFactorySetting(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	No individual return code
Explanation	This function returns all of the controller's settings to the factory state. After processing returns from this interface, write processing is being performed to the save area in the controller. Before turning off the power, ensure that you check the access status to the save area with the LJV7IF_CheckMemoryAccess function (see "9.1.5 Functions related to modifying or reading settings").
Supported version	1.00

■ Get system error information

Format	LONG LJV7IF_GetError (LONG IDeviceId, BYTE byRcvMax, BYTE* pbyErrCnt, WORD* pwErrCode);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. byRcvMax(in) Specifies the maximum amount of system error information to receive. (Size of the buffer passed in pwErrCode) pbyErrCnt(out) The buffer to receive the amount of system error information. pwErrCode(out) The buffer to receive the system error information. In order from the newest error, *pbyErrCnt items (byRcvMax items max) worth of system error information is stored.
Return value	No individual return code
Explanation	This function gets the controller's system error information. For the details of the meanings of the error codes that are returned, refer to the "LJ-V7000 Series User's Manual".
Supported version	1.00

■ Clear system error

Format	LONG LJV7IF_ClearError(LONG IDeviceId, WORD wErrCode);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. wErrCode(in) The error code for the error you wish to clear.
Return value	No individual return code
Explanation	This function clears the system error occurring on the controller. When all of the system errors that are occurring are successfully cleared, the controller will start measurements. Only the errors listed below can be cleared. 0x0084: Two heads were connected when previously started, but only one head could be recognized 0x0085: The connected head type is different than when previously started
Supported version	1.00

9.2.4 Measurement control

For communication devices, see "9.2.9.1 Communication devices".

■ Trigger

Format	LONG LJV7IF_Trigger(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	0x8080: The trigger mode is not "external trigger"
Explanation	This function issues a trigger.
Supported version	1.00

■ Start batch measurements

Format	LONG LJV7IF_StartMeasure(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	0x8080: Batch measurements are off 0x80A0: Batch measurement start processing could not be performed because the REMOTE terminal is off or the LASER_OFF terminal is on
Explanation	This function starts batch measurements. When batch measurements have already been started, nothing happens and there is no error.
Supported version	1.00

■ Stop batch measurements

Format	LONG LJV7IF_StopMeasure(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	0x8080: Batch measurements are off 0x80A0: Batch measurement stop processing could not be performed because the REMOTE terminal is off or the LASER_OFF terminal is on
Explanation	This function stops batch measurements. When batch measurements have not been started, nothing happens and there is no error.
Supported version	1.00

■ Auto zero

Format	LONG LJV7IF_AutoZero(LONG IDeviceId, BYTE byOnOff, DWORD dwOut);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. byOnOff(in) Other than 0: Auto zero on request, 0: off request. dwOut(in) Specifies the OUT to target for processing as a bit. From the LSB, OUT1, OUT2, to OUT16 are indicated by bits, and the OUT is the target for processing when the bit is 1 (upper 16 bits are reserved). Example: When you wish to set OUT1 and OUT5 to be the targets for processing Specify dwOut = 0x00000011 (... 0000 0000 0001 0001).
Return value	0x8080: The operation mode is "high-speed (profile only)"
Explanation	This function issues an auto zero request. Even when the OUT targeted for processing is configured to not be measured, an error will not occur.
Supported version	1.00

■ Timing

Format	LONG LJV7IF_Timing(LONG IDeviceId, BYTE byOnOff, DWORD dwOut);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. byOnOff(in) Same specification method as auto zero (LJV7IF_AutoZero). dwOut(in) Same specification method as auto zero (LJV7IF_AutoZero).
Return value	0x8080: The operation mode is "high-speed (profile only)"
Explanation	This function issues a timing request. Even when the OUT targeted for processing is configured to not be measured, an error will not occur.
Supported version	1.00

■ Reset

Format	LONG LJV7IF_Reset(LONG IDeviceId, DWORD dwOut);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. dwOut(in) Same specification method as auto zero (LJV7IF_AutoZero).
Return value	0x8080: The operation mode is "high-speed (profile only)"
Explanation	This function issues a reset request. Even when the OUT targeted for processing is configured to not be measured, an error will not occur.
Supported version	1.00

■ Clear memory

Format	LONG LJV7IF_ClearMemory(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	No individual return code
Explanation	When the operation mode is "high-speed (profile only)", this function clears the profile data accumulated in internal memory. When the operation mode is "advanced (with OUT measurement)", the accumulated storage data is cleared.
Supported version	1.00

9.2.5 Functions related to modifying or reading settings

For communication devices, see "9.2.9.1 Communication devices".

■ Send setting

Format	LONG LJV7IF_SetSetting(LONG IDeviceId, BYTE byDepth, LJV7IF_TARGET_SETTING TargetSetting, void* pData, DWORD dwDataSize, DWORD* pdwError);
Parameters	IDeviceId(in) Indicates the communication device to communicate with. byDepth(in) Specifies the level to reflect the setting value to that was sent. (LJV7IF_SETTING_DEPTH) TargetSetting(in) Identifies the item that is the target to send. pData(in) Specifies the buffer that stores the setting data to send. dwDataSize(in) The size in BYTES of the setting data to send. pdwError(out) Detailed setting error (see "9.2.9.4 Detailed setting errors"). * For details on the parameters, see the appendix.
Return value	No individual return code
Explanation	This function sends the setting for the specified item to the controller. For the procedure to reflect the setting on the controller, see "9.2.9.3 Write processing for settings".
Supported version	1.00

■ Get setting

Format	LONG LJV7IF_GetSetting(LONG IDeviceId, BYTE byDepth, LJV7IF_TARGET_SETTING TargetSetting, void* pData, DWORD dwDataSize);
Parameters	<p>IDeviceId(in) Specifies the communication device to communicate with.</p> <p>byDepth(in) Specifies the level of the setting value to get. (LJV7IF_SETTING_DEPTH)</p> <p>TargetSetting(in) Identifies the item that is the target to get.</p> <p>pData(out) Specifies the buffer to receive the setting data that was acquired.</p> <p>dwDataSize(in) The size of the buffer to receive the acquired data in BYTES.</p> <p>* For details on the parameters, see the appendix.</p>
Return value	No individual return code
Explanation	This function gets the setting for the specified item from the controller.
Supported version	1.00

■ Initialize setting

Format	LONG LJV7IF_InitializeSetting(LONG IDeviceId, BYTE byDepth, BYTE byTarget);
Parameters	<p>IDeviceId(in) Specifies the communication device to communicate with.</p> <p>byDepth(in) Specifies the level to reflect the initialized setting. (LJV7IF_SETTING_DEPTH)</p> <p>byTarget (in) Specifies the setting that is the target for initialization. (LJV7IF_INIT_SETTING_TARGET)</p>
Return value	No individual return code
Explanation	This function initializes the setting specified as the initialization target. For the procedure to reflect the setting on the controller, see "9.2.9.3 Write processing for settings".
Supported version	1.00

■ Request to reflect settings in the write settings area

Format	LONG LJV7IF_ReflectSetting(LONG IDeviceId, BYTE byDepth, DWORD*pdwError);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. byDepth (in) Specifies to what level the settings written in the write settings area will be reflected to. (LJV7IF_SETTING_DEPTH) pdwError(out) Detailed setting error (see "9.2.9.4 Detailed setting errors").
Return value	No individual return code
Explanation	This function reflects the settings stored in the write settings area to the running settings area. When LJV7IF_SETTING_DEPTH_SAVE is specified as a parameter, the settings in the save area can be saved. When the controller was instructed to overwrite the settings in the save area with this function, ensure that you check the access status to the save area with the LJV7IF_CheckMemoryAccess function before turning the power off.
Supported version	1.00

■ Update write settings area

Format	LONG LJV7IF_RewriteTemporarySetting(LONG IDeviceId, BYTE byDepth);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. byDepth (in) Specifies the level of the settings to update the write settings area with. (LJV7IF_SETTING_DEPTH)
Return value	No individual return code
Explanation	This function updates the contents of the write settings area with either the settings in the running settings area or the settings saved in the save area.
Supported version	1.00

■ Check the status of saving to the save area

Format	LONG LJV7IF_CheckMemoryAccess(LONG IDeviceId, BYTE* pbyBusy);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pbyBusy(out) Other than 0: Accessing the save area, 0: no access.
Return value	No individual return code
Explanation	This function checks whether or not the controller is accessing the save area with an operation such as that to save settings. When the controller was instructed to save settings to the save area with the LJV7IF_RetrurnToFactorySetting function (see "9.1.3 System control"), the LJV7IF_SetSetting function, the LJV7IF_InitializeSetting function, or the LJV7IF_ReflectSetting function, check that access to the save area has completed with this function before turning off the power.
Supported version	1.00

■ Set date/time

Format	LONG LJV7IF_SetTime(LONG IDeviceId, LJV7IF_TIME* pTime);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pTime(in) The date/time to set. For each member, see "8 Constant, Structure Definitions".
Return value	No individual return code
Explanation	This function sets the date/time for the controller.
Supported version	1.00

■ Get date/time

Format	LONG LJV7IF_GetTime(LONG IDeviceId, LJV7IF_TIME* pTime);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pTime(out) The buffer to store the acquired date/time. For each member, see "8 Constant, Structure Definitions".
Return value	No individual return code
Explanation	This function gets the date/time from the controller.
Supported version	1.00

■ Change program

Format	LONG LJV7IF_ChangeActiveProgram(LONG IDeviceId, BYTE byProgNo);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. byProgNo(in) Program number after the change. Specify as 0 to 15 (0: Program 0, 1: Program 1, and so on).
Return value	0x8080: The change program setting is "terminal"
Explanation	This function changes the active program number. When specifying the same number as the active program number in byProgNo, or when an invalid program number is specified, the operation to change the program is performed (internal memory is cleared, etc.), but the active program number is not changed.
Supported version	1.00

■ Get the active program number

Format	LONG LJV7IF_GetActiveProgram(LONG IDeviceId, BYTE* pbyProgNo);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pbyProgNo(out) The buffer to receive the active program number. It is stored as 0 to 15 (0: Program 0, 1: Program 1, and so on).
Return value	No individual return code
Explanation	This function gets the active program number.
Supported version	1.00

9.2.6 Acquiring measurement results

For communication devices, see "9.2.9.1 Communication devices".

■ Get measurement results

Format	LONG LJV7IF_GetMeasurementValue(LONG IDeviceId, LJV7IF_MEASURE_DATA* pMeasureData);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pMeasureData(out) This buffer stores the data for all 16 OUTs including the OUTs that are not measuring. The host requires the passing of a buffer LJV7IF_MEASURE_DATA[16] in size.
Return value	0x8080: The operation mode is "high-speed (profile only)"
Explanation	This function gets the newest measurement results (measurement values and judgment results).
Supported version	1.00

■ Get profiles (operation mode "high-speed (profile only)")

Format	LONG LJV7IF_GetProfile(LONG IDeviceId, LJV7IF_GET_PROFILE_REQ* pReq, LJV7IF_GET_PROFILE_RSP* pRsp, LJV7IF_PROFILE_INFO* pProfileInfo, DWORD* pdwProfileData, DWORD dwDataSize);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pReq(in) Specifies the position, etc., of the profiles to get. For each member, see "8 Constant, Structure Definitions". pRsp(out) Indicates the position, etc., of the profiles that were actually acquired. For each member, see "8 Constant, Structure Definitions". pProfileInfo(out) The profile information for the acquired profiles. For each member, see "8 Constant, Structure Definitions". pdwProfileData (out) The buffer to get the profile data. The profile data is stored in this buffer with "LJV7IF_PROFILE_HEADER - signed 32-bit profile data - LJV7IF_PROFILE_FOOTER" as a single unit of profile data, and only the number of profiles that could be acquired are returned. dwDataSize(in) pdwProfileData size in BYTES
Return value	0x8080: The operation mode is "advanced (with OUT measurement)" 0x8081: "Batch measurements on and profile compression (time axis) off" 0x80A0: No profile data
Explanation	This function gets profile data. Profile data is stored in 0.01 ?m units. For the details on the data stored in the profile data (storage order and size), see "9.2.9 Supplement".
Supported version	1.00

■ Get batch profiles (operation mode "high-speed (profile only)")

Format	LONG LJV7IF_GetBatchProfile(LONG IDDeviceId, LJV7IF_GET_BATCH_PROFILE_REQ* pReq, LJV7IF_GET_BATCH_PROFILE_RSP* pRsp, LJV7IF_PROFILE_INFO * pProfileInfo, DWORD* pdwBatchData, DWORD dwDataSize);
Parameters	<p>IDeviceId(in) Specifies the communication device to communicate with.</p> <p>pReq(in) Specifies the position, etc., of the profiles to get. For each member, see "8 Constant, Structure Definitions".</p> <p>pRsp(out) Indicates the position, etc., of the profiles that were actually acquired. For each member, see "8 Constant, Structure Definitions".</p> <p>pProfileInfo(out) The profile information for the acquired profiles. For each member, see "8 Constant, Structure Definitions".</p> <p>pdwProfileData (out) The buffer to get the profile data. The profile data is stored in this buffer with "LJV7IF_PROFILE_HEADER - signed 32-bit profile data - LJV7IF_PROFILE_FOOTER" as a single unit of profile data, and only the number of profiles that could be acquired are returned.</p> <p>dwDataSize(in) pdwProfileData size in BYTES</p>
Return value	<p>0x8080: The operation mode is "advanced (with OUT measurement)"</p> <p>0x8081: Not "batch measurements on and profile compression (time axis) off"</p> <p>0x80A0: No batch data (batch measurements not run even once)</p>
Explanation	<p>This function gets profile data. Profile data is stored in 0.01 μm units. For the details on the data stored in the profile data (storage order and size), see "9.2.9 Supplement".</p> <p>To read all of the profiles in one batch, read them with the procedure below.</p> <ol style="list-style-type: none"> 1. Call this function by specifying LJV7IF_BATCH_POS_CURRENT for byPosMode in pReq. Save the start position and the amount of profiles that were read and the batch number that was read. 2. Configure pReq as listed below and call this function again. byPosMode = LJV7IF_BATCH_POS_SPEC dwGetBatchNo = batch number that was read byGetProfCnt = starting profile number of the unread profiles in the batch 3. Update byGetProfCnt in step 2 and call this function until all of the profiles in the batch can be read.
Supported version	1.00

■ Get profiles (operation mode "advanced (with OUT measurement)")

