



**SG/SGI/SC/SCI Series
Communication Library**

— Reference Manual

V2.1

June 2024

www.cnsszn.com

Software Product License Agreement

Before using this software, customers are required to agree with the following software product license agreement (hereinafter referred to as "the Agreement") to use the software. By using or replicating whole or a part of the software, customers are deemed to agree with the Agreement, and the Agreement become valid.

I. Licensing the right of use)

On the premise that the customer complies with the content of this Agreement, Shenzhen SinceVision Technology Co., Ltd. (hereinafter referred to as "the Company") will grant the customer a non-exclusive right to use this software.

II. Restrictions

Customers are not allowed to perform the following operations or modifications on this software.

- a. Modification of the functions of the software or addition of a function to the software.
- b. All reverse engineering including reverse compiling and reverse assembling.
- c. Resale, delivery, redistribution, licensing to use, lending, or leasing of the software and the license key, etc. (But customers are allowed to redistribute this software together with applications written through this software.)

III. Intellectual Property Rights

Intellectual properties such as a copyright on the software and the manual for the software shall be reserved by the Company.

IV. Disclaimer

The Company shall have no responsibility for any damages of customers or third parties results from use of the software.

V. Support

the Company shall provide technical support upon customer's questions on the software under the Agreement. The technical support, however, does not provide any guarantees of achievement of customer's purposes.

VI. Termination of the Agreement

1. The Agreement shall terminate automatically at the time when customers stop to use of the software by way to discard the software and the replicas of the software.
2. The Company can terminate the agreement unilaterally when customers are against any one of the terms of the agreement. In this case, customers must return or discard the software and replicas immediately to SSZN.
3. Customers shall pay for damages arising to SSZN out of infringements of the agreement by customers.

VII. Applicable law

The Agreement shall be governed by the laws of China.

Introduction

Be sure to read this manual before use.

After reading this manual, please keep it in a safe place for future reference.

SG series point laser communication library provides customers with communication interface to control SG series point laser through application programs. For specific usage method, please ask our company for a sample instance.

An utmost effort has been made to ensure the contents of this manual are as complete as possible. If there are any mistakes or question, please contact SSZN sales division. Any manuals with missing pages or other paging faults will be replaced.

Contact Us

SHENZHEN SINCEVISION TECHNOLOGY CO., LTD.

Telephone: 0755-29655424

Fax: 0755-27369408

Email: support@cnsszn.com

Website: <http://www.cnsszn.com>

Address: 5th floor, Building 2, Chongwen Park, No. 3370 Liuxian Avenue, Nanshan Zhiyuan,
Nanshan District, Shenzhen.

Postal Code: 518071

Thank you for choosing SSZN machine vision relevant products

To repay customers, we will help you establish your own machine vision system with our first-class machine vision products, perfect after-sale service, and high-efficiency technical support.

More information about SSZN products

Please visit our website at <http://www.cnsszn.com> for more information about our company and products, including company introduction, product introduction, technical support, product latest release and so on.

You can also consult the telephone (0755-29655425) for more information about our company and product.

Technical Support and After-sale Service

You may get our technical support and after-sale service through the following approaches:

Email: support@cnsszn.com

Telephone: 0755-29655424

Address: 5th floor, Building 2, Chongwen Park, No. 3370 Liuxian Avenue, Nanshan Zhiyuan, Nanshan District, Shenzhen

Postal Code: 518071

Purpose of this User's Manual

By reading this manual, users can understand the basic usage of communication library of the machine vision product SG series point laser, and can do the development in actual engineering projects.

User of this User's Manual

This manual is suitable to those engineering personnel who have the basic knowledge of software and software debugging.

Main Contents of this User's Manual

This manual consists of 8 sections. The operating environment, file composition, embedding method, variable types, structure definitions, functions detailed descriptions, development routines, etc. of the SR Series 3D Camera Communication Library are described in detail.

Relevant Documents

For hardware installation of SG Series point laser, please refer to the *SG Series_Laser Displacement Sensor_Hardware User Manual* that comes with this product. For software debugging of SG Series point laser, please refer to the *SG Series_Point Laser_Software User Manual* that comes with this product.

Document Version

Number	Version	Revision Date
	1.0	April 10th, 2021
	2.0	May 7th, 2024
	2.1	June 6th, 2024
	2.2	July 19, 2024 Added “Active OUT Count Acquisition” “Active Sensor Head Count Acquisition”

Contents

1	Operating Environment	6
1.1	Execution Environment	6
2	File Structure	7
3	Embedding Method	7
3.1	C++	7
4	Structure Definitions	8
5	Function Return Code Definition	10
6	Function	12
	Example flow chart: obtaining a single measurement value	12
	Example flow chart: Data Storage/Data Output	13
	Example flow chart: Setting Change/Setting Acquisition	14
6.1	Overview of Functions	15
6.1.1	Establish/disconnect the communication path with the controller	15
6.1.2	Mode Change	15
6.1.3	Measurement Control	15
6.1.4	Setting Change	16
6.1.5	Setting Acquisition	17
6.2	Function Reference	19
6.2.1	Establish/disconnect communication connections to the Controller	19
6.2.2	Mode Change	20
6.2.3	Measurement Control	21
6.2.4	Setting Change	32
6.2.5	Setting Acquisition	56

1 Operating Environment

Operating System	Windows 10 32/64 (Professional/Enterprise) Windows 8.1 32/64 (Professional/Enterprise) Windows7 32/64 (SP1 or higher version) (Professional/Ultimate)
CPU	Core i3 2.3 GHz or more fast CPU
RAM	8 GB or more
Secondary cache memory	2 MB or more
ROM	10 GB or more
Interface	Ethernet 1000BASE-T/100BASE-TX

* We cannot guarantee the operating conditions when connecting to LAN or a router.

1.1 Execution Environment

The following is the basic environment for executing applications when the communication library of SG series point laser.

1. Microsoft C Runtime Library

This is the necessary running environment for executing DLL.

2 File Structure

SGIFPJ.dll	DLL file
SGIFPJ.lib	The import library for SGIFPJ.dll
SGIF.h	The header file that defines the communication library.

3 Embedding Method

3.1 C++

1. Linking

The library can be linked implicitly or explicitly.

To implicitly link the library, link with “SR7Link.lib”

2. Included Files

Please make sure to place the following header files in the source file.

- **SGIF.h**

4 Structure Definitions

Nmae	Ethernet setting structure
Definition	<pre>typedef struct { IN_ADDR IPAddress; } SGIF_OPENPARAM_ETHERNET;</pre>
Description	Communication settings for Ethernet communication connections. IPAddress The IP address of the connected controller. When the address is 192.168.0.10 Set IPAddress.S_un.S_un_b.s_b1=192 IPAddress.S_un.S_un_b.s_b2=168 IPAddress.S_un.S_un_b.s_b3=0 IPAddress.S_un.S_un_b.s_b4=10 and so on.
Remark	--

Nmae	Structure of the measured value						
Definition	<pre>typedef struct { int OutNo; SGIF_FLOATRESULT FloatResult; float Value; } SGIF_FLOATVALUE_OUT;</pre>						
Description	Structure of the measured value. Variable description: <table><tr><td>int OutNo;</td><td>//Sensor head (1-4)</td></tr><tr><td>SGIF_FLOATRESULT FloatResult;</td><td>// Valid or invalid data</td></tr><tr><td>float Value;</td><td>// Measured value</td></tr></table>	int OutNo;	//Sensor head (1-4)	SGIF_FLOATRESULT FloatResult;	// Valid or invalid data	float Value;	// Measured value
int OutNo;	//Sensor head (1-4)						
SGIF_FLOATRESULT FloatResult;	// Valid or invalid data						
float Value;	// Measured value						
Remark	--						

Nmae	Structure of the measured value
Definition	<pre>typedef struct { SGIF_FLOATRESULT FloatResult; float Value; } SGIF_FLOATVALUE_OUT;</pre>
Description	Structure of the measured value. Variable description: <pre>SGIF_FLOATRESULT FloatResult; // Valid or invalid data float Value; // Measured value</pre>
Remark	--