Format	LONG LJV7IF_GetProfileAdvance(LONG IDeviceId, LJV7IF_PROFILE_INFO* pProfileInfo, DWORD* pdwProfileData, DWORD dwDataSize, LJV7IF_MEASURE_DATA* pMeasureData);
Parameters	<p>IDeviceId(in) Specifies the communication device to communicate with.</p> <p>pProfileInfo(out) The profile information for the acquired profiles. For each member, see "8 Constant, Structure Definitions".</p> <p>pdwProfileData (out) The buffer to get the profile data. In the "LJV7IF_PROFILE_HEADER - signed 32-bit profile data - LJV7IF_PROFILE_FOOTER" format, one item of profile data is stored.</p> <p>dwDataSize(in) pdwProfileData size in BYTES</p> <p>pMeasureData(out) This buffer stores the data for all 16 OUTs including the OUTs that are not measuring. The host requires the passing of a buffer LJV7IF_MEASURE_DATA[16] in size.</p>
Return value	<p>0x8080: The operation mode is "high-speed (profile only)"</p> <p>0x8081: "Batch measurements on and profile compression (time axis) off"</p> <p>0x80A0: No profile data</p>
Explanation	<p>This function gets profile data.</p> <p>Profile data is stored in 0.01 μm units.</p> <p>For the details on the data stored in the profile data (storage order and size), see "9.2.9 Supplement".</p>
Supported version	1.00

■ Get batch profiles (operation mode "advanced (with OUT measurement)")

Format	LONG LJV7IF_GetBatchProfileAdvance(LONG IDeviceId, LJV7IF_GET_BATCH_PROFILE_ADVANCE_REQ* pReq, LJV7IF_GET_BATCH_PROFILE_ADVANCE_RSP* pRsp, LJV7IF_PROFILE_INFO* pProfileInfo, DWORD* pdwBatchData, DWORD dwDataSize, LJV7IF_MEASURE_DATA* pMeasureData);
Parameters	<p>IDeviceId(in) Specifies the communication device to communicate with.</p> <p>pReq(in) Specifies the position, etc., of the profiles to get. For each member, see "8 Constant, Structure Definitions".</p> <p>pRsp(out) Indicates the position, etc., of the profiles that were actually acquired. For each member, see "8 Constant, Structure Definitions"</p> <p>pProfileInfo(out) The profile information for the acquired profiles. For each member, see "8 Constant, Structure Definitions".</p> <p>dwBatchData(out) The buffer to get the profile data. The profile data is stored in this buffer with "LJV7IF_PROFILE_HEADER - signed 32-bit profile data - LJV7IF_PROFILE_FOOTER - LJV7IF_MEASURE_DATA x 16 OUTs" as a single unit of profile data, and only the number of profiles that could be acquired are returned. The results of the measurement process for the relevant profile are stored in LJV7IF_MEASURE_DATA.</p> <p>dwDataSize(in) pdwProfileData size in BYTES.</p> <p>pBatchMeasureData(out) The measurement results for the batch data that is the target to get. This buffer stores the data for all 16 OUTs including the OUTs that are not measuring.</p> <p>pMeasureData(out) The newest measurement results at the time the command was processed. This buffer stores the data for all 16 OUTs including the OUTs that are not measuring. The host requires the passing of a buffer LJV7IF_MEASURE_DATA[16] in size.</p>
Return value	<p>0x8080: The operation mode is "high-speed (profile only)"</p> <p>0x8081: Not "batch measurements on and profile compression (time axis) off"</p> <p>0x80A0: No batch data (batch measurements not run even once)</p>
Explanation	<p>This function gets profile data. Profile data is stored in 0.01 μm units. For the details on the data stored in the profile data (storage order and size), see "9.2.9 Supplement".</p> <p>To read all of the profiles in one batch, read them with the procedure below.</p> <ol style="list-style-type: none"> 1. Call this function by specifying LJV7IF_BATCH_POS_CURRENT for byPosMode in pReq. Save the start position and the amount of profiles that were read and the batch number that was read. 2. Configure pReq as listed below and call this function again. byPosMode=LJV7IF_BATCH_POS_SPEC dwGetBatchNo = batch number that was read byGetProfCnt = starting profile number of the unread profiles in the batch 3. Update byGetProfCnt in step 2 and call this function until all of the profiles in the batch can be read. <p>When calling this function, do not perform communication between the target controller and LJ-Navigation 2.</p>
Supported version	1.00

9.2.7 Store function related

For communication devices, see "9.2.9.1 Communication devices".

■ Start storage

Format	LONG LJV7IF_StartStorage(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	0x8080: The operation mode is "high-speed (profile only)" 0x8081: Storage target setting is "OFF" (no storage) 0x8082: The storage condition setting is not "terminal/command"
Explanation	When the storage condition setting is terminal/command, this function requests the start of storage.
Supported version	1.00

■ Stop storage

Format	LONG LJV7IF_StopStorage(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	0x8080: The operation mode is "high-speed (profile only)" 0x8081: Storage target setting is "off" (no storage) 0x8082: The storage condition setting is not "terminal/command"
Explanation	When the storage condition setting is terminal/command, this function requests the stop (cancellation) of storage.
Supported version	1.00

■ Get storage status

Format	LONG LJV7IF_GetStorageStatus(LONG IDeviceId, LJV7IF_GET_STORAGE_STATUS_REQ* pReq, LJV7IF_GET_STORAGE_STATUS_RSP* pRsp, LJV7IF_STORAGE_INFO* pStorageInfo);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pReq(in) Specifies the target of the storage status to get. For each member, see "8 Constant, Structure Definitions". pRsp(out) Represents the actually acquired storage status. For each member, see "8 Constant, Structure Definitions". pStorageInfo(out) Storage information destination. If dwActiveSurface in pRsp is 0, the storage status is not updated
Return value	0x8080: The operation mode is "high-speed (profile only)"
Explanation	When the storage target setting is not off, this function gets the storage status.
Supported version	1.00

■ Get data storage data

Format	LONG LJV7IF_GetStorageData(LONG IDeviceId, LJV7IF_GET_STORAGE_REQ* pReq, LJV7IF_STORAGE_INFO* pStorageInfo, LJV7IF_GET_STORAGE_RSP* pRsp, DWORD* pdwData, DWORD dwDataSize);
Parameters	<p>IDeviceId(in) Specifies the communication device to communicate with.</p> <p>pReq(in) Specifies the storage data to get. For each member, see "8 Constant, Structure Definitions".</p> <p>pStorageInfo(out) Represents the actually acquired storage information. For each member, see "8 Constant, Structure Definitions".</p> <p>pRsp(out) Represents the position and size of the actually acquired storage data. For each member, see "8 Constant, Structure Definitions".</p> <p>pdwData(out) The buffer to get the storage data. The storage data is stored in this buffer with "counter value in 10 ms units from the 32-bit base time - LJV7IF_MEASURE_DATA[16]" as a single unit of storage data, and only the amount of storage data that could be acquired is returned. For each member, see "8 Constant, Structure Definitions".</p> <p>dwDataSize(in) pdwData size in BYTES</p>
Return value	<p>0x8080: The operation mode is "high-speed (profile only)"</p> <p>0x8081: The storage target setting is not "OUT value"</p>
Explanation	<p>When the storage target setting is "OUT value", this function gets the storage data.</p> <p>If there is no accumulated data, 0 is returned for the number of items read in pRsp. If the data specified by the data number to start reading in pRep has not been accumulated, one item of the newest data that has been accumulated is returned.</p>
Supported version	1.00

■ Get profile storage data

Format	LONG LJV7IF_GetStorageProfile(LONG IDeviceId, LJV7IF_GET_STORAGE_REQ* pReq, LJV7IF_STORAGE_INFO* pStorageInfo, LJV7IF_GET_STORAGE_RSP* pRsp, LJV7IF_PROFILE_INFO* pProfileInfo, DWORD* pdwData, DWORD dwDataSize);
Parameters	<p>IDeviceId(in) Specifies the communication device to communicate with.</p> <p>pReq(in) Specifies the profiles to get. For each member, see "8 Constant, Structure Definitions".</p> <p>pStorageInfo(out) Represents the actually acquired storage information. For each member, see "8 Constant, Structure Definitions".</p> <p>pRsp(out) Indicates the position and size of the profiles that were actually acquired. For each member, see "8 Constant, Structure Definitions".</p> <p>pProfileInfo(out) Indicates the profile information that was actually acquired. For each member, see "8 Constant, Structure Definitions".</p> <p>pdwData(out) The buffer to get the storage data. The profile storage data is stored in this buffer with "counter value in 10 ms units from the 32-bit base time - LJV7IF_MEASURE_DATA[16] - LJV7IF_PROFILE_HEADER - signed 32-bit profile data - LJV7IF_PROFILE_FOOTER - LJV7IF_MEASURE_DATA[16]" as a single unit of profile storage data, and only the amount of profile storage data that could be acquired is returned. The first LJV7IF_MEASURE_DATA[16] stores the newest measurement values at that time, the second stores the measurement values for that profile. For each member, see "8 Constant, Structure Definitions".</p> <p>dwDataSize(in) pdwData size in BYTES</p>
Return value	<p>0x8080: The operation mode is "high-speed (profile only)"</p> <p>0x8081: The storage target setting is not profile, or "batch measurements on and profile compression (time axis) off"</p>
Explanation	<p>When the storage target setting is profile, this function gets the stored profile data. (Batch setting: off)</p> <p>For the details on the data stored in the profile data (storage order and size), see "9.2.9 Supplement".</p> <p>If there is no accumulated data, 0 is returned for the number of items read in pRsp. If the data specified by the data number to start reading in pRep has not been accumulated, one item of the newest data that has been accumulated is returned.</p>
Supported version	1.00

■ Get batch profile storage data

Format	LONG LJV7IF_GetStorageBatchProfile (LONG IDeviceId, LJV7IF_GET_BATCH_PROFILE_STORAGE_REQ* pReq, LJV7IF_STORAGE_INFO* pStorageInfo, LJV7IF_GET_BATCH_PROFILE_STORAGE_RSP* pRsp, LJV7IF_PROFILE_INFO* pProfileInfo, DWORD* pdwData, ·DWORD dwDataSize, DWORD* pdwTimeOffset, LJV7IF_MEASURE_DATA* pMeasureData);
Parameters	<p>IDeviceId(in) Specifies the communication device to communicate with.</p> <p>pReq(in) Specifies the profiles to get. For each member, see "8 Constant, Structure Definitions".</p> <p>pStorageInfo(out) Represents the actually acquired storage information. For each member, see "8 Constant, Structure Definitions".</p> <p>pRsp(out) Indicates the position and size of the profiles that were actually acquired. For each member, see "8 Constant, Structure Definitions".</p> <p>pProfileInfo(out) Indicates the profile information that was actually acquired. For each member, see "8 Constant, Structure Definitions".</p> <p>pdwData (out) The buffer to get the storage data. The storage data is stored in this buffer with "LJV7IF_PROFILE_HEADER - signed 32-bit profile data - LJV7IF_PROFILE_FOOTER - LJV7IF_MEASURE_DATA[16]" as a single unit of storage data, and only the storage data that could be acquired is returned. The measurement results for each profile are stored in LJV7IF_MEASURE_DATA[16]. For each member, see "8 Constant, Structure Definitions".</p> <p>dwDataSize(in) pdwData size in BYTES</p> <p>pdwTimeOffset(out) Counter value in 10 ms units from the 32-bit base time</p> <p>pMeasureData(out) The measurement results for the relevant batch data. This buffer stores the data for all 16 OUTs including the OUTs that are not measuring. The host requires the passing of a buffer LJV7IF_MEASURE_DATA[16] in size.</p>
Return value	<p>0x8080: The operation mode is "high-speed (profile only)"</p> <p>0x8081: The storage target setting is not profile, or not "batch measurements on and profile compression (time axis) off"</p> <p>0x80A0: The batch data specified by the batch number to read (dwGetBatchNo) in pReq has not been accumulated yet</p>
Explanation	<p>When the storage target setting is profile, this function gets the stored profile data. (Batch setting: on)</p> <p>For the details on the data stored in the profile data (storage order and size), see "9.2.9 Supplement"</p> <p>If the data specified by the number of the profile in the batch to get in pRep has not been accumulated, one item of the newest data that has been accumulated is returned.</p>
Supported version	1.00

9.2.8 High-speed data communication related

For communication devices, see "9.2.9.1 Communication devices".

■ Initialize USB high-speed data communication

Format	LONG LJV7IF_HighSpeedDataUsbCommunicationInitalize (LONG IDDeviceId, void (*pCallBack)(BYTE*, DWORD, DWORD, DWORD, DWORD), DWORD dwProfileCnt, DWORD dwThreadId);
Parameters	IDDeviceId(in) Specifies the communication device to communicate with. pCallBack(in) Specifies the callback function to call when data is received by high-speed communication. dwProfileCnt(in) Specifies the frequency to call the callback function. The callback function is called when the specified number of profiles is received. dwThreadId(in) Thread ID.
Return value	No individual return code
Explanation	This function initializes high-speed communication for a USB connected controller.
Supported version	1.00

■ Initialize Ethernet high-speed data communication

Format	LONG LJV7IF_HighSpeedDataEthernetCommunicationInitalize (LONG IDDeviceId, LJV7IF_ETHERNET_CONFIG* pEthernetConfig, WORD wHighSpeedPortNo, void (*pCallBack)(BYTE*, DWORD, DWORD, DWORD, DWORD), DWORD dwProfileCnt, DWORD dwThreadId);
Parameters	IDDeviceId(in) Specifies the communication device to communicate with. pEthernetConfig(in) Specifies the Ethernet settings used in high-speed communication. wHighSpeedPortNo(in) Specifies the port number used in high-speed communication. pCallBack(in) Specifies the callback function to call when data is received by high-speed communication. dwProfileCnt(in) Specifies the frequency to call the callback function. The callback function is called when the specified number of profiles is received. dwThreadId(in) Thread ID.
Return value	No individual return code
Explanation	This function initializes high-speed communication for an Ethernet connected controller.
Supported version	1.00

■ Request preparation before starting high-speed data communication

Format	LONG LJ7IF_PreStartHighSpeedDataCommunication (LONG IDeviceId, LJ7IF_HIGH_SPEED_PRE_START_REQ* pReq, LJ7IF_PROFILE_INFO* pProfileInfo);
Parameters	IDeviceId(in) Specifies the communication device to communicate with. pReq(in) Specifies what data to send high-speed communication from. pProfileInfo(out) Stores the profile information.
Return value	0x8080: The operation mode is "advanced (with OUT measurement)" 0x8081: The data specified as the send start position does not exist 0x80A1: Already performing high-speed data communication
Explanation	This function performs the preparation to start high-speed data communication. High-speed communication times out and is aborted if not even one packet of a high-speed profile is received within 30 seconds during high-speed communication. For details, see "8.3.1 Supplement".
Supported version	1.00

■ Start high-speed data communication

Format	LONG LJ7IF_StartHighSpeedDataCommunication(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	0x80A0: A high-speed data communication connection was not established 0x80A2: High-speed data communication was not prepared before starting 0x80A4: The send target data was cleared
Explanation	This function starts high-speed data communication. If high-speed data communication does not operate correctly, see "9.2.9.9 High-speed data communication troubleshooting".
Supported version	1.00

■ Stop high-speed data communication

Format	LONG LJ7IF_StopHighSpeedDataCommunication(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	No individual return code
Explanation	This function stops high-speed data communication.
Supported version	1.00

■ Finalize high-speed data communication

Format	LONG LJ7IF_HighSpeedDataCommunicationFinalize(LONG IDeviceId);
Parameters	IDeviceId(in) Specifies the communication device to communicate with.
Return value	No individual return code
Explanation	This function finalizes high-speed data communication.
Supported version	1.00

9.2.9 Supplement

9.2.9.1 Communication devices

The controller that will communicate with the PC is specified as a communication device. The maximum number of controllers that can be communicated with simultaneously is defined by LJV7IF_DEVICE_COUNT (8.1 Constant definitions).

In interfaces that involve communication, you can specify the controller to target for communication with IDeviceID. IDeviceID can be specified as 0 to (LJV7IF_DEVICE_COUNT-1).

- Only one controller can communicate via USB.
Example: When USB has been opened specifying the communication device 0, and USB is opened again specifying communication device 1, it will fail and 0x1001 (LJV7IF_RC_ERR_OPEN) is returned.
- One controller can communicate with three PCs via Ethernet communication.
When a fourth PC connects to the controller, the PC with the oldest date/time of last communication of the three connected PCs is disconnected.
- High-speed communication is only possible between a single controller and a single PC.

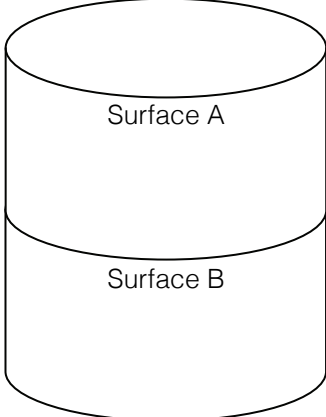
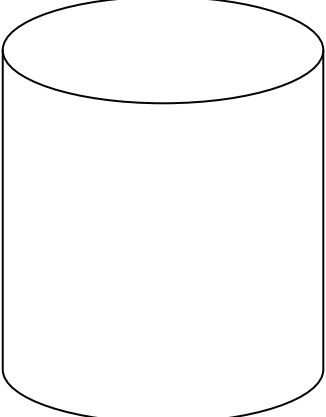
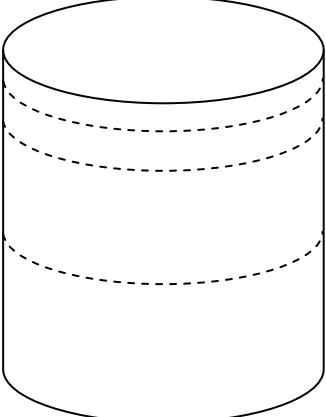
9.2.9.2 Internal memory

The size of measurement data that can be saved and the save interval differs according to the memory allocation setting and the operation mode.

Use caution when using the following functions.

- LJV7IF_GetProfile (Get profiles (operation mode "high-speed (profile only)"))
- LJV7IF_GetBatchProfile (Get batch profiles (operation mode "high-speed (profile only)"))
- LJV7IF_GetStorageStatus (Get storage status)
- LJV7IF_GetStorageData (Get data storage data)
- LJV7IF_GetStorageProfile (Get profile storage data)
- LJV7IF_GetStorageBatchProfile (Get batch profile storage data)

The memory usage area for each memory allocation setting is listed below.

Double buffer	Entire area (overwrite)	Entire area (do not overwrite) ^{*1}
The internal memory is split into two surfaces, surface A and surface B. Each time the program changes, surface A and surface B are alternately used. ^{*2} 	The entire internal memory area is used. 	Of the entire internal memory area, the areas where data has not accumulated can be used to accumulate data. The save data is not deleted, even when the program is changed. 

^{*1} Can only be specified when the operation mode is "advanced (with OUT measurement)".

^{*2} The memory area being used by the active program is called the active surface. The unused surfaces are inactive surfaces.

9.2.9.2.1 For operation mode: high-speed (profile only)

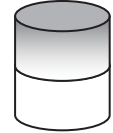
Save target data: profile

When the memory allocation is "double buffer"

The change program operation and the memory usage states transition as shown below.

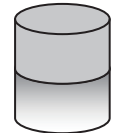
(*: Active surface)

Program number	Surface A	Surface B
1 (measuring)	Saving*	No data



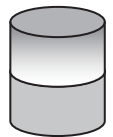
↓ Change program (save target surface is surface B)

Program number	Surface A	Surface B
2 (measuring)	Program number 1 data	Saving*



↓ Change program (save target surface is surface A. Surface A data is deleted.)

Program number	Surface A	Surface B
3 (measuring)	Saving*	Program number 2 data



Internal memory save status (white: no data ⇔ gray: saved) ↑

When the memory allocation is "entire area (overwrite)"

The memory is deleted when a change program operation is performed.

When reading profiles, if the profiles were read by specifying 1 (erase the read profiles) in byErase for LJV7IF_GET_PROFILE_REQ or LJV7IF_GET_BATCH_PROFILE_REQ, the read profiles (batch) and the profiles (batch) older than those are deleted from memory. During high-speed communication, sent profiles are deleted from memory when the controller sends the acquired profiles.

When performing an operation to continuously get profiles from the controller, the memory will become full if the speed that the PC reads and deletes profiles is slower than the speed that data is saved on the controller. The data to get afterward or the data that was previously acquired is deleted according to the operation when memory full setting.

9.2.9.2.2 For operation mode: advanced (with OUT measurement)

Save target data: storage target data (OUT measurement values/profiles)

When the memory allocation is "double buffer"

When the memory allocation is "entire area (overwrite)"

See "9.2.9.2.1 For operation mode: high-speed (profile only)".

When the memory allocation is "entire area (do not overwrite)"

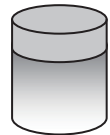
Change program operation and memory states transition example

Program number	Surface 1	Surface 2	Surface 3
1 (measuring)	Saving*	No data	No data



↓ Change program

Program number	Surface 1	Surface 2	Surface 3
2 (measuring)	Program number 1 data	Saving*	No data



↓ Change program

Program number	Surface 1	Surface 2	Surface 3
1 (measuring)	Program number 1 data	Program number 2 data	Saving*



↓ Change program and repeat measurements

Program number	Surface 1	Surface 2	...	Surface N
1 (measuring)	Program number 1 data	Program number 2 data		Saving*



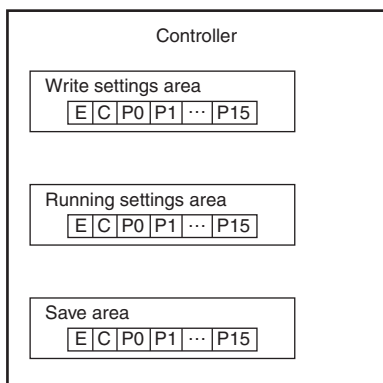
9.2.9.3 Write processing for settings

When sending and initializing settings, the settings to modify are reflected to the specified setting storage levels (write settings area/running area/save area) in the following order in the controller.

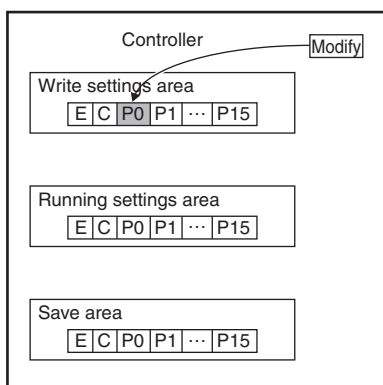
1. Modify the specified settings in the write settings area.
2. Reflect all the settings in the write settings area to the specified level.

Specific example: To change program 0 in the save area

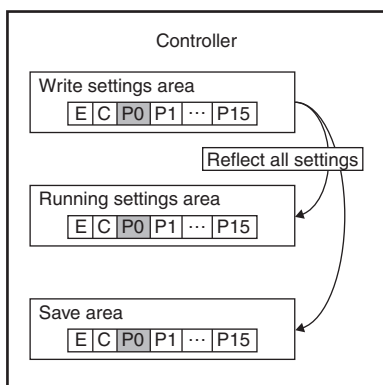
0. State before modification



1. Modify the specified settings in the write settings area.



2. Reflect all the settings in the write settings area to the specified level.



* E: Environment settings C: Common measurement settings
P*: Program settings (* is the program number)

Notes

When reflecting the settings to the specified storage level, all the settings in the write settings area are reflected. Therefore, when there is a difference in the state between the write settings area and the running settings area/save area in step 0 of the specific example, those settings are also reflected from the values in the write settings to the running settings area/save area.

9.2.9.4 Detailed setting errors

The settings have a consistency that must be observed. The detailed setting errors (dwError) that are returned for the send setting and reflect write settings area request commands for settings that not satisfy this consistency are listed below.

dwError value	Error details
0x01000000	The operation mode is "high-speed (profile only)" and the memory allocation is set to entire area (do not overwrite)
0x1X000000 (*1)	The sampling period is outside the configurable range
0x1X06YY00 (*2)	Invalid measurement mode (when set to simple 3D measurements in batch measurement OUT, etc.)
0x1X06YY01 (*2)	Measurement mode "operator" is set for OUT1
0x1X06YY02 (*2)	When the measurement mode is operator, the operator target OUT is invalid
0x1X06YY03 (*2)	Measurement value filter cutoff frequency error
0x1X06YY04 (*2)	Prohibited combination in the measuring range settings (when configured to specify the measuring area when batch is off, etc.)
0x1X06YY05 (*2)	Measuring range setting (when the batch count and the measuring range do not match (also including simple 3D measurement Y direction measurement range))
0x1X06YY06 (*2)	When the measuring range setting is reference OUT, the consistency conditions were not fulfilled
0x1X06YY07 (*2)	Could not match the tolerance setting (upper/lower limit and hysteresis)
0x1X06YY08 (*2)	Could not match the scaling setting (measurement value 1/2, span after calculation)
0x1X070000 (*2)	Could not match the analog output scaling setting (OUT display value 1/2, span after calculation)
0x1X080000 (*2)	The amount of storage has exceeded the configurable range

*1: X indicates the program number and 0 to F is stored there.

*2: YY indicates the OUT number and 00 to 0F is stored there.

9.2.9.5 Profile data values

Profile data is output as signed 32-bit data, but for points where the profile could not be correctly detected, the following values are output.

Value (hexadecimal notation)	Name	Reason
-2147483648 (0x80000000)	Invalid data	This value is output when the peak could not be detected.
-2147483647 (0x80000001)		This value is output when the data is invalid due to a setting such as a mask having been set.
-2147483646 (0x80000002)	Dead zone data	This value is output when the data is located in a dead zone. This value is only output when dead zone processing is enabled.
-2147483645 (0x80000003)	Judgment wait data	This value is output when there is an insufficient amount of profiles for averaging.

9.2.9.6 Profile data amount calculation method

The amount of profile data to get with LJV7IF_GetProfile and other functions is a value multiplied by a correction factor determined from the settings below with 800 as the base.

Setting			Correction factor	Comment
Category	Item	Setting value		
Imaging settings	Measurement range X direction	Full	1.00	Initial value
		Middle	0.75	
		Small	0.50	
	Binning	Off	1.00	Initial value
		On	0.50	
Profile settings	Combine (wide)	Off	1.00	Initial value
		On	2.00	
	Compression (X axis)	Off	1.00	Initial value
		2	0.50	
		4	0.25	

For example, the amount of profile data with the settings below is 300 (= 800 x 0.75 x 1.00 x 1.00 x 0.50) items of data.

Measurement range X direction: Middle
Binning: Off
Combine (wide): Off
Compression (X axis): 2

However, when the amount of profile data found as a result of the equation above is less than 200, the profile compression (X axis) setting is adjusted so that the amount of profile data is 200 or higher.

For example, in a situation like that below, the amount of profile data is 300.

Measurement range X direction: Middle
Binning: Off
Combine (wide): Off
Compression (X axis): 4

The specific calculation is described below.

1. $800 \times 0.75 \times 1.00 \times 1.00 \times 0.25 = 150$
2. The result is less than 200, so the profile compression (X axis) setting is adjust to 2 instead of 4
3. $800 \times 0.75 \times 1.00 \times 1.00 \times 0.5 = 300$
4. The result is 200 or higher, so the amount of profile data above is confirmed

9.2.9.7 Profile data storage order

In functions that get profile data, such as LJV7IF_GetProfile, the profile data stored in the area between LJV7IF_PROFILE_HEADER and LJV7IF_PROFILE_FOOTER is 1 unit of profile data found with "Profile data amount calculation method", and the data is stored in the order below.

- Storage order
 1. 1st head profiles (when compression (time axis) is on, MAX profiles)
 2. 1st head MIN profiles
 3. 2nd head profiles (when compression (time axis) is on, MAX profiles)
 4. 2nd head MIN profiles
- Notes
 - 3 and 4 only exist when the number of heads is 2 and wide is off.
 - 2 and 4 only exist when compression (time axis) is on.

9.2.9.8 Specific examples

(i) For configuration 1 (initial settings)

Heads: 2

Measurement range X direction: Full

Binning: Off

Wide: Off

Compression (X axis): Off

Compression (time axis): Off

The amount of profile data is 800

The profile data storage order is as follows. (See profile data storage order for 1)

1. Head A profiles (800)
3. Head B profiles (800)

When getting 10 profiles with LJV7IF_GetProfile, the data below is stored in pdwProfileData.

Profile 1	LJV7IF_PROFILE_HEADER
	1. Head A profiles (800)
	3. Head B profiles (800)
	LJV7IF_PROFILE_FOOTER
⋮	
Profile 10	LJV7IF_PROFILE_HEADER
	1. Head A profiles (800)
	3. Head B profiles (800)
	LJV7IF_PROFILE_FOOTER

(ii) For configuration 2

Heads: 2

Measurement range X direction: Full

Binning: on

Wide: off

Compression (X axis): 2

Compression (time axis): On

The amount of profile data is 200

The profile data storage order is as follows. (See profile data storage order for 1)

1. Head A MAX profiles (200)
2. Head A MIN profiles (200)
3. Head B MAX profiles (200)
4. Head B MIN profiles (200)

When getting 10 profiles with LJV7IF_GetProfile, the data below is stored in pdwProfileData.

Profile 1	LJV7IF_PROFILE_HEADER
	1. Head A MAX profiles (200)
	2. Head A MIN profiles (200)
	3. Head B MAX profiles (200)
	4. Head B MIN profiles (200)
	LJV7IF_PROFILE_FOOTER
⋮	
Profile 10	LJV7IF_PROFILE_HEADER
	1. Head A MAX profiles (200)
	2. Head A MIN profiles (200)
	3. Head B MAX profiles (200)
	4. Head B MIN profiles (200)
	LJV7IF_PROFILE_FOOTER

9.2.9.9 High-speed data communication troubleshooting

Symptom	Item to check	Remedy
The application exits with an error after high-speed data communication starts.	Have the callback function call protocols been specified correctly?	Match the callback function call protocols with those in the header file.
The profile data to be obtained becomes abnormal at irregular intervals.	Is the data that is used by the main thread being used by a callback function without first acquiring exclusive processing access?	Acquire exclusive processing access for shared data.
The profile data to be obtained becomes abnormal at regular intervals.	Is heavy processing (such as the saving of files) being performed within a callback function?	Change the callback function so that its processing time becomes shorter.
	Is the communication speed of your communication path sufficient?	Change to a high-speed communication path such as 100BASE-T.
High-speed communication is interrupted.	Is heavy processing (such as the saving of files) being performed within a callback function?	Change the callback function so that its processing time becomes shorter.
	Is the communication speed of your communication path sufficient?	Change to a high-speed communication path such as 100BASE-T.