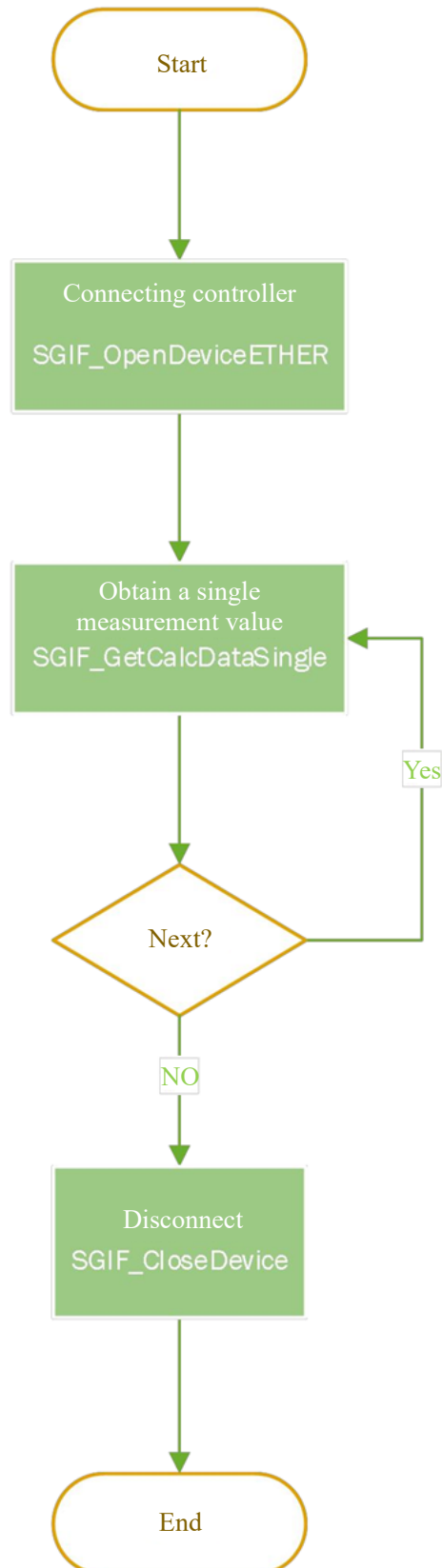
5 Function Return Code Definition

Return code	Definition	Description
RC_OK	Operation completed successfully	
RC_NAK_COMMAND	Command error	
RC_NAK_COMMAND_LENGTH	Command length error	
RC_NAK_TIMEOUT	Timeout error	
RC_NAK_CHECKSUM	Checksum error	
RC_NAK_INVALID_STATE	status error	
RC_NAK_OTHER	Other errors	
RC_NAK_PARAMETER	Parameter error	
RC_NAK_OUT_STAGE	Parameter range error (OUT calculation count limit)	The number of reuses of an OUT value in the OUT calculation exceeds the limit. Check the command sent.
RC_NAK_OUT_HEAD_NUM	Parameter range error (OUT/ sensor head number)	The number of the sensor head or OUT being used exceeds the active sensor head /OUT count.
RC_NAK_PARAM_RANGE_DATA_STORED	The OUT number corresponding to the stored data exceeds the active OUT count.	
RC_NAK_OUT_INVALID_CALC	Parameter range error (OUT specification)	The calculation range setting of specific OUT includes OUT itself, or the target OUT has not been set for the calculation of AVE/P-P/MIN/MAX.
RC_NAK_OUT_VOID	There is no OUT for specifying calculation output.	
RC_NAK_INVALID_CYCLE	Parameter range error (sampling period)	The specified sampling period is faster than the fastest sampling period based on active OUT counting, active sensor head counting, measurement mode and calculation method. Choose a slower sampling period or change other parameters.

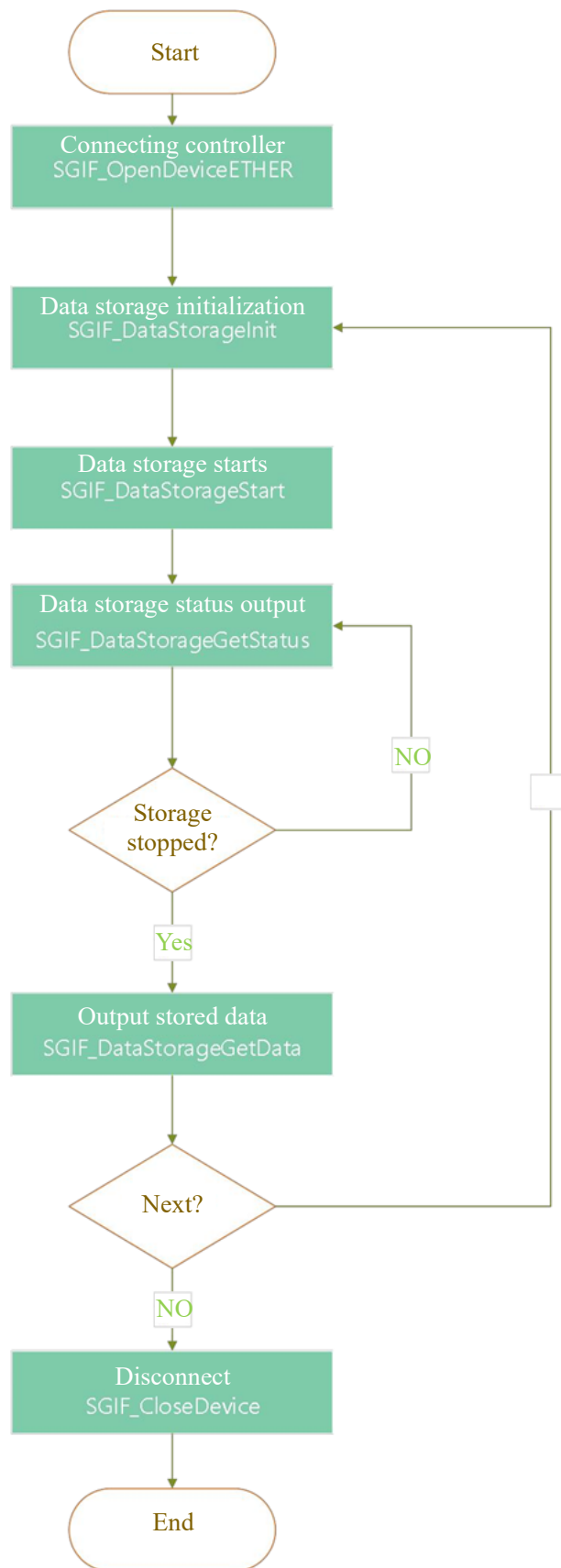
RC_NAK_CTRL_ERROR	Controller error	
RC_NAK_SRAM_ERROR	Setting value error	
RC_ERR_OPEN_DEVICE	Connection to controller failed	
RC_ERR_NO_DEVICE	Not connected to controller	
RC_ERR_SEND	Command sending error	
RC_ERR_RECEIVE	Response received error	
RC_ERR_TIMEOUT	Timeout error	
RC_ERR_NODATA	Data is empty	
RC_ERR_NOMEMORY	there is no memory available	
RC_ERR_DISCONNECT	The connection line between the device and the sensor head is disconnected	
RC_ERR_UNKNOWN	Undefined error	
RC_ERR_DEVID_OVER	Parameter range error (controller)	Exceeding the parameter device range
RC_ERR_NET_NO_CONN	Not connected to the controller	
RC_ERR_IP_EXISTED	IP connected error	The IP is connected, and try to connect it again without releasing the IP.
RC_ERR_SELECT	Socket connection error	
RC_ERR_NULL_PARAMETER	NULL input parameter error	
RC_ERR_DIY_FUNC	Error caused by dir function	
RC_ERR_OVER_PARAMENTER	Parameter exceeds the limit	

6 Function

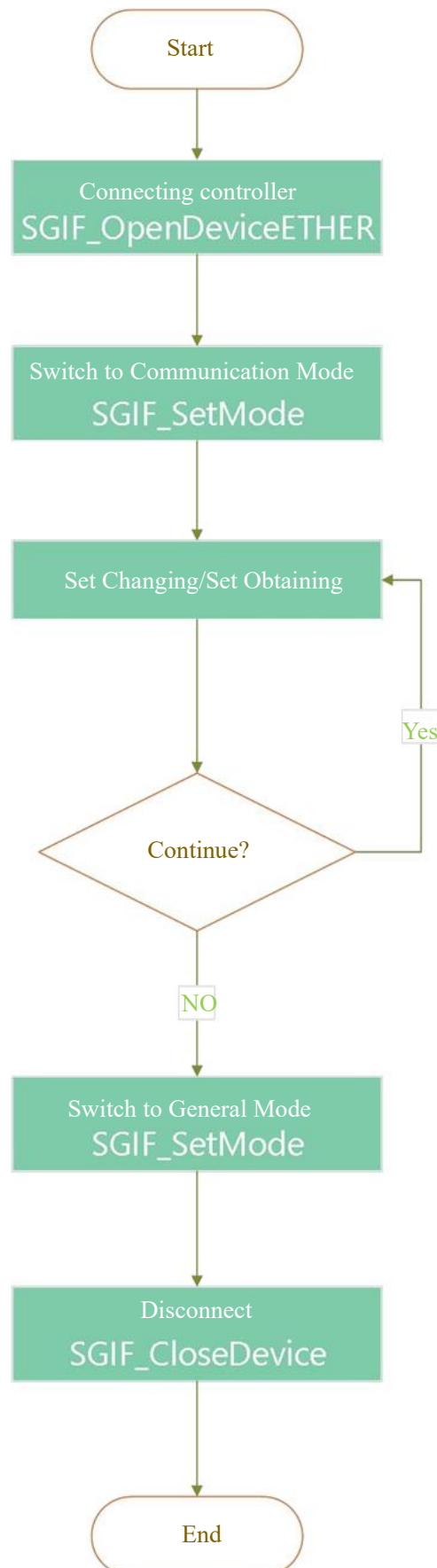
Example flow chart: obtaining a single measurement value