10 Common Return Codes

10.1 Return codes returned by the communication library

The return codes shown below are judged in the communication library and returned to the application. Specifically, these codes are returned when the library fails to communicate with the controller or when processing could not be completed due to a state dependency in the communication library.

Definition name	Data (Lower 2 bytes)	Cause
LJV7IF_RC_OK	0x0000	Normal termination
LJV7IF_RC_ERR_OPEN	0x1000	Failed to open the communication path.
LJV7IF_RC_ERR_NOT_OPEN	0x1001	The communication path was not established.
LJV7IF_RC_ERR_SEND	0x1002	Failed to send the command.
LJV7IF_RC_ERR_RECEIVE	0x1003	Failed to receive a response.
LJV7IF_RC_ERR_TIMEOUT	0x1004	A timeout occurred while waiting for the response.
LJV7IF_RC_ERR_NOMEMORY	0x1005	Failed to allocate memory.
LJV7IF_RC_ERR_PARAMETER	0x1006	An invalid parameter was passed.
LJV7IF_RC_ERR_RECV_FMT	0x1007	The received response data was invalid.
LJV7IF_RC_ERR_HISPEED_NO_DEVICE	0x1009	High-speed communication initialization could not be performed.
LJV7IF_RC_ERR_HISPEED_OPEN_YET	0x100A	High-speed communication was initialized.
LJV7IF_RC_ERR_HISPEED_RECV_YET	0x100B	Error already occurred during high-speed communication (for high-speed communication)
LJV7IF_RC_ERR_BUFFER_SHORT	0x100C	The buffer size passed as an argument is insufficient.

10.2 Return codes returned from the controller

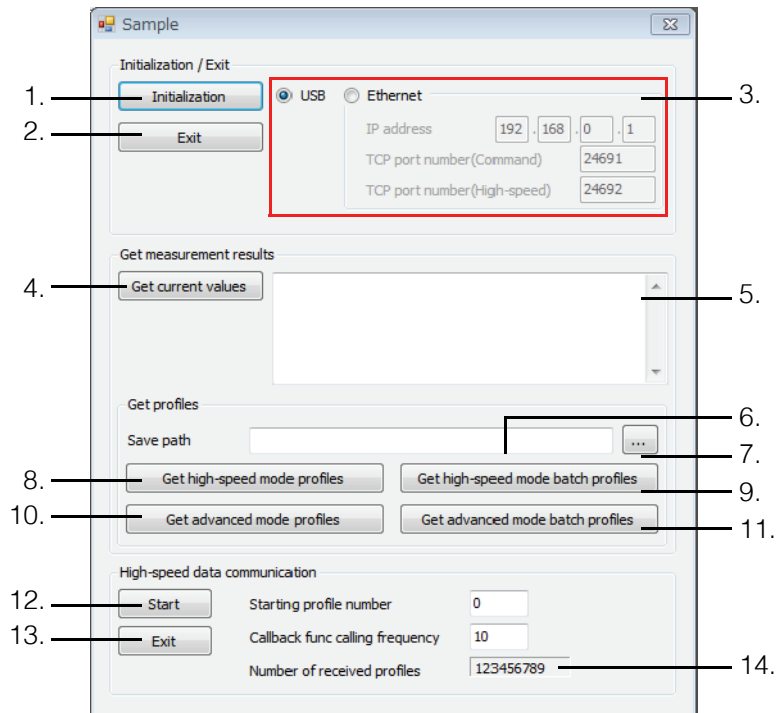
The return codes shown below are returned when communication with the controller was successful but the controller could not process the command.

Data (Lower 2 bytes)	Cause
0x8041	Status error (when a system error has occurred, etc.)
0x8042	Parameter error (when an invalid parameter was set, etc.)

11 Sample Program

This chapter describes the sample program which has been included as an example of how to create an application using the communication library.

11.1 User interface specification



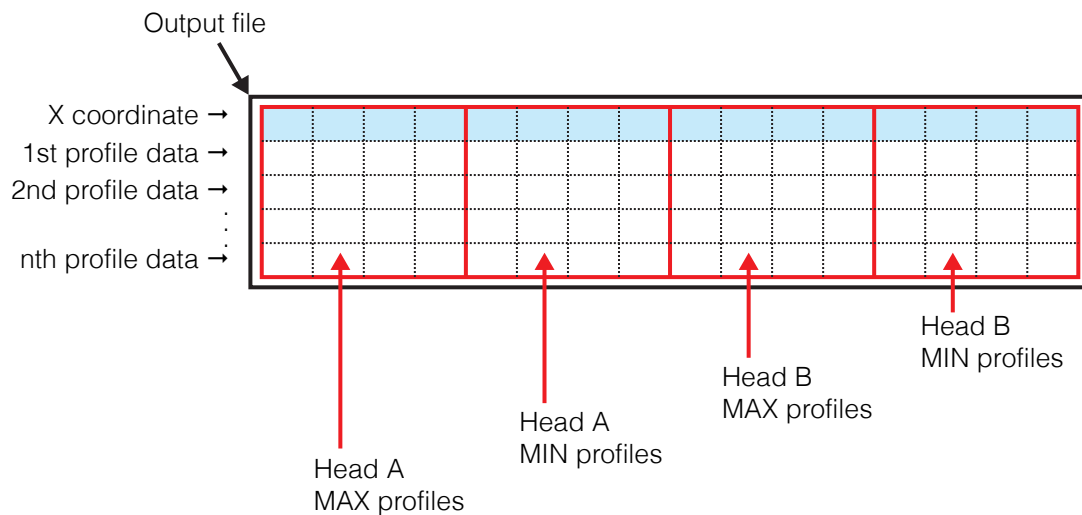
1.	Initialize	Initializes the DLL and opens the communication path.
2.	Finalize	Closes the communication path and finalizes the DLL.
3.	Select communication path	Selects the communication path.
4.	Get newest values	Gets the newest OUT measurement values.
5.	Newest values display	Shows the OUT values obtained by getting the current values.
6.	Save path	Sets the path to save the data acquired with get profiles.
7.	... (Browse)	Opens the save file dialog box so the save path can be selected.
8.	High-speed mode get profiles	Gets 10 profiles of the current profile from the controller and saves them to file.
9.	High-speed mode get batch profiles	Gets all the profiles in the committed batch data from the controller and saves them to file.
10.	Advanced mode get profiles	Gets the current profile from the controller and saves it to file.
11.	Advanced mode get batch profiles	Gets all the profiles in the committed batch data from the controller and saves them to file.
12.	Start	Starts high-speed data communication. The profiles from the controller are received from the starting profile number. The communication path selected and configured in 3 is used.
13.	Stop	Stops high-speed data communication.
14.	Number of received high-speed profiles	Shows the number of profiles that are received in high-speed data communication. When the number of profiles exceeds 4294967295 ($=2^{32}-1$), the counter starts again from 0.

11.2 Save file format

This section describes the file format for profiles that are saved in the get profile process (8 to 11).

- File format: TSV (tab separated text file)
- Values: The values acquired with the get profile interface are saved unmodified.
- Units: See "9.2.6 Acquiring measurement results".

Image of the arrangement of data in the save file (2 heads, compression (time axis) on)



12 Appendix

■ Setting details

As shown in the tables below, the parameters that can be set are configured by the combination of category and setting item for each setting type.

Depending on the parameter, additional details can be specified in setting target 1 to setting target 4.

Environment settings (specify "01h" for setting type)

		Category
		Nothing specified (00h)
Setting item	00 h	Device name
	01 h	Operation at next power on
	02 h	High-speed communication band restriction
	03 h	MTU during high-speed communication
	04 h	IP address
	05 h	Subnet mask
	06 h	Gateway
	07 h	TCP command port number
	08 h	TCP high-speed power number
	09 h	UDP port number
	0A h	Baud rate
	0B h	Parity

Common measurement settings (specify "02h" for setting type)

		Category
		Nothing specified (00h)
Setting item	00 h	Operation mode
	01 h	Memory allocation
	02 h	Operation when memory full
	03 h	Parallel imaging
	04 h	Strobe output time
	05 h	Head installation
	06 h	TRG minimum input time
	07 h	ENCODER minimum input time
	08 h	Control terminal minimum input time
	09 h	Change program

Program (specify "10h to 1Fh" for setting type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15)

		Category									
		Trigger settings (00h)	Imaging settings (01h)	Profile settings (02h)	Master (03h)	Position correction settings (04h)	Profile mask settings (05h)	OUT settings (06h)	Terminal settings (07h)	Storage settings (08h)	Program name (09h)
Setting item	00 h	Standard setting/advanced setting	Standard setting/advanced setting	Standard setting/advanced setting	Standard setting/advanced setting	Standard setting/advanced setting	Standard setting/advanced setting	Standard setting/advanced setting	Standard setting/advanced setting	Standard setting/advanced setting	Program name
	01 h	Trigger mode	Binning	Combine (wide)	Master profile	Dual head mode on/off	Mask area settings group	OUT name	Judgment output condition	Storage target	
	02 h	Sampling period	X direction	Compression (X axis)		Dual head mode target head	Mask area settings individual	Minimum display unit	Target OUT	Storage condition	
	03 h	Batch measurement	Z direction	Compression (time axis)		Dual head mode X correction amount		Measurement mode	Scaling	Terminal/command detail	
	04 h	Inter-trigger pitch	Measurement target	Time axis compression count		Dual head mode Z correction amount		Measurement value hold count		OUT measurement value edge detail	
	05 h	Inter-trigger pitch count	CMOS sensitivity	Dead zone processing enabled/disabled		Angle correction on/off		Measurement value filter		OUT measurement value level detail	
	06 h	Mutual interference prevention	Exposure time	Reverse (X)		Angle correction detail		Measurement value filter detail			
	07 h	Input mode	Imaging mode	Reverse (Z)		Backup correction on/off		Scaling measurement value 1			
	08 h	Skipping	Multi emission (light intensity optimization) detail	Shift (X)		Backup correction detail		Scaling display value 1			
	09 h	Points to skip	Multi emission (composite) detail	Shift (Z)		XZ correction selection		Scaling measurement value 2			
	0A h	Batch count	Mask settings	Median (X axis)				Scaling display value 2			
	0B h		Control mode	Smoothing		X correction detail		Measuring mode			
	0C h		Upper limit value	Median (time axis)		Z correction detail		Measuring range			
	0D h		Lower limit value	Averaging		Feature point correction detail		Offset			
	0E h		FB target range	Invalid data processing (time axis)				Upper limit			
	0F h		Peak detection level	Tilt correction				Lower limit			
	10 h		Invalid data interpolation count	Height correction				Hysteresis			
	11 h		Peak selection					ZERO			
	12 h		Peak width filter					TIMING/RESET			
	13 h		Imaging internal settings					Auto zero base value			

Environment settings

(specify "01h" for setting type)

Always specify 0 for the environment settings category.

Specify the values below for the setting item.

Always specify 0 for target 1 to target 4 for the environment settings.

<Device name (00h)>

Byte	Parameter details
1	Device name, byte 1
2	Device name, byte 2
3	Device name, byte 3
to	to
32	Device name, byte 32

Device name Device name string (32 characters max)
* 0 is not appended to the end

<Operation at next power on (01h)>

Byte	Parameter details
1	Operation at next power on
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Operation at next power on 0: BOOT→IP addresses fixed,
1: IP address fixed, 2: BOOTP

<High-speed communication band restriction (02h)>

Byte	Parameter details
1	High-speed communication band restriction
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

High-speed communication band restriction
0: Off, 1: 500 Mbps,
2: 200 Mbps, 3: 100 Mbps

<MTU during high-speed communication (03h)>

Byte	Parameter details
1	MTU setting
2	
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

MTU setting 1500 to 9216

<IP address (04h)>

<Subnet mask (05h)>

<Gateway (06h)>

Byte	Parameter details
1	IP address, 1st byte
2	IP address, 2nd byte
3	IP address, 3rd byte
4	IP address, 4th byte

IP address For error judgment conditions, refer to the LJ-V7000 Series User's Manual.

<TCP command port number (07h)>

<TCP high-speed power number (08h)>

<UDP port number (09h)>

Byte	Parameter details
1	Port number
2	
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Port number For error judgment conditions, refer to the LJ-V7000 Series User's Manual.

<Baud rate (0Ah)>

Byte	Parameter details
1	Baud rate
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Baud rate 0: 9600, 1: 19200, 2: 38400,
3: 57600, 4: 115200

<Parity (0Bh)>

Byte	Parameter details
1	Parity
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Parity 0: None, 1: Even, 2: Odd

Common measurement settings (specify "02h" for setting type)

Always specify 0 for the common measurement settings category.

Specify the values below for the setting item.

Always specify 0 for target 1 to target 4 for the common measurement settings.

<Operation mode (00h)>

Byte	Parameter details
1	Operation mode
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Operation mode 0: High-speed, 1: Advanced

<Memory allocation (01h)>

Byte	Parameter details
1	Memory allocation setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Memory allocation setting 0: Double buffer,
1: Entire area (overwrite),
2: Entire area (do not overwrite)

<Operation when memory full (02h)>

Byte	Parameter details
1	Operation when memory full
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Operation when memory full 0: Overwrite, 1: Stop

<Parallel imaging (03h)>

Byte	Parameter details
1	Parallel imaging
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Parallel imaging 0: Disabled, 1: Enabled

<Strobe output time (04h)>

Byte	Parameter details
1	Strobe output time
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Strobe output time 0: 10 μ s, 1: 20 μ s, 2: 50 μ s, 3: 100 μ s, 4: 250 μ s, 5: 500 μ s, 6: 1 ms, 7: 2 ms, 8: 5 ms, 9: 10 ms, 10: 20 ms

<Head installation (05h)>

Byte	Parameter details
1	Head installation status
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Head installation status 0: Not configured, 1: Independent, 2: Horizontal/same direction, 3: Horizontal/reverse direction, 4: Interleaved/same direction, 5: Interleaved/reverse direction

<TRG minimum input time (06h)>

Byte	Parameter details
1	Constant when TRG input terminal
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Constant when TRG input terminal 0: 7 μ s, 1: 10 μ s, 2: 20 μ s, 3: 50 μ s, 4: 100 μ s, 5: 200 μ s, 6: 500 μ s, 7: 1 ms

<ENCODER minimum input time (07h)>

Byte	Parameter details
1	Constant when ENCODER input terminal
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Constant when ENCODER input terminal 0: 120 ns, 1: 150 ns, 2: 250 ns, 3: 500 ns, 4: 1 μ s, 5: 2 μ s, 6: 5 μ s, 7: 10 μ s, 8: 20 μ s

<Control terminal minimum input time (08h)>

Byte	Parameter details
1	Control terminal minimum input time
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Control terminal minimum input time 0: 250 μ s, 1: 1 ms

<Change program (09h)>

Byte	Parameter details
1	Change program
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Change program 0: Terminal, 1: Command

Program (specify "10h to 1Fh" for setting type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15)

For program categories, see the table (page 59).

Program (specify "10h to 1Fh" for setting type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Trigger settings (00h)

Always specify 0 for the setting item and for setting target 1 to setting target 4 for the trigger settings.

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting 0: Standard setting,
1: Advanced setting,
2: Standard setting (with advanced setting modification),
3: Advanced setting (with advanced setting modification)

<Trigger mode (01h)>

Byte	Parameter details
1	Trigger mode
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Trigger mode 0: Continuous trigger, 1: External trigger,
2: Encoder trigger

<Sampling period (02h)>

Byte	Parameter details
1	Sampling period
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Sampling period 0: 10 Hz, 1: 20Hz, 2: 50 Hz, 3: 100 Hz,
4: 200 Hz, 5: 500 Hz, 6: 1 KHz, 7: 2 KHz,
8: 4 KHz, 9: 4.13 KHz, 10: 8 KHz,
11: 16 KHz, 12: 32 KHz, 13: 64 KHz

<Batch measurement (03h)>

Byte	Parameter details
1	Batch measurement
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Batch measurement 0: Batch off, 1: Batch on

<Inter-trigger pitch (04h)>

Byte	Parameter details
1	Inter-trigger pitch
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Inter-trigger pitch 0: Pitch off, 1: Pitch on

<Inter-trigger pitch count (05h)>

Byte	Parameter details
1	Pitch count
2	
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Pitch count 0 to 50000

* Display unit = 0.001 to 50.000 mm

<Mutual interference prevention (06h)>

Byte	Parameter details
1	Mutual interference prevention
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Mutual interference prevention 0: Mutual interference off,
1: Mutual interference on

<Input mode (07h)>

Byte	Parameter details
1	Encoder trigger input mode
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Encoder trigger input mode 0: Single phase 1x multiplier
(no orientation),
1: Dual phase 1x multiplier,
2: Dual phase 2x multiplier,
3: Dual phase 4x multiplier

<Skipping (08h)>

Byte	Parameter details
1	Encoder trigger skipping
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Encoder trigger skipping 0: Skipping off,
1: Skipping on

<Points to skip (09h)>

Byte	Parameter details
1	Encoder trigger skipping count
2	
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Encoder trigger skipping count 2 to 1000

<Batch count (0Ah)>

Byte	Parameter details
1	Batch count
2	
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Batch count 50 to 15000

Program (specify "10h to 1Fh" for settings type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Imaging settings (01h)

Specify the value for the setting item and for setting target 1 to setting target 4 for the imaging settings according to the table below.

			Setting target 1	Setting target 2	Setting target 3	Setting target 4
Setting item	00 h	Standard setting/advanced setting	Ignored	Ignored	Ignored	Ignored
	01 h	Binning	Ignored	Ignored	Ignored	Ignored
	02 h	X direction	Ignored	Ignored	Ignored	Ignored
	03 h	Z direction	Ignored	Ignored	Ignored	Ignored
	04 h	Measurement target	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	05 h	CMOS sensitivity	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	06 h	Exposure time	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	07 h	Imaging mode	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	08 h	Multi emission (light intensity optimization) detail	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	09 h	Multi emission (composite) detail	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0A h	Mask setting	Head A (wide): 0 Head B: 1	Upper mask: 0 to 2 Lower mask: 3 to 5	Ignored	Ignored
	0B h	Control mode	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0C h	Upper limit value	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0D h	Lower limit value	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0E h	FB target range	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0F h	Peak detection level	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	10 h	Invalid data interpolation count	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	11 h	Peak selection	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	12 h	Peak width filter	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	13 h	Imaging internal settings	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting
 0: Standard setting, 1: Advanced setting,
 2: Standard setting (with advanced setting modification), 3: Advanced setting (with advanced setting modification)

<Binning (01h)>

Byte	Parameter details
1	Binning
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Binning 0: Binning off, 1: Binning on

<X direction (02h)>

Byte	Parameter details
1	Measurement range X direction
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Measurement range X direction 0: Full, 1: Middle, 2: Small

<Z direction (03h)>

Byte	Parameter details
1	Measurement range Z direction
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Measurement range Z direction 0: Full, 1: Middle, 2: Small

<Measurement target (04h)>

Byte	Parameter details
1	Measurement target
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Measurement target 0: Standard,
 1: Light intensity difference size,
 2: Multi reflection, 3: Translucent

<CMOS sensitivity (05h)>

Byte	Parameter details
1	Light reception sensitivity characteristics
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Light reception sensitivity characteristics
 0: High accuracy,
 1: High dynamic range 1,
 2: High dynamic range 2,
 3: High dynamic range 3

<Exposure time (06h)>

Byte	Parameter details
1	Exposure time
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Exposure time 0: 15 μ s, 1: 30 μ s, 2: 60 μ s, 3: 120 μ s, 4: 240 μ s,
 5: 480 μ s, 6: 960 μ s, 7: 1920 μ s, 8: 5 ms,
 9: 10 ms

<Imaging mode (07h)>

Byte	Parameter details
1	Imaging mode
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Imaging mode 0: Standard, 1: Multi emission (composite),
 2: Multi emission (light intensity optimization)

<Multi emission (light intensity optimization) detail (08h)>

Byte	Parameter details
1	Emission count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Emission count 0: 2 Times, 1: 4 Times

<Multi emission (composite) detail (09h)>

Byte	Parameter details
1	Emission count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Emission count 0: 3 times, 1: 5 times

<Mask setting (0Ah)>

Byte	Parameter details
1	Enabled/disabled
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	X coordinate 1
6	
7	Z coordinate 1
8	
9	X coordinate 2
10	
11	Z coordinate 2
12	
13	X coordinate 3
14	
15	Z coordinate 3
16	

Enabled/disabled 0: Mask disabled, 1: Rectangle,
 2: Triangle

X coordinate 1 to 3 0 to 639
Z coordinate 1 to 3 0 to 479

<Control mode (0Bh)>

Byte	Parameter details
1	Control mode
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Control mode 0: Auto, 1: Manual

<Upper limit value (0Ch)>

Byte	Parameter details
1	FB upper limit value
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

FB upper limit value 1 to 99

<Lower limit value (0Dh)>

Byte	Parameter details
1	FB lower limit value
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

FB lower limit value 1 to 99

<FB target range (0Eh)>

Byte	Parameter details
1	FB target range start
2	
3	FB target range end
4	

FB target range start/end 0 to 639

<Peak detection level (0Fh)>

Byte	Parameter details
1	Peak detection level
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Peak detection level 1 to 5

<Invalid data interpolation count (10h)>

Byte	Parameter details
1	Invalid data interpolation count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Invalid data interpolation count 0 to 255

<Peak selection (11h)>

Byte	Parameter details
1	Peak selection
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Peak selection 0: Standard, 1: Near, 2: Far,
3: Remove X multi reflection,
4: Remove Y multi reflection,
5: Make invalid data

<Peak width filter (12h)>

Byte	Parameter details
1	Peak width filter
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Peak width filter 0: Filter off, 1: Filter on

<Imaging internal settings (13h)>

Byte	Parameter details
1	Head specific setting
2	Head specific setting
3	Head specific setting
4	Head specific setting
5	Head specific setting
6	Head specific setting
7	Head specific setting
8	Head specific setting
9	Head specific setting
10	Head specific setting
11	Head specific setting
12	Head specific setting
13	Head specific setting
14	Head specific setting
15	Head specific setting
16	Head specific setting

Note: These are head specific settings used in the controller.
Do not reference or use them.

Program (specify "10h to 1Fh" for settings type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Profile settings (02h)

Specify the value for the setting item and for setting target 1 to setting target 4 for the profile settings according to the table below.

			Setting target 1	Setting target 2	Setting target 3	Setting target 4
Setting item	00 h	Standard setting/advanced setting	Ignored	Ignored	Ignored	Ignored
	01 h	Combine (wide)	Ignored	Ignored	Ignored	Ignored
	02 h	Compression (X axis)	Ignored	Ignored	Ignored	Ignored
	03 h	Compression (time axis)	Ignored	Ignored	Ignored	Ignored
	04 h	Time axis compression count	Ignored	Ignored	Ignored	Ignored
	05 h	Dead zone processing enabled/disabled	Ignored	Ignored	Ignored	Ignored
	06 h	Reverse (X)	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	07 h	Reverse (Z)	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	08 h	Shift (X)	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	09 h	Shift (Z)	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0A h	Median (X axis)	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0B h	Smoothing	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0C h	Median (time axis)	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0D h	Averaging	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0E h	Invalid data processing (time axis)	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	0F h	Tilt correction	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	10 h	Height correction	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting

0: Standard setting, 1: Advanced setting,
2: Standard setting (with advanced setting
modification), 3: Advanced setting (with
advanced setting modification)

<Combine (wide) (01h)>

Byte	Parameter details
1	Wide setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Wide setting 0: Wide off, 1: Wide on

<Compression (X axis) (02h)>

Byte	Parameter details
1	Compression (X axis)
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Compress (X axis) 0: Compress off, 1: 2, 2: 4

<Compression (time axis) (03h)>

Byte	Parameter details
1	Compression (time axis)
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Compress (time axis) 0: Compress off, 1: Compression on

<Time axis compression count (04h)>

Byte	Parameter details
1	Time axis compression count
2	
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Time axis compression count 2 to 1000

<Dead zone processing enabled/disabled (05h)>

Byte	Parameter details
1	Dead zone processing enabled/disabled
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Dead zone processing enabled/disabled

0: Dead zone processing disabled,
1: Dead zone processing enabled

<Reverse (X) (06h)>**<Reverse (Z) (07h)>**

Byte	Parameter details
1	Reverse
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Reverse 0: Reverse off, 1: Reverse on

<Shift (X) (08h)>**<Shift (Z) (09h)>**

Byte	Parameter details
1	Shift amount
2	
3	
4	

Shift amount Specify the shift amount in the range set for each head

<Median (X axis) (0Ah)>**<Median (time axis) (0Ch)>**

Byte	Parameter details
1	Median count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Media count 0: Off, 1: 3 points, 2: 5 points, 3: 7 points,
4: 9 points**<Smoothing (0Bh)>**

Byte	Parameter details
1	Smoothing count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Smoothing count 0: 1 time, 1: 2 times, 2: 4 times,
3: 8 times, 4: 16 times, 5: 32 times,
6: 64 times**<Averaging (0Dh)>**

Byte	Parameter details
1	Averaging count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Averaging count 0: 1 time, 1: 2 times, 2: 4 times,
3: 8 times, 4: 16 times, 5: 32 times,
6: 64 times, 7: 128 times, 8: 256 times**<Invalid data processing (time axis) (0Eh)>**

Byte	Parameter details
1	Processing times
2	Resume times
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Processing times 0 to 255

Resume times 0 to 255

<Tilt correction (0Fh)>

Byte	Parameter details
1	on/off
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	Linear calculation area count
6	Reserved (fixed as 0)
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Area start position 1
10	
11	
12	
13	Area end position 1
14	
15	
16	
17	Area start position 2
18	
19	
20	
21	Area end position 2
22	
23	
24	
25	Post-correction angle
26	Correction angle
27	
28	

On/off 0: Correction disabled,
 1: Correction enabled

Linear calculation area count 0: Area 2 disabled,
 1: Area 2 enabled

Area start position 1/2 Specify the position in the range
 set for each head

Area end position 1/2 Specify the position in the range
 set for each head

Post-correction angle -4500 to +4500

Correction angle -4500 to +4500

* The unit for the post-correction angle and the correction angle is -45.00 to +45.00 deg

<Height correction (10h)>

Byte	Parameter details
1	on/off
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	Area start position 1
6	
7	
8	
9	Area end position 1
10	
11	
12	
13	Area start position 2
14	
15	
16	
17	Area end position 2
18	
19	
20	
21	Post-correction height
22	
23	
24	
25	Correction span
26	
27	
28	

On/off 0: Correction disabled,
 1: Correction enabled

Area start position 1/2 Specify the position in the range
 set for each head

Area end position 1/2 Specify the position in the range
 set for each head

Post-correction height 0 to 99999

Correction factor 0 to 131072

* The post-correction height unit is 0 to 999.99 mm

* The correction factor indicates a value from 0 to 2x in 1/65536 units

Program (specify "10h to 1Fh" for setting type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Master (03h)

Specify the value for the setting item and for setting target 1 to setting target 4 for master according to the table below.

			Setting target 1	Setting target 2	Setting target 3	Setting target 4
Setting item	00 h	Standard setting/advanced setting	Ignored	Ignored	Ignored	Ignored
	01 h	Master profile	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting

0: Standard setting, 1: Advanced setting,
2: Standard setting (with advanced setting
modification), 3: Advanced setting (with
advanced setting modification)

<Master profile (01h)>

Wide off

Byte	Parameter details
1	Enabled/disabled
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	Profile data amount
6	
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	X coordinate data start position
10	
11	
12	X direction pitch
13	
14	
15	
16	Profile
17	
18	
19	
20	to
to	
3213	
3214	
3215	Profile
3216	

800 items worth

Wide on

Byte	Parameter details
1	Enabled/disabled
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	Profile data amount
6	
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	X coordinate data start position
10	
11	
12	X direction pitch
13	
14	
15	
16	Profile
17	
18	
19	
20	to
to	
3213	
3214	
3215	Profile
3216	

1600 items worth

Enabled/disabled
Profile data amount

X coordinate data start position
X direction pitch
Profile

0: Master disabled, 1: Master enabled
When wide is off: 0 to 800,
When wide is on: 0 to 1600
Specify the position in the range set for each head
Specify the pitch in the range set for each head
20-bit signed integer

Program (specify "10h to 1Fh" for settings type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Position correction settings (04h)

Specify the value for the setting item and for setting target 1 to setting target 4 for the position correction settings according to the table below.