Example flow chart: Data Storage/Data Output



Example flow chart: Setting Change/Setting Acquisition



6.1 Overview of Functions

6.1.1 Establish/disconnect the communication path with the controller

Function Name	Overview
SGIF_OpenDeviceETHER	Establish a connection via Ethernet
SGIF_CloseDevice 错误!未找到引用源。 错误	Disconnect

6.1.2 Mode Change

Function Name	Overview
SGIF_SetMode	Mode Change

6.1.3 Measurement Control

Function Name	Overview
SGIF_GetCalcDataSingle	Measured value output
SGIF_GetCalcDataMulti	Multiple measured value output
SGIF_GetCalcDataALL	All measured values output
SGIF_SetTimingSingle	Timing ON/OFF(single)
SGIF_SetTimingMulti	Timing ON/OFF(multiple)
SGIF_SetTimingSync	Timing ON/OFF(sync.)
SGIF_SetZeroSingle	Auto Zero (single)
SGIF_SetZeroMulti	Auto Zero (multiple)
SGIF_SetZeroSync	Auto Zero (sync.)
SGIF_SetResetSingle	Reset (single)
SGIF_SetResetMulti	Reset (multiple)
SGIF_SetResetSync	Reset (sync.)
SGIF_SetProgramNo	Program switching
SGIF_GetProgramNo	Program confirmation
SGIF_DataStorageStart	Data storage starts
SGIF_DataStorageStop	Data storage stopped
SGIF_DataStorageGetData	Data storage/data output
SGIF_DataStorageInit	Data storage initialization
SGIF_DataStorageGetStatus	Data storage status output
SGIF_GetDeviceRuningTime	Obtain the camera runtime (in seconds)

SGIF_GetEncoderStoreValue	Encoder value stored and data output (only supported by SCI
SGIF_GetEncoderCurrentValue	Read current encoder value (only supported by SCI series
SGIF_GetEncoderReset	Zero-reset encoder value (only supported by SCI series device)

6.1.4 Setting Change

Function Name	Overview
SGIF_SetTolerance	Tolerance setting
SGIF_SetAbleMode	Able Mode setting
SGIF_SetAbleMinMax	Able Control range setting
SGIF_SetMeasureMode	Measurement mode setting
SGIF_SetBasicPoint	Reference point setting
SGIF_SetNumAlarm	Alarm quantity setting
SGIF_SetNumRecovery	Restore quantity setting
SGIF_SetAlarmLevel	Alarm level setting
SGIF_AbleStart	Able Calibration begins
SGIF_AbleStop	Able Calibration completed
SGIF_AbleCancel	Able Calibration stopped
SGIF_SetReflectionMode	Installation mode setting
SGIF_SetMask	Shielding settings
SGIF_SetMedian	Median setting
SGIF_SetLaserCtrlGroup	Laser Ctrl Group setting
SGIF_SetMutualInterferencePreventionGroup	Prevent mutual interference group settings
SGIF_SetCalcMethod	Calculation method setting
SGIF_SetCalcTarget	Measurement target setting
SGIF_SetOutAddSub	OUT (Add, Sub) to be calculated setting
SGIF_SetOutOperation	OUT (Ave, Max, Min, P-P) to be calculated setting
SGIF_SetScaling	Zoom setting
SGIF_SetFilter	Filter setting
SGIF_SetTriggerMode	Trigger mode setting
SGIF_SetOffset	Offset setting
SGIF_SetCalcMode	Detection mode (measurement mode) setting
SGIF_SetDisplayUnit	Measurement unit display setting
SGIF_SetSynchronization	Synchronization setting

SGIF_SetStorageTarget	Storage (OUT number specification) setting
SGIF_SetSamplingCycle	Sampling period setting
SGIF_SetMutualInterferencePrevention	Prevent mutual interference setting
SGIF_SetStrobeTime	Gating time setting
SGIF_SetDataStorage	Data storage setting
SGIF_SetRunningMode	Running mode setting (SCI device is not supported)
SGIF_SetEncoderMode	Encoder input mode setting (SCI device is not supported)
SGIF_SetEncoderNumber	Select an encoder (only supported by SCI devices)
SGIF_SetEncoderParameters	Set encoder parameters (only supported by SCI devices)
SGIF_SetAnalogOutput	Set analog output (only supported by SCI devices)
SGIF_SetIODigitalOutput	Set I/O digital output (only supported by SCI devices)

6.1.5 Setting Acquisition

Function name	Overview
SGIF_SetTolerance	Tolerance setting
SGIF_GetTolerance	Tolerance setting acquisition
SGIF_GetAbleMode	Able setting acquisition
SGIF_GetAbleMinMax	Able control range acquisition
SGIF_GetMeasureMode	Measurement mode acquisition
SGIF_GetBasicPoint	Reference point acquisition
SGIF_GetNumAlarm	Alarm processing times acquisition
SGIF_GetNumRecovery	Alarm recovery times acquisition
SGIF_GetAlarmLevel	Alarm level acquisition
SGIF_GetReflectionMode	Installation mode acquisition
SGIF_GetMask	Shield setting acquisition
SGIF_GetMedian	Median acquisition
SGIF_GetLaserCtrlGroup	Laser Ctrl group acquisition
SGIF_GetMutualInterferencePreventionGroup	Prevent mutual interference group acquisition
SGIF_GetCalcMethod	Calculation method acquisition
SGIF_GetCalcTarget	Measurement target acquisition
SGIF_GetOutAddSub	OUT (Add, Sub) to be calculated acquisition
SGIF_GetOutOperation	OUT (Ave, Max, Min, P-P) to be calculated acquisition
SGIF_GetScaling	Zoom setting acquisition

SGIF_GetFilter	Filter setting acquisition
SGIF_GetTriggerMode	Trigger mode acquisition
SGIF_GetOffset	Offset acquisition
SGIF_GetCalcMode	Detection mode (measurement mode) acquisition
SGIF_GetDisplayUnit	Measurement unit display acquisition
SGIF_GetSynchronization	Synchronization setting acquisition
SGIF_GetStorageTarget	Storage (OUT number setting) acquisition
SGIF_GetSamplingCycle	Sampling period acquisition
SGIF_GetMutualInterferencePrevention	Prevent mutual interference acquisition
SGIF_GetStrobeTime	Gating time acquisition
SGIF_GetDataStorage	Data storage acquisition
SGIF_GetNumOfUsedHeads	Active OUT count acquisition
SGIF_GetNumOfUsedOut	Active sensor head count acquisition
SGIF_GetEncoderParameters	Read encoder parameters (only supported by SCI devices)
SGIF_GetAnalogOutputChannel	Get analog output channel number (only supported by SCI devices)
SGIF_GetIOOutputChannel	Get I/O output channel number (only supported by SCI

6.2 Function Reference

6.2.1 Establish/disconnect communication connections to the Controller

- **IPAddress** Open the device network connection.

Format	RC WINAPI SGIF_OpenDeviceETHER(IN int DeviceID,
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OpenParam Network connection parameters. For each parameter, please refer to " 4 Structure Definition ".
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Establish a connection so that the device can communicate with the controller connected via Ethernet.
Version	V1.0

- **Turn off the device network connection.**

Format	RC WINAPI SGIF_CloseDevice(IN int DeviceID);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Disconnect Ethernet.
Version	V1.0

6.2.2 Mode Change

- **Mode Switching**

Format	RC WINAPI SGIF_SetMode(IN int DeviceID, IN SGIF_MODE Mode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. Mode Mode switching (0: general mode, 1: communication mode). <pre>typedefenum { SGIF_MODE_NORMAL, // General mode, SGIF_MODE_COMMUNICATION, // Communication mode } SGIF_MODE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	The command format used to switch between General Mode and Communication Mode defaults to General Mode. General Mode Measurement period: ① receiving measurement control commands; ② Do not accept commands such as writing/reading set values. Tolerance/function setting period: ① No communication is accepted; ② The measurement will stop. ommunication Mode When receiving the "Q0" command, the mode will be changed to the communication mode, and writing/reading of the set value is allowed; When in communication mode, the measurement will stop.
Version	V1.0

6.2.3 Measurement Control

● Measured Value Output

Format	RC WINAPI SGIF_GetCalcDataSingle(IN int DeviceID, IN int OutNo, OUT SGIF_FLOATVALUE_OUT * CalcData);
Parameters	DeviceID Specifies which device outputs the measured value, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. CalcData For the measured values, please refer to " 4 Structure Definition " for each parameter.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Specifies a single OUT measured value output.
Version	V1.0

● Multiple Measured Value Output

Format	RC WINAPI SGIF_GetCalcDataMulti(IN int DeviceID, IN SGIF_OUTNO OutNo, OUT SGIF_FLOATVALUE_OUT * CalcData);
Parameters	DeviceID Specifies which device outputs the measured value, ranging from 0 to 10. OutNo Output OUT number combination, ranging from 0x0001-0x000F (example: when OUT02 and OUT04 are specified, the combination is 0x000A). CalcData Array of measured values. For each parameter, please refer to " 4 Structure Definition ".
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Multiple OUT measurement values form a measurement value array for output.
Version	V1.0

● All Measured Values Output

Format	RC WINAPI SGIF_GetCalcDataALL(IN int DeviceID, OUT SGIF_FLOATVALUE_OUT * CalcData);
Parameters	DeviceID Specify which communication device to use to obtain measurement values, ranging from 0 to 10. CalcData Array of measured values. For each parameter, please refer to " 4 Structure Definition "
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Output the measured value array of all sensor heads.
Version	V1.0

● Timing ON/OFF(single)

Format	RC WINAPI SGIF_SetTimingSingle(IN int DeviceID, IN int OutNo, IN BOOL OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, ranging 1 to 4. OnOff On or off, 0: off, not 0: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Perform a single sensor head timing operation.
Version	V1.0

● Timing ON/OFF(Multiple)

Format	RC WINAPI SGIF_SetTimingMulti(IN int DeviceID, IN SGIF_OUTNO OutNo, IN BOOL OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number combination, ranging from 0x0001-0x000F (example: when OUT02 and OUT04 are specified, the combination is 0x000A). <pre>typedef enum { SGIF_OUTNO_01 = 0x0001, // OUT01 SGIF_OUTNO_02 = 0x0002, // OUT02 SGIF_OUTNO_03 = 0x0004, // OUT03 SGIF_OUTNO_04 = 0x0008, // OUT04 SGIF_OUTNO_ALL = 0x000F, // All OUTs } SGIF_OUTNO;</pre> OnOff On or off, 0: off, non-zero: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Multiple OUT simultaneously perform timing operations.
Version	V1.0

● Timing ON/OFF(sync.)

Format	RC WINAPI SGIF_SetTimingSync(IN int DeviceID, IN BOOL OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OnOff On or off, 0: off, non-zero: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Perform synchronization timing operation on the OUT set for synchronization.
Version	V1.0

● Auto Zero (single)

Format	RC WINAPI SGIF_SetZeroSingle(IN int DeviceID, IN int OutNo, IN BOOL OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1-4. OnOff On or off, 0: off, non-zero: on. Return value > 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for
Return value	Specify the sensor head to zero.
Description	Reads the level of the specified input port on the controller.
Version	V1.0

● Auto Zero (multiple)

Format	RC WINAPI SGIF_SetZeroMulti(IN int DeviceID, IN SGIF_OUTNO OutNo, IN BOOL OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number combination, ranging from 0x0001-0x000F (example: when OUT02 and OUT04 are specified, the combination is 0x000A). typedef enum { SGIF_OUTNO_01 = 0x0001, // OUT01 SGIF_OUTNO_02 = 0x0002, // OUT02 SGIF_OUTNO_03 = 0x0004, // OUT03 SGIF_OUTNO_04 = 0x0008, // OUT04 SGIF_OUTNO_ALL = 0x000F, // All OUTs } SGIF_OUTNO; OnOff On or off, 0: off, non-zero: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Perform zeroing operations on multiple OUTs.
Version	V1.0

- **Auto Zero (sync.)**

Format	RC WINAPI SGIF_SetZeroSync(IN int DeviceID, IN BOOL OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OnOff On or off, 0: off, non-zero: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Perform a zeroing operation on OUT with synchronization status set.
Version	V1.0

- **Reset (single)**

Format	RC WINAPI SGIF_SetResetSingle(IN int DeviceID , IN int OutNo);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Specify a single OUT to reset.
Version	V1.0

● Reset (multiple)

Format	RC WINAPI SGIF_SetResetMulti(IN int DeviceID, IN SGIF_OUTNO OutNo);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number combination, ranging from 0x0001-0x000F (example: when OUT02 and OUT04 are specified, the combination is 0x000A). <pre>typedef enum { SGIF_OUTNO_01 = 0x0001, SGIF_OUTNO_02 = 0x0002, // OUT01 SGIF_OUTNO_03 = 0x0004, // OUT02 SGIF_OUTNO_04 = 0x0008, // OUT03 SGIF_OUTNO_ALL = 0x000F, // OUT04 } SGIF_OUTNO;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Specify the OUT combination to reset.
Version	V1.0

● Reset (sync.)

Format	RC WINAPI SGIF_SetResetSync(IN int DeviceID);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Perform a reset operation on the synchronization OUT setting
Version	V1.0

● Program Switching

Format	RC WINAPI SGIF_SetProgramNo(IN int DeviceID, IN int ProgramNo);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. ProgramNo Program number (range: 0 to 7)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Program Confirmation

Format	RC WINAPI SGIF_GetProgramNo(IN int DeviceID, OUT int * ProgramNo);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. ProgramNo Program number (range: 0 to 7)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Data Storage Starts

Format	RC WINAPI SGIF_DataStorageStart(IN int DeviceID);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Start storing data
Version	V1.0

● Data Storage Stopped

Format	RC WINAPI SGIF_DataStorageStop(IN int DeviceID);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Stop storing data
Version	V1.0

● Data Storage/Data Output

Format	RC WINAPI SGIF_DataStorageGetData(IN int DeviceID, IN int OutNo, IN int NumOfBuffer, OUT SGIF_FLOATVALUE * OutBuffer, OUT int *NumReceived);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. NumOfBuffer Buffer size (maximum number of data that can be received), maximum number of data that can be received: 1200000. OutBuffer Data array, please refer to " 4 Structure Definition " for various parameters. NumReceived The actual number of data received.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Read the data already stored on the device.
Version	V1.0

● Data Storage Initialization

Format	RC WINAPI SGIF_DataStorageInit(IN int DeviceID);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	The next execution can only be carried out after data storage initialization
Version	V1.0

- Data Storage Status Output

Format	RC WINAPI SGIF_DataStorageGetStatus(IN int DeviceID, OUT BOOL * IsStorage, OUT int * NumStorageData);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. IsStorage Storage status (0: Data storage has stopped, 1: Data is being stored). NumStorageData The quantity and array stored in each output port, according to the specified OUT number, is up to 1200000. The number of output values is the same as the number of OUT used, in order of OUT01, OUT02...
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Get the current data storage status.
Version	V1.0

- Obtain The Camera Runtime (in seconds)

Format	RC WINAPI SGIF_GetDeviceRuningTime(IN int DeviceID, IN int HeadNo, OUT unsigned int * RuningTime);
Parameters	DeviceID Device ID number, ranging 0-63. HeadNo 0: Controller; 1-4: Sensor head number RuningTime Running time
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Get the running time of the camera
Version	V1.0

● **Encoder Value Stored and Data Output (only supported by SCI series device)**

Format	RC WINAPI SGIF_GetEncoderStoreValue(IN int DeviceID, IN int OutNo, IN int NumofBuffer, out int* OutBuffer, out int*NumReceived);
Parameters	DeviceID Device ID number, ranging 0-63. OutNo Encoder number UmofBuffer Buffer size (maximum number of data that can be accepted) OutBuffer Data array UmReceived The number of data actually received
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Acquiring stored data of encoder values
Version	V1.0

● **Read Current Encoder Value (only supported by SCI series device)**

Format	RC WINAPI SGIF_GetEncoderCurrentValue(IN int DeviceID, IN int OutNo, out int * OutBuffer),
Parameters	DeviceID Device ID number, ranging 0-63. OutNo Encoder number (1-3) OutBuffer Data pointer
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Get the current value of the encoder
Version	V1.0

- **Zero-reset Encoder Value (only supported by SCI series device)**

Format	RC WINAPI SGIF_EncoderReset(IN int DeviceID, IN int outNo);
Parameters	DeviceID Device ID number, ranging 0-63. OutNo Encoder number (1-3)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	The encoder value is set to zero.
Version	V1.0

6.2.4 Setting Change

Note: These commands are only accepted when the controller is in "Communication Mode".