			Setting target 1	Setting target 2	Setting target 3	Setting target 4
Setting item	00h	Standard setting/advanced setting	Ignored	Ignored	Ignored	Ignored
	01h	Dual head mode on/off	Ignored	Ignored	Ignored	Ignored
	02h	Dual head mode target head	Ignored	Ignored	Ignored	Ignored
	03h	Dual head mode X correction amount	Ignored	Ignored	Ignored	Ignored
	04h	Dual head mode Z correction amount	Ignored	Ignored	Ignored	Ignored
	05h	Angle correction on/off	Head A (wide): 0 Head B: 1	Position correction 1: 0 Position correction 2: 1	Ignored	Ignored
	06h	Angle correction detail	Head A (wide): 0 Head B: 1	Position correction 1: 0 Position correction 2: 1	Ignored	Ignored
	07h	Backup correction on/off	Head A (wide): 0 Head B: 1	Position correction 1: 0 Position correction 2: 1	Ignored	Ignored
	08h	Backup correction detail	Head A (wide): 0 Head B: 1	Position correction 1: 0 Position correction 2: 1	Ignored	Ignored
	09h	XZ correction selection	Head A (wide): 0 Head B: 1	Position correction 1: 0 Position correction 2: 1	Ignored	Ignored
	0Ah	Ignored	Ignored	Ignored	Ignored	Ignored
	0Bh	X correction detail	Head A (wide): 0 Head B: 1	Position correction 1: 0 Position correction 2: 1	Ignored	Ignored
	0C h	Z correction detail	Head A (wide): 0 Head B: 1	Position correction 1: 0 Position correction 2: 1	Ignored	Ignored
	0Dh	Feature point correction detail	Head A (wide): 0 Head B: 1	Position correction 1: 0 Position correction 2: 1	Ignored	Ignored

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting

0: Standard setting, 1: Advanced setting,
2: Standard setting (with advanced setting modification), 3: Advanced setting (with advanced setting modification)

<Dual head mode on/off (01h)>

Byte	Parameter details
1	Dual head mode
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Dual head mode 0: Dual head off,
1: Dual head on

<Dual head mode target head (02h)>

Byte	Parameter details
1	Correction target head
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Correction target head 0: Head A, 1: Head B

<Dual head mode X correction amount (03h)>

<Dual head mode Z correction amount (04h)>

Byte	Parameter details
1	Correction amount
2	
3	
4	

Correction amount 10000.0 to +10000.0

<Angle correction on/off (05h)>

<Backup correction on/off (07h)>

Byte	Parameter details
1	on/off
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

On/off 0: Off, 1: On

Byte	Parameter details
1	Linear calculation area count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	Area start position 1
6	
7	
8	
9	Area end position 1
10	
11	
12	
13	Area start position 2
14	
15	
16	
17	Area end position 2
18	
19	
20	
21	Correction reference
22	Reserved (fixed as 0)
23	Reserved (fixed as 0)
24	Reserved (fixed as 0)

<X correction detail (0Bh)>

Byte	Parameter details
1	Area start position
2	
3	
4	
5	Area end position
6	
7	
8	
9	Edge direction
10	Detection direction
11	Detection number
12	Reserved (fixed as 0)
13	Edge level
14	
15	
16	

Byte	Parameter details
1	XY correction selection
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Byte	Parameter details
1	Area start position
2	
3	
4	
5	Area end position
6	
7	
8	
9	Height type
10	Reserved (fixed as 0)
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)

Byte	Parameter details
1	Correction target selection
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Program (specify "10h to 1Fh" for settings type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Profile mask settings (05h)

Specify the value for the setting item and for setting target 1 to setting target 4 for the profile mask settings according to the table below.

			Setting target 1	Setting target 2	Setting target 3	Setting target 4
Setting item	00 h	Standard setting/advanced setting	Ignored	Ignored	Ignored	Ignored
	01 h	Mask area settings group	Head A (wide): 0 Head B: 1	Ignored	Ignored	Ignored
	02 h	Mask area settings individual	Head A (wide): 0 Head B: 1	Mask 1 to 5: 0 to 4	Ignored	Ignored

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting
 0: Standard setting, 1: Advanced setting,
 2: Standard setting (with advanced setting modification), 3: Advanced setting (with advanced setting modification)

<Mask area settings group (01h)>

Byte	Parameter details
1	Area selection
2	Position correction selection
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	X coordinate 1
6	
7	
8	
9	Z coordinate 1
10	
11	
12	
13	X coordinate 2
14	
15	
16	
17	Z coordinate 2
18	
19	
20	
21	X coordinate 3
22	
23	
24	
25	Z coordinate 3
26	
27	
28	
to	
140	

Area selection
 0: Disabled, 1: Rectangle, 2: Triangle
 Position correction selection
 0: No position correction, 1: Position correction 1, 2: Position correction 2

X coordinate 1/2/3

Z coordinate 1/2/3

When rectangle is selected, select the coordinates as the two points (X1, Z1) and (X2, Z2).
 When triangle is selected, select the coordinates as the three points (X1, Z1), (X2, Z2), and (X3, Z3).

* The above shows one area's worth of settings.

<Mask area settings individual (02h)>

Byte	Parameter details
1	Area selection
2	Position correction selection
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	X coordinate 1
6	
7	
8	
9	Z coordinate 1
10	
11	
12	
13	X coordinate 2
14	
15	
16	
17	Z coordinate 2
18	
19	
20	
21	X coordinate 3
22	
23	
24	
25	Z coordinate 3
26	
27	
28	

Area selection

Position correction selection

X coordinate 1/2/3

Z coordinate 1/2/3

0: Disabled, 1: Rectangle, 2: Triangle
 0: No position correction, 1: Position correction 1, 2: Position correction 2
 When rectangle is selected, select the coordinates as the two points (X1, Z1) and (X2, Z2).
 When triangle is selected, select the coordinates as the three points (X1, Z1), (X2, Z2), and (X3, Z3).

Program (specify "10h to 1Fh" for settings type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → OUT settings (06h)

Specify the value for the setting item and for setting target 1 to setting target 4 for the OUT settings according to the table below.

			Setting target 1	Setting target 2	Setting target 3	Setting target 4
Setting item	00h	Standard setting/advanced setting	Ignored	Ignored	Ignored	Ignored
	01h	OUT name	OUT number: 0 to 15	Ignored	Ignored	Ignored
	02h	Minimum display unit	OUT number: 0 to 15	Ignored	Ignored	Ignored
	03h	Measurement mode	OUT number: 0 to 15	Ignored	Ignored	Ignored
	04h	Measurement value hold count	OUT number: 0 to 15	Ignored	Ignored	Ignored
	05h	Measurement value filter	OUT number: 0 to 15	Ignored	Ignored	Ignored
	06h	Measurement value filter detail	OUT number: 0 to 15	Ignored	Ignored	Ignored
	07h	Scaling measurement value 1	OUT number: 0 to 15	Ignored	Ignored	Ignored
	08h	Scaling display value 1	OUT number: 0 to 15	Ignored	Ignored	Ignored
	09h	Scaling measurement value 2	OUT number: 0 to 15	Ignored	Ignored	Ignored
	0Ah	Scaling display value 2	OUT number: 0 to 15	Ignored	Ignored	Ignored
	0Bh	Measuring mode	OUT number: 0 to 15	Ignored	Ignored	Ignored
	0Ch	Measuring range	OUT number: 0 to 15	Ignored	Ignored	Ignored
	0Dh	Offset	OUT number: 0 to 15	Ignored	Ignored	Ignored
	0Eh	Upper limit	OUT number: 0 to 15	Ignored	Ignored	Ignored
	0Fh	Lower limit	OUT number: 0 to 15	Ignored	Ignored	Ignored
	10h	Hysteresis	OUT number: 0 to 15	Ignored	Ignored	Ignored
	11h	ZERO	OUT number: 0 to 15	Ignored	Ignored	Ignored
	12h	TIMING/RESET	OUT number: 0 to 15	Ignored	Ignored	Ignored
	13h	Auto zero base value	OUT number: 0 to 15	Ignored	Ignored	Ignored

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting
 0: Standard setting,
 1: Advanced setting,
 2: Standard setting (with advanced setting modification),
 3: Advanced setting (with advanced setting modification)

<OUT name (01h)>

Byte	Parameter details
1	OUT name, byte 1
2	OUT name, byte 2
3	OUT name, byte 3
to	to
20	OUT name, byte 20

OUT name OUT name string (20 characters max)
 * 0 is not appended to the end

<Minimum display unit (02h)>

Byte	Parameter details
1	Minimum display unit
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Minimum display unit
 The unit changes according to the measurement mode assigned to the OUT
 Length system 0: 1 mm, 1: 0.1 mm, 2: 0.01 mm, 3: 0.001 mm, 4: 1 μm, 5: 0.1 μm
 Area system 0: 1 mm², 1: 0.1 mm², 2: 0.01 mm², 3: 0.001 mm², 4: 0.0001 mm², 5: 0.00001 mm²
 Angle system 0: 1 deg, 1: 0.1 deg, 2: 0.01 deg

<Measurement mode (03h)>

Byte	Parameter details
1	Minimum display unit
2	Measurement mode
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Minimum display unit Length system
 0: 1 mm, 1: 0.1 mm, 2: 0.01 mm, 3: 0.001 mm, 4: 1 μm, 5: 0.1 μm
 Area system
 0: 1 mm², 1: 0.1 mm², 2: 0.01 mm², 3: 0.001 mm², 4: 0.0001 mm², 5: 0.00001 mm²
 Angle system
 0: 1 deg, 1: 0.1 deg, 2: 0.01 deg
 Measurement mode 0: Off, 1: Height, 2: Level difference, 3: Position, 4: Center position, 5: Width, 6: Thickness, 7: Angle, 8: R measurement, 9: Sectional area, 10: Master comparison (Z), 11: Distance (point-point), 12: Distance (point-line), 13: Height (profile compression (time axis) on), 14: Position (profile compression (time axis) on), 15: Deflection width (profile compression (time axis) on), 16: Height (simple 3D), 17: Level difference (simple 3D), 18: Position (simple 3D), 19: Operator

The unique parameters are assigned as shown below for each measurement mode.

0: OFF

Byte	Parameter details
5	Reserved (fixed as 0)
to	
92	

1: Height (when profile compression (time axis) is off)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Position correction selection
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement target head 0: Head A, 1: Head B,
2: Combined profile (only when profile combine (wide) is on)

Measurement target selection 0: Peak, 1: Bottom,
2: Inflection point,
3: Intersection point (line-line),
4: Tangent point (line-arc),
5: Center of a circle, 6: Average

Position correction selection 0: No position correction,
1: Position correction 1,
2: Position correction 2

Unique parameters are assigned to each measurement target.
For details on the unique parameters, see "Measurement Area Details" (page 84).

2: Level difference

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Position correction selection
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
M	
M+1	Reserved (fixed as 0)
to	
48	
49	Reference target head
50	Reference target selection
51	Position correction selection
52	Reserved (fixed as 0)
53	Unique parameters determined by the reference target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement or reference target head 0: Head A, 1: Head B,
2: Combined profile (only when profile combine (wide) is on)

Measurement or reference target selection 0: Peak, 1: Bottom, 2: Inflection point,
3: Intersection point (line-line),
4: Tangent point (line-arc), 5: Center of a circle,
6: Average

Position correction selection (common) 0: No position correction,
1: Position correction 1, 2: Position correction 2

Unique parameters are assigned to each measurement or reference target.
For details on the unique parameters, see "Measurement Area Details" (page 84).

3: Position (when profile compression (time axis) is off)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Position correction selection
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement target head 0: Head A, 1: Head B,
2: Combined profile (only when profile combine (wide) is on)

Measurement target selection 0: Peak, 1: Bottom,
2: Inflection point,
3: Intersection point (line-line),
4: Tangent point (line-arc),
5: Center of a circle, 7: Edge

Position correction selection 0: No position correction,
1: Position correction 1,
2: Position correction 2

Unique parameters are assigned to each measurement target.
For details on the unique parameters, see "Measurement Area Details" (page 84).

4: Center position

5: Width

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Position correction selection
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
M	
M+1	Reserved (fixed as 0)
to	
48	
49	Reference target head
50	Reference target selection
51	Position correction selection
52	Reserved (fixed as 0)
53	Unique parameters determined by the reference target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement or reference target head 0: Head A, 1: Head B,
2: Combined profile (only when profile combine (wide) is on)

Measurement or reference target selection 0: Peak, 1: Bottom, 2: Inflection point,
3: Intersection point (line-line),
4: Tangent point (line-arc), 5: Center of a circle,
7: Edge

Position correction selection (common) 0: No position correction, 1: Position correction 1,
2: Position correction 2

Unique parameters are assigned to each measurement or reference target.
For details on the unique parameters, see "Measurement Area Details" (page 84).

6: Thickness

Byte	Parameter details
5	Measurement target selection
6	Position correction selection (head A)
7	Position correction selection (head B)
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement target selection

8: Maximum thickness, 9: Minimum thickness,

10: Average thickness,

11: Maximum thickness position,

12: Minimum thickness position

Position correction selection (head A or B)

0: No position correction, 1: Position correction 1,

2: Position correction 2

Unique parameters are assigned to each measurement target.

For details on the unique parameters, see "Measurement Area Details" (page 84).

7: Angle

Byte	Parameter details	
5	Measurement target head	
6	Measurement reference selection	
7	Angle range	
8	Measurement target position correction	
9	Reference target position correction	
10	Reserved (fixed as 0)	
11	Reserved (fixed as 0)	
12	Reserved (fixed as 0)	
13	Linear calculation area count	Line 1
14	Reserved (fixed as 0)	
15	Reserved (fixed as 0)	
16	Reserved (fixed as 0)	
17	Area start position 1	
18		
19		
20		
21	Area end position 1	
22		
23		
24		
25	Area start position 2	
26		
27		
28		
29	Area end position 2	
30		
31		
32		
33	Linear calculation area count	Line 2
34	Reserved (fixed as 0)	
35	Reserved (fixed as 0)	
36	Reserved (fixed as 0)	
37	Area start position 1	
38		
39		
40		
41	Area end position 1	
42		
43		
44		
45	Area start position 2	
46		
47		
48		
49	Area end position 2	
50		
51		
52		
53	Reserved (fixed as 0)	
to		
92		

Measurement target head

0: Head A, 1: Head B,

2: Combined profile (only when profile combine (wide) is on)

Measurement reference selection

0: Horizontal reference, 1: Linear reference

Angle range

0: 0 to 180 deg, 1: -90 to 90 deg

Position correction selection (common)

0: No position correction, 1: Position correction 1,

2: Position correction 2

Linear calculation area count

0: Area 2 disabled, 1: Area 2 enabled

Area start position 1/2
Specify the position in the range set for each head

Area end position 1/2
Specify the position in the range set for each head

8: R measurement

Byte	Parameter details
5	Measurement target head
6	Position correction selection
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Arc calculation area count
10	Reserved (fixed as 0)
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)
13	Area start position 1
14	
15	
16	
17	Area end position 1
18	
19	
20	
21	Area start position 2
22	
23	
24	
25	Area end position 2
26	
27	
28	
29	Reserved (fixed as 0)
to	
92	

Measurement target head 0: Head A, 1: Head B, 2: Combined profile (only when profile combine (wide) is on)

Position correction selection 0: No position correction, 1: Position correction 1, 2: Position correction 2