- **Tolerance Setting**

Format	RC WINAPI SGIF_SetTolerance(IN int DeviceID, IN int OutNo, IN int UpperLimit, IN int LowerLimit, IN int Hystersys);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number UpperLimit Upper limit, input as a valid number (integer after decimal point removed, valid digits: 6). LowerLimit Lower limit, input as a valid digit (integer after decimal point is removed, and the valid digit is 6). Hysteresis Hysteresis, input as a valid digit (integer after decimal point is removed, and the valid digit is 6).
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Condition to be met: UpperLimit is greater than LowerLimit
Version	V1.0

● Able Mode Setting

Format	RC WINAPI SGIF_SetAbleMode(IN int DeviceID, IN int HeadNo, IN SGIF_ABLEMODE AbleMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number AbleMode Mode (0: AUTO, 1: MANNUAL). typedef enum { SGIF_ABLEMODE_AUTO, // AUTO SGIF_ABLEMODE_MANUAL, // manual } SGIF_ABLEMODE;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Used to set the mode "AUTO" or "MANUAL".
Version	V1.0

● Able Control Range Setting

Format	RC WINAPI SGIF_SetAbleMinMax(IN int DeviceID, IN int HeadNo, IN int Min, IN int Max);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. Min Lower limit, range: 1-99. Max Upper limit, range: 1-99.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Available in "MANUAL" mode.
Version	V1.0

● Measurement Mode Setting

Format	RC WINAPI SGIF_SetMeasureMode(IN int DeviceID, IN int HeadNo, IN SGIF_MEASUREMODE MeasureMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. MeasureMode Measurement mode (0: Normal; 1: Semi-transparent; 2: Transparent; 3: Transparent 2; 4: Reflective resin) <pre>typedef enum { SGIF_MEASUREMODE_NORMAL, // Normal SGIF_MEASUREMODE_HALF_T, // Semi-transparent SGIF_MEASUREMODE_TRAN_1, // Transparent SGIF_MEASUREMODE_TRAN_2, // Transparent 2 SGIF_MEASUREMODE_MRS, // Reflective resin } SGIF_MEASUREMODE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Set the measurement method for the sensor head.
Version	V1.0

● Reference Point Setting

Format	RC WINAPI SGIF_SetBasicPoint(IN int DeviceID, IN int HeadNo, IN SGIF_BASICPOINT BasicPoint);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. BasicPoint Reference point (0: NEAR; 1: FAR) typedef enum { SGIF_BASICPOINT_NEAR, // NEAR SGIF_BASICPOINT_FAR, // FAR } SGIF_BASICPOINT;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Alarm Quantity Setting

Format	RC WINAPI SGIF_SetNumAlarm(IN int DeviceID, IN int HeadNo, IN int AlarmNum);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. AlarmNum Number of alarms, unsigned values (0 to 9999).
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Restore Quantity Setting

Format	RC WINAPI SGIF_SetNumRecovery(IN int DeviceID , IN int HeadNo, IN int RecoveryNum);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. RecoveryNum Recovery times, unsigned values (0 to 9999).
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Alarm Level Setting

Format	RC WINAPI SGIF_SetAlarmLevel(IN int DeviceID, IN int HeadNo, IN int AlarmLevel);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. AlarmLevel Alarm level, range: 0 to 9.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- Able Calibration Begins

Format	RC WINAPI SGIF_AbleStart(IN int DeviceID, IN int HeadNo);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- Able Calibration Completed

Format	RC WINAPI SGIF_AbleStop(IN int DeviceID);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- Able Calibration Stopped

Format	RC WINAPI SGIF_AbleCancel(IN int DeviceID);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Installation Mode Setting

Format	RC WINAPI SGIF_SetReflectionMode(IN int DeviceID, IN int HeadNo, IN SGIF_REFLECTIONMODE ReflectionMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number ReflectionMode Reflection mode (0: diffuse reflection, 1: specular reflection). <pre>typedef enum { SGIF_REFLECTIONMODE_DIFFUSION, // diffuse reflection SGIF_REFLECTIONMODE_MIRROR, // specular reflection } SGIF_REFLECTIONMODE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Shielding Settings

Format	RC WINAPI SGIF_SetMask(IN int DeviceID, IN int HeadNo, IN BOOL OnOff, IN int Pos1, IN int Pos2);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. OnOff Switch, 0: off, non-zero: on. Pos1 Point 1 Pos2 Point 2
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Median Setting

Format	RC WINAPI SGIF_SetMedian(IN int DeviceID, IN int HeadNo, IN SGIF_MEDIAN Median);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. Median Median (0: OFF, 1:7, 2:15, 3:31). <pre>typedef enum { SGIF_MEDIAN_OFF, // OFF SGIF_MEDIAN_7, // 7 point SGIF_MEDIAN_15, // 15 point SGIF_MEDIAN_31, // 31 point } SGIF_MEDIAN;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Laser Ctrl Group Setting

Format	RC WINAPI SGIF_SetLaserCtrlGroup(IN int DeviceID, IN int HeadNo, IN SGIF_LASER_CTRL_GROUP LaserCtrlGroup);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. LaserCtrlGroup Laser Ctrl group (0: laser control 1, 1: laser control 2) <pre>typedef enum { SGIF_LASER_CTRL_GROUP_1, // LASER CTRL 1 SGIF_LASER_CTRL_GROUP_2, // LASER CTRL 2 } SGIF_LASER_CTRL_GROUP;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Prevent Mutual Interference Group Settings

Format	RC WINAPI SGIF_SetMutualInterferencePreventionGroup(IN int DeviceID, IN int HeadNo, IN SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP Group);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range: 1-4. Group Group settings (0: A, 1: B, 2: C). typedef enum { SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP_A, // Group A SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP_B, // Group B SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP_C, // Group C } SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	There are groups A, B, and C, and by specifying the sensor head, it can be set to the corresponding group.
Version	V1.0

● Calculation Method Setting

Format	RC WINAPI SGIF_SetCalcMethod(IN int DeviceID, IN int OutNo, IN SGIF_CALCMETHOD CalcMethod, IN int HeadOutNo);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. CalcMethod Calculation method (0:head, 1:OUT, 2:ADD, 3:SUB, 4:AVE, 5: MAX, 6:MIN, 7:P-P) <pre>typedef enum { SGIF_CALCMETHOD_HEADA, // head A SGIF_CALCMETHOD_HEADB, // head B SGIF_CALCMETHOD_HEAD_HEADA_PLUS_HEADB, // head A+head B SGIF_CALCMETHOD_HEAD_HEADA_MINUS_HEADB, // head A-head B SGIF_CALCMETHOD_HEAD_HEADA_TRANSPARENT, // head A transparent object SGIF_CALCMETHOD_HEAD_HEADB_TRANSPARENT, // head B transparent object SGIF_CALCMETHOD_HEAD = 0, // head SGIF_CALCMETHOD_OUT, // OUT SGIF_CALCMETHOD_ADD, // ADD SGIF_CALCMETHOD_SUB, // SUB SGIF_CALCMETHOD_AVE, // AVE SGIF_CALCMETHOD_MAX, // MAX SGIF_CALCMETHOD_MIN, // MIN SGIF_CALCMETHOD_PP, // P-P } SGIF_CALCMETHOD;</pre> HeadOutNo Sensor head or OUT number, range 1-4.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	When CalcMethod is equal to 0, HeadOutNo: 1-4 represents the sensor head number (H01 to H04); When CalcMethod equals 1, HeadOutNo: 1-4 represents OUT numbers (O01 to O04); When CalcMethod is equal to 2-7, the parameter HeadOutNo can be set to any value.
Version	V1.0

● Measurement Target Setting

Format	RC WINAPI SGIF_GetCalcTarget(IN int DeviceID, IN int OutNo, OUT SGIF_CALCTARGET * CalcTarget);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. CalcTarget Calculation tags (0: 1, 1: 2, 2: 3, 3: 4, 4: 1-2, 5: 1-3, 6: 1-4, 7: 2-3, 8: 2-4, 9: 3-4). <pre>typedef enum { SGIF_CALCTARGET_PEAK_1, // peak 1 SGIF_CALCTARGET_PEAK_2, // peak 2 SGIF_CALCTARGET_PEAK_3, // peak 3 SGIF_CALCTARGET_PEAK_4, // peak 4 SGIF_CALCTARGET_PEAK_1_2, // peak 1-peak 2 SGIF_CALCTARGET_PEAK_1_3, // peak 1-peak 3 SGIF_CALCTARGET_PEAK_1_4, // peak 1-peak 4 SGIF_CALCTARGET_PEAK_2_3, // peak 2-peak 3 SGIF_CALCTARGET_PEAK_2_4, // peak 2-peak 4 SGIF_CALCTARGET_PEAK_3_4, // peak 3-peak 4 } SGIF_CALCTARGET;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	This function needs to be used when the detection mode is "Transparent" or "Transparent 2".
Version	V1.0

- OUT (Add, Sub) To Be Calculated Setting

Format	RC WINAPI SGIF_GetOutAddSub(IN int DeviceID, IN int OutNo, OUT int * AddSub1, OUT int * AddSub2);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. AddSub1 OUT to be calculated (OUT itself cannot be set) AddSub2 OUT to be calculated (OUT itself cannot be set)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- OUT (Ave, Max, Min, P-P) To Be Calculated Setting

Format	RC WINAPI SGIF_GetOutOperation(IN int DeviceID, IN int OutNo, OUT SGIF_OUTNO * TargetOut);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. TargetOut Multiple OUT combination, ranging from 0x0001-0x000F (example: OUT02: 0x0002, OUT04: 0x0008, when specifying OUT02 and OUT04, the combination is 0x000A). typedef enum { SGIF_OUTNO_01 = 0x0001, // OUT01 SGIF_OUTNO_02 = 0x0002, // OUT02 SGIF_OUTNO_03 = 0x0004, // OUT03 SGIF_OUTNO_04 = 0x0008, // OUT04 SGIF_OUTNO_ALL = 0x000F, // All OUTs } SGIF_OUTNO;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Zoom Setting