Arc calculation area count 0: Area 2 disabled, 1: Area 2 enabled

Area start position 1/2 Specify the position in the range set for each head

Area end position 1/2 Specify the position in the range set for each head

9: Sectional area

Byte	Parameter details
5	Measurement target head
6	Measurement reference selection
7	Measurement target position correction
8	Reference target 1 position correction
9	Reference target 2 position correction
10	Reserved (fixed as 0)
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)
13	Measurement area start position
14	
15	
16	
17	Measurement area end position
18	
19	
20	
21	Linear calculation area count
22	Reserved (fixed as 0)
23	Reserved (fixed as 0)
24	Reserved (fixed as 0)
25	Area start position 1
26	
27	
28	
29	Area end position 1
30	
31	
32	
33	Area start position 2
34	
35	
36	
37	Area end position 2
38	
39	
40	
41	Linear calculation area count
42	Reserved (fixed as 0)
43	Reserved (fixed as 0)
44	Reserved (fixed as 0)
45	Area start position 1
46	
47	
48	
49	Area end position 1
50	
51	
52	
53	Area start position 2
54	
55	
56	
57	Area end position 2
58	
59	
61	
61	Reserved (fixed as 0)
to	
92	

Measurement target head 0: Head A, 1: Head B, 2: Combined profile (only when profile combine (wide) is on)

Measurement reference selection 0: Reference straight line 1, 1: Reference straight line 2, 2: Master reference

Position correction selection (common) 0: No position correction, 1: Position correction 1, 2: Position correction 2

Linear calculation area count

0: Area 2 disabled, 1: Area 2 enabled

Area start position 1/2

Specify the position in the range set for each head

Area end position 1/2

Specify the position in the range set for each head

10: Master comparison (Z)

Byte	Parameter details
5	Measurement target selection
6	Position correction selection
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Area start position
10	
11	
12	
13	Area end position
14	
15	
16	
17	Reserved (fixed as 0)
to	
92	

Measurement target selection 8: Maximum thickness,
9: Minimum thickness,
10: Average thickness,
11: Maximum thickness
position,
12: Minimum thickness position

Position correction selection 0: No position correction,
1: Position correction 1,
2: Position correction 2

Area start position Specify the position in the
range set for each head

Area end position Specify the position in the
range set for each head

11: Distance (point-point)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Position correction selection
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
M	
M+1	Reserved (fixed as 0)
to	
48	
49	Reference target head
50	Reference target selection
51	Position correction selection
52	Reserved (fixed as 0)
53	Unique parameters determined by the reference target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement or reference target head
0: Head A, 1: Head B,
2: Combined profile (only when profile combine
(wide) is on)

Measurement or reference target selection
0: Peak, 1: Bottom, 2: Inflection point,
3: Intersection point (line-line),
4: Tangent point (line-arc), 5: Center of a circle

Position correction selection (common)
0: No position correction, 1: Position correction 1,
2: Position correction 2

Unique parameters are assigned to each
measurement or reference target.
For details on the unique parameters, see
"Measurement Area Details" (page 84).

12: Distance (point-line)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Position correction selection
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
M	
M+1	Reserved (fixed as 0)
to	
48	
49	Reference target head
50	Position correction selection
51	Reserved (fixed as 0)
52	Reserved (fixed as 0)
53	Linear calculation area count
54	Reserved (fixed as 0)
55	Reserved (fixed as 0)
56	Reserved (fixed as 0)
57	Area start position 1
58	
59	
60	
61	Area end position 1
62	
63	
64	
65	Area start position 2
66	
67	
68	
69	Area end position 2
70	
71	
72	
73	Reserved (fixed as 0)
to	
92	

Measurement or reference target head
0: Head A, 1: Head B,
2: Combined profile (only when profile combine (wide) is on)

Measurement target selection
0: Peak, 1: Bottom, 2: Inflection point,
3: Intersection point (line-line),
4: Tangent point (line-arc), 5: Center of a circle

Position correction selection (common)
0: No position correction, 1: Position correction 1,
2: Position correction 2

Linear calculation area count
0: Area 2 disabled, 1: Area 2 enabled

Area start position 1/2
Specify the position in the range set for each head

Area end position 1/2
Specify the position in the range set for each head

Unique parameters are assigned to each measurement target.
For details on the unique parameters, see "Measurement Area Details" (page 84).

13: Height (when profile compression (time axis) is on)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement target head 0: Head A, 1: Head B,
2: Combined profile (only when profile combine (wide) is on)

Measurement target selection 0: Peak, 1: Bottom,
13: Middle value

Unique parameters are assigned to each measurement target.
For details on the unique parameters, see "Measurement Area Details" (page 84).

14: Position (when profile compression (time axis) is on)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement target head 0: Head A, 1: Head B,
2: Combined profile (only when profile combine (wide) is on)

Measurement target selection 0: Peak, 1: Bottom, 7: Edge

Unique parameters are assigned to each measurement target.
For details on the unique parameters, see "Measurement Area Details" (page 84).

15: Deflection width (when profile compression (time axis) is on)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Measurement target head 0: Head A, 1: Head B,
2: Combined profile (only when
profile combine (wide) is on)

Measurement target selection 14: P-P (Z), 15: P-P (X)

Unique parameters are assigned to each measurement target.

For details on the unique parameters, see "Measurement Area Details" (page 84).

16: Height (simple 3D)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
M	
M+1	Reserved (fixed as 0)
to	
16	
17	Y-coordinate start position
18	
19	Y-coordinate end position
20	
21	Reserved (fixed as 0)
to	
92	

Measurement target head 0: Head A, 1: Head B,
2: Combined profile (only when
profile combine (wide) is on)

Measurement target selection 0: Peak, 1: Bottom, 6: Average,
16: P-P

Y-coordinate start position 1 to batch count

Y-coordinate end position 1 to batch count

Unique parameters are assigned to each measurement target.

For details on the unique parameters, see "Measurement Area Details" (page 84).

17: Level difference (simple 3D)

Byte	Parameter details
5	Measurement target head
6	Measurement target selection
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
M	
M+1	Reserved (fixed as 0)
to	
16	
17	Y-coordinate start position
18	
19	Y-coordinate end position
20	
21	Reference target head
22	Reference target selection
23	Reserved (fixed as 0)
24	Reserved (fixed as 0)
25	Unique parameters determined by the reference target
to	
M	
M+1	Reserved (fixed as 0)
to	
32	
33	Y-coordinate start position
34	
35	Y-coordinate end position
36	
37	Reserved (fixed as 0)
to	
92	

Measurement or reference target head
0: Head A, 1: Head B,
2: Combined profile (only when profile combine
(wide) is on)

Measurement or reference target selection

0: Peak, 1: Bottom, 6: Average

Y-coordinate start position 1 to batch count

Y-coordinate end position 1 to batch count

Unique parameters are assigned to each measurement target.

For details on the unique parameters, see "Measurement Area Details" (page 84).

18: Position (simple 3D)

Byte	Parameter details
5	Measurement target head
6	Output coordinate
7	Measurement target selection
8	Reserved (fixed as 0)
9	Unique parameters determined by the measurement target
to	
M	
M+1	Reserved (fixed as 0)
to	
16	
17	Y-coordinate start position
18	
19	Y-coordinate end position
20	
21	Reserved (fixed as 0)
to	
92	

Measurement target head	0: Head A, 1: Head B, 2: Combined profile (only when profile combine (wide) is on)
Output coordinate	0: X coordinate, 1: Y coordinate
Measurement target selection	0: Peak, 1: Bottom
Y-coordinate start position	1 to batch count
Y-coordinate end position	1 to batch count

Unique parameters are assigned to each measurement target.

For details on the unique parameters, see "Measurement Area Details" (page 84).

19: Operator

Byte	Parameter details
5	Operator classification
6	Reserved (fixed as 0)
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Unique parameters determined by the operator classification
to	
N	
N+1	Reserved (fixed as 0)
to	
92	

Operator classification 0: Addition, 1: Subtraction,
2: Absolute value, 3: AVE, 4: P-P,
5: MAX, 6: MIN

Unique parameters are assigned to each operand.
The unique parameters are described below.

0: Addition

1: Subtraction

Byte	Parameter details
9	Operand A
10	Operand B
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)

Operand A and B The OUT numbers that correspond to the operands for the A + B and A - B operations

2: Absolute value

Byte	Parameter details
9	Operand
10	Reserved (fixed as 0)
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)

Operand The OUT number that corresponds to the operand for the absolute value operation

3: AVE

4: P-P

5: MAX

6: MIN

Byte	Parameter details
9	OUT1
10	OUT2
11	OUT3
12	OUT4
13	OUT5
14	OUT6
15	OUT7
16	OUT8
17	OUT9
18	OUT10
19	OUT11
20	OUT12
21	OUT13
22	OUT14
23	OUT15
24	OUT16

OUT1 to OUT16 0: Do not use as an operand,
1: Use as an operand

<Measurement value hold count (04h)>

Byte	Parameter details
1	Measurement value hold count
2	
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Measurement value hold count 0 to 999

<Measurement value filter (05h)>

Byte	Parameter details
1	Measurement value filter
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Measurement value filter 0: Off, 1: Moving average,
2: Low-pass filter, 3: High-pass filter

<Measurement value filter detail (06h)>

Unique parameters are assigned to each measurement value filter.

1: Moving average

Byte	Parameter details
1	Average count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Average count 0: 4 times, 1: 16 times, 2: 64 times,
3: 256 times, 4: 1024 times, 5: 4096 times

2: Low-pass filter

3: High-pass filter

Byte	Parameter details
1	Cutoff frequency
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Cutoff frequency 0: 0.1 Hz, 1: 0.3 Hz, 2: 1 Hz,
3: 3 Hz, 4: 10 Hz, 5: 30 Hz,
6: 100 Hz, 7: 300 Hz, 8: 1000 Hz

<Scaling measurement value 1 (07h)>

<Scaling display value 1 (08h)>

<Scaling measurement value 2 (09h)>

<Scaling display value 2 (0Ah)>

Byte	Parameter details
1	Measurement value
2	
3	
4	

Measurement value Display range lower limit for the
minimum display unit \leq relevant value \leq
display range upper limit for the
minimum display unit
* However, for the mm unit in the length
system, a range of -10000 mm <
relevant value < +10000 mm

<Measuring mode (0Bh)>

Byte	Parameter details
1	Measuring mode
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Measuring mode 0: Normal, 1: Peak hold, 2: Bottom hold,
3: Peak to peak hold,
4: Average hold, 5: Sample hold,
6: Peak, 7: Bottom, 8: Peak to peak,
9: Average

<Measuring range (0Ch)>

Byte	Parameter details
1	Measuring range
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Measuring range 0: Terminal/command,
1: Measuring area, 2: OUT reference,
3: Internal timing (level),
4: Internal timing (edge)

The payload content for the measuring range details changes according to the measuring range value.

■ Terminal/command

Byte	Parameter details
5	Reserved (fixed as 0)
6	Reserved (fixed as 0)
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Reserved (fixed as 0)
10	Reserved (fixed as 0)
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)
13	Reserved (fixed as 0)
14	Reserved (fixed as 0)
15	Reserved (fixed as 0)
16	Reserved (fixed as 0)

None

■ Measuring area

Byte	Parameter details
5	Measuring start position
6	
7	Measuring end position
8	
9	Reserved (fixed as 0)
10	Reserved (fixed as 0)
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)
13	Reserved (fixed as 0)
14	Reserved (fixed as 0)
15	Reserved (fixed as 0)
16	Reserved (fixed as 0)

Measuring start position 1 to batch count
Measuring end position 1 to batch count

■ OUT reference

Byte	Parameter details
5	Reference OUT
6	Reserved (fixed as 0)
7	Reserved (fixed as 0)
8	Reserved (fixed as 0)
9	Reserved (fixed as 0)
10	Reserved (fixed as 0)
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)
13	Reserved (fixed as 0)
14	Reserved (fixed as 0)
15	Reserved (fixed as 0)
16	Reserved (fixed as 0)

Reference OUT 0: OUT1, 1: OUT2,
2: OUT3, and so on to 15: OUT16

■ Internal timing (level)

Byte	Parameter details
5	Upper limit value
6	
7	
8	
9	Lower limit value
10	
11	
12	
13	Reserved (fixed as 0)
14	Reserved (fixed as 0)
15	Reserved (fixed as 0)
16	Reserved (fixed as 0)

Upper limit value/lower limit value Display range lower limit for the minimum display unit \leq relevant value \leq display range upper limit for the minimum display unit

* However, for the mm unit in the length system, a range of -10000 mm < relevant value < +10000 mm

* The minimum unit is the same as the minimum display unit setting

■ Internal timing (edge)

Byte	Parameter details
5	Edge threshold
6	
7	
8	
9	Edge direction
10	Reserved (fixed as 0)
11	Reserved (fixed as 0)
12	Reserved (fixed as 0)
13	Measurement count
14	
15	
16	

Edge threshold Display range lower limit for the minimum display unit \leq relevant value \leq display range upper limit for the minimum display unit

* However, for the mm unit in the length system, a range of -10000 mm < relevant value < +10000 mm

* The minimum unit is the same as the minimum display unit setting

Edge direction 0: Rising, 1: Falling

Measurement count When batch off: Integer from 1 to 999,999
When batch on: Integer from 1 to batch count

<Offset (0Dh)>

Byte	Parameter details
1	Offset
2	
3	
4	

Offset Display range lower limit for the minimum display unit \leq relevant value \leq display range upper limit for the minimum display unit

* However, for the mm unit in the length system, a range of -10000 mm < relevant value < +10000 mm

<Upper limit (0Eh)><Lower limit (0Fh)>

Byte	Parameter details
1	Tolerance upper limit/lower limit
2	
3	
4	

Tolerance upper limit/lower limit Display range lower limit for the minimum display unit \leq relevant value \leq display range upper limit for the minimum display unit

* However, for the mm unit in the length system, a range of -10000 mm \leq relevant value \leq +10000 mm

<Hysteresis (10h)>

Byte	Parameter details
1	Hysteresis
2	
3	
4	

Hysteresis 0 \leq relevant value \leq display range upper limit for the minimum display unit

* However, for the mm unit in the length system, a range of 0 < relevant value < +10000 mm

<ZERO (11h)>

Byte	Parameter details
1	ZERO
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

ZERO 0: None, 1: ZERO1, 2: ZERO2

<TIMING/RESET (12h)>

Byte	Parameter details
1	TIMING/RESET
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

TIMING/RESET 0: None, 1: TIMING1/RESET1,
2: TIMING2/RESET2

<Auto zero base value (13h)>

Byte	Parameter details
1	Auto zero base value
2	
3	
4	

Auto zero base value Display range lower limit for the minimum display unit \leq relevant value \leq display range upper limit for the minimum display unit

* However, for the mm unit in the length system, a range of -10000 mm < relevant value < +10000 mm

Program (specify "10h to 1Fh" for setting type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Terminal settings (07h)

Specify the value for the setting item and for setting target 1 to setting target 4 for the terminal settings according to the table below.