Format	RC WINAPI SGIF_SetScaling(IN int DeviceID, IN int OutNo, IN int InputValue1, IN int OutputValue1, IN int InputValue2, IN int OutputValue2);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number InputValue1 The measured value (input value) at point 1, input as a valid number (integer after decimal point removed, valid digits: 6). OutputValue1 The display value (output value) at point 1, output as a valid number (integer after decimal point removed, valid digits: 6). InputValue2 The measured value (input value) at point 1, input as a valid number (integer after decimal point removed, valid digits: 6). OutputValue2 The display value (output value) at point 2, output as a valid number (integer after decimal point removed, valid digits: 6).
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Required conditions: 1. $\text{InputValue2} - \text{InputValue1} \neq 0;$ 2. $\left \frac{\text{OutputValue2} - \text{OutputValue1}}{\text{InputValue2} - \text{InputValue1}} \right \leq 2$
Version	V1.0

● Filter Setting

Format	RC WINAPI SGIF_SetFilter(IN int DeviceID, IN int OutNo, IN SGIF_FILTERMODE FilterMode, IN SGIF_FILTERPARA FilterPara);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. FilterMode Filter mode (0: moving average). typedef enum { SGIF_FILTERMODE_MOVING_AVERAGE, // Moving average } SGIF_FILTERMODE; FilterPara Filter parameters (0:1, 1:4, 2:16, 3:64, 4:256, 5:1024, 6:4096, 7:16384, 65536, 9:262144). typedef enum { SGIF_FILTERPARA_AVE_1 = 0, // 1 time SGIF_FILTERPARA_AVE_4, // 4 times SGIF_FILTERPARA_AVE_16, // 16 times SGIF_FILTERPARA_AVE_64, // 64 times SGIF_FILTERPARA_AVE_256, // 256 times SGIF_FILTERPARA_AVE_1024, // 1024 times SGIF_FILTERPARA_AVE_4096, // 4096 times SGIF_FILTERPARA_AVE_16384, // 16384 times SGIF_FILTERPARA_AVE_65536, // 65536 times SGIF_FILTERPARA_AVE_262144, // 262144 times } SGIF_FILTERPARA;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Trigger Mode Setting

Format	RC WINAPI SGIF_SetTriggerMode(IN int DeviceID, IN int OutNo, IN SGIF_TRIGGERMODE TriggerMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. TriggerMode Trigger mode (0: trigger 1, 1: trigger 2). typedef enum { SGIF_TRIGGERMODE_EXT1, // external trigger 1 SGIF_TRIGGERMODE_EXT2, // external trigger 2 } SGIF_TRIGGERMODE;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Offset Setting

Format	RC WINAPI SGIF_SetOffset(IN int DeviceID, IN int OutNo, IN int Offset);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. Offset Offset (vailid digits: 6), range: -999.999 to 999.999.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Detection Mode (Measurement Mode) Setting

Format	RC WINAPI SGIF_SetCalcMode(IN int DeviceID, IN int OutNo, IN SGIF_CALCMODE CalcMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. CalcMode Detection mode number (0: Normal, 1: Peak hold, 2: Valley hold, 3: P-P hold, 4: Sample hold). <pre>typedef enum { SGIF_CALCMODE_NORMAL, // Normal SGIF_CALCMODE_PEAKHOLD, // Peak hold SGIF_CALCMODE_BOTTOMHOLD, // Valley hold SGIF_CALCMODE_PEAKTOPEAKHOLD, // P-P hold SGIF_CALCMODE_SAMPLEHOLD, // Sample hold } SGIF_CALCMODE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Measurement Unit Display Setting

Format	RC WINAPI SGIF_SetDisplayUnit(IN int DeviceID, IN int OutNo, IN SGIF_DISPLAYUNIT DisplayUnit);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. DisplayUnit Display units (0: 0.01 mm, 1: 0.001 mm, 2: 0.0001 mm, 3: 0.00001 mm, 4: 0.1 μm, 5: 0.01 μm, 6: 0.001 μm). <pre>typedef enum { SGIF_DISPLAYUNIT_0000_01MM = 0, //0.01mm SGIF_DISPLAYUNIT_000_001MM, //0.001mm SGIF_DISPLAYUNIT_00_0001MM, //0.0001mm SGIF_DISPLAYUNIT_0_00001MM, //0.00001mm SGIF_DISPLAYUNIT_00000_1UM, //0.1μm SGIF_DISPLAYUNIT_0000_01UM, //0.01μm SGIF_DISPLAYUNIT_000_001UM, //0.0001μm } SGIF_DISPLAYUNIT;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Synchronization Setting

Format	RC WINAPI SGIF_SetSynchronization(IN int DeviceID, IN int OutNo, IN BOOL OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. OnOff Switch: 0: off, non-zero: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Specify OUT synchronization
Version	V1.0

● **Storage (OUT Number Specification) Setting**

Format	RC WINAPI SGIF_SetStorageTarget(IN int DeviceID, IN int OutNo, IN BOOL OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. OnOff Switch: 0: off, non-zero: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Sampling Period Setting

Format	RC WINAPI SGIF_SetSamplingCycle(IN int DeviceID, IN SGIF_SAMPLINGCYCLE SamplingCycle);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. SamplingCycle Sampling period (SG5000 series: 0-8 sequentially represent 590kHz, 400kHz, 200kHz, 88kHz, 50kHz, 20kHz, 10kHz, 5kHz, 1kHz; SG3000 series: 10-16 sequentially represent 88kHz, 50kHz, 20kHz, 10kHz, 5kHz, 1kHz) <pre>typedef enum { //SG5000 series SGIF_5000_SAMPLINGCYCLE_590KHZ, // 590kHz SGIF_5000_SAMPLINGCYCLE_400KHZ, // 400kHz SGIF_5000_SAMPLINGCYCLE_200KHZ, // 200kHz SGIF_5000_SAMPLINGCYCLE_88KHZ, // 88kHz SGIF_5000_SAMPLINGCYCLE_50KHZ, // 50kHz SGIF_5000_SAMPLINGCYCLE_20KHZ, // 20kHz SGIF_5000_SAMPLINGCYCLE_10KHZ, // 10kHz SGIF_5000_SAMPLINGCYCLE_5KHZ, // 5kHz SGIF_5000_SAMPLINGCYCLE_1KHZ, // 1kHz+ //SG3000 series SGIF_3000_SAMPLINGCYCLE_88KHZ = 10, // 88kHz 88kHz SGIF_3000_SAMPLINGCYCLE_50KHZ, // 50kHz SGIF_3000_SAMPLINGCYCLE_20KHZ, // 20kHz SGIF_3000_SAMPLINGCYCLE_10KHZ, // 10kHz SGIF_3000_SAMPLINGCYCLE_5KHZ, // 5kHz SGIF_3000_SAMPLINGCYCLE_1KHZ, // 1kHz } } SGIF_SAMPLINGCYCLE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- Prevent Mutual Interference Group Settings

Format	RC WINAPI SGIF_SetMutualInterferencePrevention(IN int DeviceID, IN SGIF_MUTUAL_INTERFERENCE_PREVENTION OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OnOff Group setting (Group A: 0, Group B: 1, Group C: 2) typedef enum { SGIF_MUTUAL_INTERFERENCE_PREVENTION_OFF, // OFF SGIF_MUTUAL_INTERFERENCE_PREVENTION_AB_ON, // AB-ON SGIF_MUTUAL_INTERFERENCE_PREVENTION_ABC_ON, // ABC-ON } SGIF_MUTUAL_INTERFERENCE_PREVENTION;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Specify OUT settings to prevent mutual interference between groups.
Version	V1.0

- Gating Time Setting

Format	RC WINAPI SGIF_SetStrobeTime(IN int DeviceID, IN SGIF_STOROBETIME StrobeTime);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. StrobeTime Gating time (0: 2 ms, 1: 5 ms, 2: 10ms, 3: 20 ms) typedef enum { SGIF_STOROBETIME_2MS, // 2ms SGIF_STOROBETIME_5MS, // 5ms SGIF_STOROBETIME_10MS, // 10ms SGIF_STOROBETIME_20MS, // 20ms } SGIF_STOROBETIME;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Data Storage Setting

Format	RC WINAPI SGIF_SetDataStorage(IN int DeviceID, IN int NumStorage, IN SGIF_STORAGECYCLE StorageCycle);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. NumStorage The number of the data to be stored, ranging from 0 to 1200000. StorageCycle Storage period (0: 1x, 1: 2x, 2: 5x, 3: 10x, 4: 20x, 5: 50x, 6: 100x, 7: 200x, 8: 500x, 9: 1000x, 10: synchronous input). <pre>typedef enum { SGIF_STORAGECYCLE_1, // sampling rate x 1 SGIF_STORAGECYCLE_2, // sampling rate x 2 SGIF_STORAGECYCLE_5, // sampling rate x 5 SGIF_STORAGECYCLE_10, // sampling rate x 10 SGIF_STORAGECYCLE_20, // sampling rate x 20 SGIF_STORAGECYCLE_50, // sampling rate x 50 SGIF_STORAGECYCLE_100, // sampling rate x 100 SGIF_STORAGECYCLE_200, // sampling rate x 200 SGIF_STORAGECYCLE_500, // sampling rate x 500 SGIF_STORAGECYCLE_1000, // sampling rate x 1000 SGIF_STORAGECYCLE_TIMING, // Timing sync } SGIF_STORAGECYCLE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Running Mode Setting (SCI device is not supported)**

Format	RC WINAPI SGIF_SetRunningMode(IN int DeviceID, IN SGIF_EmRunningMode Runningmode, IN SGIF_EmDirection CatchDirection);
Parameters	DeviceID Device ID number, ranging from 0 to 63. Runningmode Operation mode 0:OFF / 1: timing / 2: trigger. CatchDirection Detection direction 0: positive&negative timing / 1 : positive timing / 2: negative timing
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Encoder Input Mode Setting (SCI device is not supported)**

Format	RC WINAPI SGIF_SetEncoderMode(IN int DeviceID, IN SGIF_EmEncoderInput Inputmode, IN SGIF_EmInputTime InputTime, IN int TrigInterval);
Parameters	DeviceID Device ID number, ranging from 0 to 63. Inputmode Operation mode 0: 1 phase 1 increment / 1: 2 phase 1 increment / 2: 2 phase 2 increment / 3: 2 phase 4 increment. MinInputTime Minimum input time TrigInterval Trigger interval
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Select An Encoder (only supported by SCI devices)**

Format	RC WINAPI SGIF_SetEncoderNumber(IN int DeviceID, IN int outNo);
Parameters	DeviceID Device ID number, ranging from 0 to 63. OutNo Encoder number (1-3)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Set Encoder Parameters (only supported by SCI devices)**

Format	RC WINAPI SGIF_SetEncoderParameters(IN int DeviceID, IN int OutNo, IN int InputMode, IN int InputTime, IN int interval);
Parameters	DeviceID Device ID number, ranging from 0 to 63. OutNo Encoder number (1-3) InputMode Input Mode (0-3) InputTime Minimum input time (100, 200, 500, 1000, 2000, 5000, 10000, 20000) (ns) Interval Interval (1- 10000) OperatingMode Operation mode (0-2) DetectionDirection Detection direction (0-2)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Set Analog Output (only supported by SCI devices)**

Format	RC WINAPI SGIF_SetAnalogoutput(IN int DeviceID, IN int OutNo, IN int Outputchannel),
Parameters	DeviceID Device ID number, ranging from 0 to 63. OutNo Analog number (1-3) OutputChannel Output channel (0-3)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Set I/O Digital Output (only supported by SCI devices)**

Format	RC WINAPI SGIF_SetIoDigitalOutput(IN int DeviceID, IN int OutNo, IN int level);
Parameters	DeviceID Device ID number, ranging from 0 to 63. OutNo Digital output number (1-4) Level Number of output bits (0:OFF, 1:HI, 2:GO, 3:LO, 4:STROBE)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

6.2.5 Setting Acquisition

Note: These commands are only accepted when the controller is in "Communication Mode".

- Tolerance Setting Acquisition

Format	RC WINAPI SGIF_GetTolerance(IN int DeviceID, IN int OutNo, OUT int * UpperLimit, OUT int * LowerLimit, OUT int * Hysteresis);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. UpperLimit Upper limit, output as valid numbers (integer after decimal point is removed) LowerLimit Lower limit, output as valid numbers (integer after decimal point is removed) Hysteresis Hysteresis, output as a valid digit (integer after decimal point is removed).
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Condition to be met: UpperLimit is greater than LowerLimit
Version	V1.0

- Able Setting Acquisition

Format	RC WINAPI SGIF_GetAbleMode(IN int DeviceID, IN int HeadNo, OUT SGIF_ABLEMODE * AbleMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. AbleMode Able mode (0: AUTO, 1: MANUAL). typedef enum { SGIF_ABLEMODE_AUTO, // AUTO SGIF_ABLEMODE_MANUAL, // MANUAL } SGIF_ABLEMODE;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Able Control Range Acquisition

Format	RC WINAPI SGIF_GetAbleMinMax(IN int DeviceID, IN int HeadNo, OUT int * Min, OUT int * Max);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. Min Minimum value, range: 1 to 99. Max Maximum value, range: 1 to 99.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Measurement Mode Acquisition

Format	RC WINAPI SGIF_GetMeasureMode(IN int DeviceID, IN int HeadNo, OUT SGIF_MEASUREMODE * MeasureMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. MeasureMode Measurement mode (0: normal; 1: semi-transparent object; 2: transparent object; 3: transparent object 2; 2: reflective resin) typedef enum { SGIF_MEASUREMODE_NORMAL, // Normal SGIF_MEASUREMODE_HALF_T, // Semi-transparent SGIF_MEASUREMODE_TRAN_1, // Transparent SGIF_MEASUREMODE_TRAN_2, // Transparent 2 SGIF_MEASUREMODE_MRS, // Reflective resin SGIF_MEASUREMODE_OPAQUE = SGIF_MEASUREMODE_MRS, // Semi opaque object } SGIF_MEASUREMODE;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Reference Point Acquisition

Format	RC WINAPI SGIF_GetBasicPoint(IN int DeviceID, IN int HeadNo, OUT SGIF_BASICPOINT * BasicPoint);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. BasicPoint Reference point (0: NEAR, 1: FAR). typedef enum { SGIF_BASICPOINT_NEAR, // NEAR SGIF_BASICPOINT_FAR, // FAR } SGIF_BASICPOINT;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Alarm Processing Times Acquisition

Format	RC WINAPI SGIF_GetNumAlarm(IN int DeviceID, IN int HeadNo, OUT int * AlarmNum);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. AlarmNum Number of alarms.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Alarm Recovery Times Acquisition

Format	RC WINAPI SGIF_GetNumRecovery(IN int DeviceID, IN int HeadNo, OUT int * RecoveryNum);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. RecoveryNum Recovery times.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Alarm Level Acquisition

Format	RC WINAPI SGIF_GetAlarmLevel(IN int DeviceID, IN int HeadNo, OUT int * AlarmLevel);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. AlarmLevel Alarm level, range 0-9.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Installation Mode Acquisition

Format	RC WINAPI SGIF_GetReflectionMode(IN int DeviceID, IN int HeadNo, OUT SGIF_REFLECTIONMODE * ReflectionMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. ReflectionMode Reflection mode (0: diffuse reflection, 1: specular reflection). typedef enum { SGIF_REFLECTIONMODE_DIFFUSION, // Diffuse reflection SGIF_REFLECTIONMODE_MIRROR, // Specular reflection } SGIF_REFLECTIONMODE;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Acquire installation mode
Version	V1.0

● Shield Setting Acquisition

Format	RC WINAPI SGIF_GetMask(IN int DeviceID, IN int HeadNo, OUT BOOL * OnOff, OUT int * Pos1, OUT int * Pos2);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. OnOff Switch, 0: off, non-zero, on. Pos1 Point 1 Pos2 Point 2
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Median Acquisition

Format	RC WINAPI SGIF_GetMedian(IN int DeviceID, IN int HeadNo, OUT SGIF_MEDIAN * Median);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. OnOff Switch, 0: off, non-zero, on. Median Median (0: OFF, 1:7, 2:15, 3:31). typedef enum { SGIF_MEDIAN_OFF, // OFF SGIF_MEDIAN_7, // 7 point SGIF_MEDIAN_15, // 15 point SGIF_MEDIAN_31, // 31 point } SGIF_MEDIAN;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	
Version	V1.0

● Laser Ctrl Group Acquisition

Format	RC WINAPI SGIF_GetLaserCtrlGroup(IN int DeviceID, IN int HeadNo, OUT SGIF_LASER_CTRL_GROUP * LaserCtrlGroup);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. LaserCtrlGroup Laser Ctrl Group (0: LASER CTRL1, 1: LASER CTRL2). typedef enum { SGIF_LASER_CTRL_GROUP_1, // LASER CTRL 1 SGIF_LASER_CTRL_GROUP_2, // LASER CTRL 2 } SGIF_LASER_CTRL_GROUP;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Prevent Mutual Interference Group Acquisition

Format	RC WINAPI SGIF_GetMutualInterferencePreventionGroup(IN int DeviceID, IN int HeadNo, OUT SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP * Group);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. HeadNo Sensor head number, range 1-4. Group Group (0:A , 1:B , 2:C) <pre>typedef enum { SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP_A, // Group A SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP_B, // Group B SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP_C, // Group C } SGIF_MUTUAL_INTERFERENCE_PREVENTION_GROUP;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Calculation Method Acquisition