			Setting target 1	Setting target 2	Setting target 3	Setting target 4
Setting item	00 h	Standard setting/advanced setting	Ignored	Ignored	Ignored	Ignored
	01 h	Judgment output condition	Condition number: 0 to 11	Ignored	Ignored	Ignored
	02 h	Target OUT	Analog output: 0 or 1	Ignored	Ignored	Ignored
	03 h	Scaling	Analog output: 0 or 1	Ignored	Ignored	Ignored

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting

0: Standard setting, 1: Advanced setting,
2: Standard setting (with advanced setting modification), 3: Advanced setting (with advanced setting modification)

<Judgment output condition (01h)>

Byte	Parameter details
1	Condition
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	OUT1 judgment result
6	OUT2 judgment result
7	OUT3 judgment result
to	to
20	OUT16 judgment result

Condition 0: None, 1: AND, 2: OR
OUT1 to 16 judgment condition 0: Not specified,
2: HI, 4: GO, 8: LO

* The judgment result can be specified by bits with logical OR
When HI and GO are both specified, the measurement result value is "6"

<Target OUT (02h)>

Byte	Parameter details
1	Target OUT
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Target OUT 0: OUT1, 1: OUT2, 2: OUT3, and so on to
15: OUT16, 255: none

<Scaling (03h)>

Byte	Parameter details
1	OUT display value 1
2	
3	
4	
5	Output voltage 1
6	
7	
8	
9	OUT display value 2
10	
11	
12	
13	Output voltage 2
14	
15	
16	

OUT display value 1/2

Length (mm) -999.999 mm to 999.999 mm
Area (mm²) -9999.99mm² to 9999.99 mm²
Angle (deg)..... -9999.99 deg to 9999.99 deg

Length (mm) 0.01 μm units
Area (mm²) 0.00001 mm² units
Angle (deg)..... 0.001 deg units

Output voltage -10.5 V to 10.5 V (1mV units)

Program (specify "10h to 1Fh" for setting type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Storage settings (08h)

Always specify 0 for the setting item and setting target 1 to setting target 4 for the storage settings.

<Standard setting/advanced setting (00h)>

Byte	Parameter details
1	Standard setting/advanced setting
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Standard setting/advanced setting

0: Standard setting, 1: Advanced setting,
2: Standard setting (with advanced setting
modification), 3: Advanced setting (with
advanced setting modification)

<Storage target (01h)>

Byte	Parameter details
1	Storage target
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Storage target 0: Off, 1: OUT value, 2: Profile

<Storage condition (02h)>

Byte	Parameter details
1	Storage condition
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)

Storage condition 0: Terminal/command, 1: At OUT update,
2: OUT measurement value edge, 3: OUT
measurement value level

<Terminal/command detail (03h)>

Byte	Parameter details
1	Storage data amount
2	
3	
4	

Storage data amount 0 to buffer size upper limit

<OUT measurement value edge detail (04h)>

Byte	Parameter details
1	Storage data amount
2	
3	
4	
5	Threshold
6	
7	
8	
9	Hysteresis
10	
11	
12	
13	Target OUT
14	Edge direction
15	Reserved (fixed as 0)
16	Reserved (fixed as 0)

Storage data amount 0 to buffer size upper limit

Threshold The range that can be input in the OUT
minimum display unit

Hysteresis $0 \leq \text{relevant value} \leq \text{display range}$
upper limit for the minimum display unit

Target OUT 0: OUT1, 1: OUT2,
2: OUT3, and so on to 15: OUT16

Edge direction 0: Rising, 1: Falling

<OUT measurement value level detail (05h)>

Byte	Parameter details
1	Upper limit value
2	
3	
4	
5	Lower limit value
6	
7	
8	
13	Target OUT
14	Reserved (fixed as 0)
15	Reserved (fixed as 0)
16	Reserved (fixed as 0)

Upper limit value The range that can be input in the OUT
minimum display unit

Lower limit value The range that can be input in the OUT
minimum display unit

Target OUT 0: OUT1, 1: OUT2,
2: OUT3, and so on to 15: OUT16

Program (specify "10h to 1Fh" for setting type, 10h: program 0, 11h: program 1, and so on to 1Fh: program 15) → Program name (09h)

Specify 0 for the setting item for the program name.

Always specify 0 for setting target 1 to setting target 4 for the program name.

<Program name (00h)>

Byte	Parameter details
1	Program name, byte 1
2	Program name, byte 2
3	Program name, byte 3
to	to
24	Program name, byte 24

Program name Program name string (24 characters max)
* 0 is not appended to the end

■ Measurement Area Details

The correction target selection of feature point correction of position correction and the unique parameters of the measurement target of the measurement mode are shown below.

The byte numbers shown here indicate the byte numbers from the start of the corresponding data block.

- 0: Peak**
- 1: Bottom**
- 6: Average**
- 8: Maximum thickness**
- 9: Minimum thickness**
- 10: Average thickness**
- 11: Maximum thickness position**
- 12: Minimum thickness position**
- 13: Middle value**
- 14: P-P (Z; only when profile compression (time axis) is on)**
- 16: P-P (only for the height -simple 3D)**

Byte	Parameter details
1	Area start position
2	
3	
4	
5	Area end position
6	
7	
8	

Area start position Specify the position in the range set for each head

Area end position Specify the position in the range set for each head

2: Inflection point

Byte	Parameter details
1	Area start position
2	
3	
4	
5	Area end position
6	
7	
8	
9	Inflection point shape
10	Detection direction
11	Detection number
12	Sensitivity

Area start position Specify the position in the range set for each head

Area end position Specify the position in the range set for each head

Inflection point shape 0: Valley, 1: Peak

Detection direction 0: + direction, 1: - direction

Detection number 1 to 10

Sensitivity 0 to 100

3: Intersection point (line-line)

Byte	Parameter details
1	Linear calculation area count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	Area start position 1
6	
7	
8	
9	Area end position 1
10	
11	
12	
13	Area start position 2
14	
15	
16	
17	Area end position 2
18	
19	
20	
21	Linear calculation area count
22	Reserved (fixed as 0)
23	Reserved (fixed as 0)
24	Reserved (fixed as 0)
25	Area start position 1
26	
27	
28	
29	Area end position 1
30	
31	
32	
33	Area start position 2
34	
35	
36	
37	Area end position 2
38	
39	
40	

Line 1

Line 2

Linear calculation area count 0: Area 2 disabled, 1: Area 2 enabled

Area start position 1/2 Specify the position in the range set for each head

Area end position 1/2 Specify the position in the range set for each head

4: Tangent point (line-arc)

Byte	Parameter details	
1	Linear calculation area count	Line
2	Reserved (fixed as 0)	
3	Reserved (fixed as 0)	
4	Reserved (fixed as 0)	
5	Area start position 1	
6		
7		
8		
9	Area end position 1	
10		
11		
12		
13	Area start position 2	
14		
15		
16		
17	Area end position 2	
18		
19		
20		
21	Arc calculation area count	Arc
22	Reserved (fixed as 0)	
23	Reserved (fixed as 0)	
24	Reserved (fixed as 0)	
25	Area start position 1	
26		
27		
28		
29	Area end position 1	
30		
31		
32		
33	Area start position 2	
34		
35		
36		
37	Area end position 2	
38		
39		
40		

Linear calculation area count 0: Area 2 disabled,
1: Area 2 enabled

Arc calculation area count 0: Area 2 disabled,
1: Area 2 enabled

Area start position 1/2 Specify the position in the range
set for each head

Area end position 1/2 Specify the position in the range
set for each head

5: Center of a circle

Byte	Parameter details
1	Arc calculation area count
2	Reserved (fixed as 0)
3	Reserved (fixed as 0)
4	Reserved (fixed as 0)
5	Area start position 1
6	
7	
8	
9	Area end position 1
10	
11	
12	
13	Area start position 2
14	
15	
16	
17	Area end position 2
18	
19	
20	

Arc calculation area count 0: Area 2 disabled,
1: Area 2 enabled

Area start position 1/2 Specify the position in the
range set for each head

Area end position 1/2 Specify the position in the
range set for each head

7: Edge (when profile compression (time axis) is off)

Byte	Parameter details
1	Area start position
2	
3	
4	
5	Area end position
6	
7	
8	
9	Edge direction
10	Detection direction
11	Detection number
12	Reserved (fixed as 0)
13	Edge level
14	
15	
16	

Area start position Specify the position in the range set for
each head

Area end position Specify the position in the range set for
each head

Edge direction 0: Rising, 1: Falling

Detection direction 0: + direction, 1: - direction

Detection number 1 to 10

Edge level 32-bit signed integer value

7: Edge (when profile compression (time axis) is on)

Byte	Parameter details
1	Area start position
2	
3	
4	
5	Area end position
6	
7	
8	
9	Detection target
10	Edge direction
11	Detection direction
12	Detection number
13	Edge level
14	
15	
16	

Area start position Specify the position in the range set for each head

Area end position Specify the position in the range set for each head

Detection target 0: Upper profile, 1: Lower profile

Edge direction 0: Rising, 1: Falling

Detection direction 0: + direction, 1: - direction

Detection number 1 to 10

Edge level 32-bit signed integer value

Byte	Parameter details
1	Area start position
2	
3	
4	
5	Area end position
6	
7	
8	
9	Edge direction
10	Detection direction
11	Detection number
12	Reserved (fixed as 0)
13	Edge level
14	
15	
16	

Area start position Specify the position in the range set for each head

Area end position Specify the position in the range set for each head

Edge direction 0: Rising, 1: Falling

Detection direction 0: + direction, 1: - direction

Detection number 1 to 10

Edge level 32-bit signed integer value

15: P-P (X; only when profile compression (time axis) is on)

Byte	Parameter details
1	Area start position
2	
3	
4	
5	Area end position
6	
7	
8	
9	Edge direction
10	Detection direction
11	Detection number
12	Reserved (fixed as 0)
13	Edge level
14	
15	
16	

Area start position Specify the position in the range set for each head

Area end position Specify the position in the range set for each head

Edge direction 0: Rising, 1: Falling

Detection direction 0: + direction, 1: - direction

Detection number 1 to 10

Edge level 32-bit signed integer value

Revision date	Revision number	Revision details
June 2012	2nd edition	Ver. 2
October 2012	Revised 1st edition	Z-phase specifications were added.

WARRANTIES AND DISCLAIMERS

- (1) KEYENCE warrants the Products to be free of defects in materials and workmanship for a period of one (1) year from the date of shipment. If any models or samples were shown to Buyer, such models or samples were used merely to illustrate the general type and quality of the Products and not to represent that the Products would necessarily conform to said models or samples. Any Products found to be defective must be shipped to KEYENCE with all shipping costs paid by Buyer or offered to KEYENCE for inspection and examination. Upon examination by KEYENCE, KEYENCE, at its sole option, will refund the purchase price of, or repair or replace at no charge any Products found to be defective. This warranty does not apply to any defects resulting from any action of Buyer, including but not limited to improper installation, improper interfacing, improper repair, unauthorized modification, misapplication and mishandling, such as exposure to excessive current, heat, coldness, moisture, vibration or outdoors air. Components which wear are not warranted.
- (2) KEYENCE is pleased to offer suggestions on the use of its various Products. They are only suggestions, and it is Buyer's responsibility to ascertain the fitness of the Products for Buyer's intended use. KEYENCE will not be responsible for any damages that may result from the use of the Products.
- (3) The Products and any samples ("Products/Samples") supplied to Buyer are not to be used internally in humans, for human transportation, as safety devices or fail-safe systems, unless their written specifications state otherwise. Should any Products/Samples be used in such a manner or misused in any way, KEYENCE assumes no responsibility, and additionally Buyer will indemnify KEYENCE and hold KEYENCE harmless from any liability or damage whatsoever arising out of any misuse of the Products/Samples.
- (4) **OTHER THAN AS STATED HEREIN, THE PRODUCTS/SAMPLES ARE PROVIDED WITH NO OTHER WARRANTIES WHATSOEVER. ALL EXPRESS, IMPLIED, AND STATUTORY WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF PROPRIETARY RIGHTS, ARE EXPRESSLY DISCLAIMED. IN NO EVENT SHALL KEYENCE AND ITS AFFILIATED ENTITIES BE LIABLE TO ANY PERSON OR ENTITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, PUNITIVE, SPECIAL OR CONSEQUENTIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, ANY DAMAGES RESULTING FROM LOSS OF USE, BUSINESS INTERRUPTION, LOSS OF INFORMATION, LOSS OR INACCURACY OF DATA, LOSS OF PROFITS, LOSS OF SAVINGS, THE COST OF PROCUREMENT OF SUBSTITUTED GOODS, SERVICES OR TECHNOLOGIES, OR FOR ANY MATTER ARISING OUT OF OR IN CONNECTION WITH THE USE OR INABILITY TO USE THE PRODUCTS, EVEN IF KEYENCE OR ONE OF ITS AFFILIATED ENTITIES WAS ADVISED OF A POSSIBLE THIRD PARTY'S CLAIM FOR DAMAGES OR ANY OTHER CLAIM AGAINST BUYER.** In some jurisdictions, some of the foregoing warranty disclaimers or damage limitations may not apply.

BUYER'S TRANSFER OBLIGATIONS:

If the Products/Samples purchased by Buyer are to be resold or delivered to a third party, Buyer must provide such third party with a copy of this document, all specifications, manuals, catalogs, leaflets and written information provided to Buyer pertaining to the Products/Samples.

Specifications are subject to change without notice.

KEYENCE CORPORATION

1-3-14, Higashi-Nakajima, Higashi-Yodogawa-ku, Osaka, 533-8555, Japan PHONE: +81-6-6379-2211

www.keyence.com

AUSTRIA

Phone: +43 22 36-3782 66-0

BELGIUM

Phone: +32 1 528 1222

BRAZIL

Phone: +55-11-3045-4011

CANADA

Phone: +1-905-366-7655

CHINA

Phone: +86-21-68757500

CZECH REPUBLIC

Phone: +420 222 191 483

FRANCE

Phone: +33 1 56 37 78 00

GERMANY

Phone: +49 6102 36 89-0

HONG KONG

Phone: +852-3104-1010

HUNGARY

Phone: +36 1 802 73 60

INDIA

Phone: +91-44-4299-4192

ITALY

Phone: +39-02-6688220

KOREA

Phone: +82-31-789-4300

MALAYSIA

Phone: +60-3-2092-2211

MEXICO

Phone: +52-81-8220-7900

NETHERLANDS

Phone: +31 40 20 66 100

POLAND

Phone: +48 71 36861 60

ROMANIA

Phone: +40 269-232-808

SINGAPORE

Phone: +65-6392-1011

SLOVAKIA

Phone: +421 2 5939 6461

SLOVENIA

Phone: +386 1-4701-666

SWITZERLAND

Phone: +41 43-45577 30

TAIWAN

Phone: +886-2-2718-8700

THAILAND

Phone: +66-2-369-2777

UK & IRELAND

Phone: +44-1908-696900

USA

Phone: +1-201-930-0100