Format	RC WINAPI SGIF_GetCalcMethod(IN int DeviceID, IN int OutNo, OUT SGIF_CALCMETHOD * CalcMethod, OUT int * HeadOutNo);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. CalcMethod Calculation method (0: head, 1: OUT, 2: ADD, 3: SUB, 4: AVE, 5: MAX, 6: MIN, 7: P-P). <pre>typedef enum { SGIF_CALCMETHOD_HEADA, // head A SGIF_CALCMETHOD_HEADB, // head B SGIF_CALCMETHOD_HEAD_HEADA_PLUS_HEADB, // head A+head B SGIF_CALCMETHOD_HEAD_HEADA_MINUS_HEADB, // head A-head B SGIF_CALCMETHOD_HEAD_HEADA_TRANSPARENT, // head A transparent object SGIF_CALCMETHOD_HEAD_HEADB_TRANSPARENT, // head B transparent object SGIF_CALCMETHOD_HEAD = 0, // head SGIF_CALCMETHOD_OUT, // OUT SGIF_CALCMETHOD_ADD, // ADD SGIF_CALCMETHOD_SUB, // SUB SGIF_CALCMETHOD_AVE, // AVE SGIF_CALCMETHOD_MAX, // MAX SGIF_CALCMETHOD_MIN, // MIN SGIF_CALCMETHOD_PP, // P-P } SGIF_CALCMETHOD;</pre> HeadOutNo Sensor head or OUT number, range 1-4.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	When CalcMethod is equal to 0, HeadOutNo: 1-4 represents the sensor head number (H01 to H04); When CalcMethod equals 1, HeadOutNo: 1-4 represents OUT numbers (O01 to O04); When CalcMethod is equal to 2-7, the parameter HeadOutNo can be set to any value.
Version	V1.0

- Measurement Target Acquisition

Format	RC WINAPI SGIF_GetCalcTarget(IN int DeviceID, IN int OutNo, OUT SGIF_CALCTARGET * CalcTarget);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. CalcTarget Calculation tags (0: 1, 1: 2, 2: 3, 3: 4, 4: 1-2, 5: 1-3, 6: 1-4, 7: 2-3, 8: 2-4, 9: 3-4). <pre>typedef enum { SGIF_CALCTARGET_PEAK_1, // peak 1 SGIF_CALCTARGET_PEAK_2, // peak 2 SGIF_CALCTARGET_PEAK_3, // peak 3 SGIF_CALCTARGET_PEAK_4, // peak 4 SGIF_CALCTARGET_PEAK_1_2, // peak 1-peak 2 SGIF_CALCTARGET_PEAK_1_3, // peak 1-peak 3 SGIF_CALCTARGET_PEAK_1_4, // peak 1-peak 4 SGIF_CALCTARGET_PEAK_2_3, // peak 2-peak 3 SGIF_CALCTARGET_PEAK_2_4, // peak 2-peak 4 SGIF_CALCTARGET_PEAK_3_4, // peak 3-peak 4 } SGIF_CALCTARGET;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	This function needs to be used when the detection mode is "Transparent" or "Transparent 2".
Version	V1.0

- **OUT (Add, Sub) To Be Calculated Acquisition**

Format	RC WINAPI SGIF_GetOutAddSub(IN int DeviceID, IN int OutNo, OUT int * AddSub1, OUT int * AddSub2);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. AddSub1 OUT number 1 to be calculated. AddSub2 OUT number 2 to be calculated.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **OUT (Ave, Max, Min, P-P) To Be Calculated Acquisition**

Format	RC WINAPI SGIF_GetOutOperation(IN int DeviceID, IN int OutNo, OUT SGIF_OUTNO * TargetOut);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. TargetOut Multiple OUTs or HEADs combination, ranging from 0x0001-0x000F (example: OUT02: 0x0002, OUT04: 0x0008, when specifying OUT02 and OUT04, the combination is 0x000A). <pre>typedef enum { SGIF_OUTNO_01 = 0x0001, // OUT01 SGIF_OUTNO_02 = 0x0002, // OUT02 SGIF_OUTNO_03 = 0x0004, // OUT03 SGIF_OUTNO_04 = 0x0008, // OUT04 SGIF_OUTNO_ALL = 0x000F, // All OUTs } SGIF_OUTNO;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Zoom Setting Acquisition

Format	RC WINAPI SGIF_GetScaling(IN int DeviceID, IN int OutNo, OUT int * InputValue1, OUT int * OutputValue1, OUT int * InputValue2, OUT int * OutputValue2);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. InputValue1 The measured value (input value) at point 1, input as a valid number (integer after decimal point removed, valid digits: 6). OutputValue1 The display value (output value) at point 1, output as a valid number (integer after decimal point removed, valid digits: 6). InputValue2 The measured value (input value) at point 2, input as a valid number (integer after decimal point removed, valid digits: 6). OutputValue2 The display value (output value) at point 2, output as a valid number (integer after decimal point removed, valid digits: 6).
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Filter Setting Acquisition

Format	RC WINAPI SGIF_GetFilter(IN int DeviceID, IN int OutNo, OUT SGIF_FILTERMODE * FilterMode, OUT SGIF_FILTERPARA * FilterPara);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. FilterMode Filter mode (0: Moving average). typedef enum { SGIF_FILTERMODE_MOVING_AVERAGE, // moving average } SGIF_FILTERMODE; FilterPara Filter parameters (0:1, 1:4, 2:16, 3:64, 4:256, 5:1024, 6:4096, 7:16384, 65536, 9:262144). typedef enum { SGIF_FILTERPARA_AVE_1 = 0, // 1 time SGIF_FILTERPARA_AVE_4, // 4 times SGIF_FILTERPARA_AVE_16, // 16 times SGIF_FILTERPARA_AVE_64, // 64 times SGIF_FILTERPARA_AVE_256, // 256 times SGIF_FILTERPARA_AVE_1024, // 1024 times SGIF_FILTERPARA_AVE_4096, // 4096 times SGIF_FILTERPARA_AVE_16384, // 16384 times SGIF_FILTERPARA_AVE_65536, // 65536 times SGIF_FILTERPARA_AVE_262144, // 262144 times }SGIF_FILTERPARA;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Trigger Mode Acquisition**

Format	RC WINAPI SGIF_GetTriggerMode(IN int DeviceID, IN int OutNo, OUT SGIF_TRIGGERMODE * TriggerMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. TriggerMode Trigger mode (0: trigger 1, 1: trigger 2). typedef enum { SGIF_TRIGGERMODE_EXT1, // external trigger 1 SGIF_TRIGGERMODE_EXT2, // external trigger 2 } SGIF_TRIGGERMODE;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Offset Acquisition**

Format	RC WINAPI SGIF_GetOffset(IN int DeviceID, IN int OutNo, OUT int * Offset);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. Offset Amount of offset
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Detection Mode (Measurement Mode) Acquisition

Format	RC WINAPI SGIF_GetCalcMode(IN int DeviceID, IN int OutNo, OUT SGIF_CALCMODE * CalcMode);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. CalcMode Detection mode number (0: Normal, 1: Peak hold, 2: Valley hold, 3: P-P hold, 4: Sample hold). <pre>typedef enum { SGIF_CALCMODE_NORMAL, // Normal SGIF_CALCMODE_PEAKHOLD, // Peak hold SGIF_CALCMODE_BOTTOMHOLD, // Valley hold SGIF_CALCMODE_PEAKTOPEAKHOLD, // P-P hold SGIF_CALCMODE_SAMPLEHOLD, // Sample hold } SGIF_CALCMODE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- Measurement Unit Display Acquisition

Format	RC WINAPI SGIF_GetDisplayUnit(IN int DeviceID, IN int OutNo, OUT SGIF_DISPLAYUNIT * DisplayUnit);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. DisplayUnit Display units (0: 0.01 mm, 1: 0.001 mm, 2: 0.0001 mm, 3: 0.00001 mm, 4: 0.1 μm, 5: 0.01 μm, 6: 0.001 μm). <pre>typedef enum { SGIF_DISPLAYUNIT_0000_01MM = 0, // 0.01mm SGIF_DISPLAYUNIT_000_001MM, // 0.001mm SGIF_DISPLAYUNIT_00_0001MM, // 0.0001mm SGIF_DISPLAYUNIT_0_00001MM, // 0.00001mm SGIF_DISPLAYUNIT_00000_1UM, // 0.1um SGIF_DISPLAYUNIT_0000_01UM, // 0.01um SGIF_DISPLAYUNIT_000_001UM, // 0.001um } SGIF_DISPLAYUNIT;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- Synchronization Setting Acquisition

Format	RC WINAPI SGIF_GetSynchronization(IN int DeviceID, IN int OutNo, OUT BOOL * OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. OnOff Switch, 0: off, 1: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Storage (OUT number setting) Acquisition**

Format	RC WINAPI SGIF_GetStorageTarget(IN int DeviceID, IN int OutNo, OUT BOOL * OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OutNo Output OUT number, range 1 to 4. OnOff Switch, 0: off, 1: on.
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Get the specified OUT on or off status for data storage.
Version	V1.0

● Sampling Period Acquisition

Format	RC WINAPI SGIF_GetSamplingCycle(IN int DeviceID, OUT SGIF_SAMPLINGCYCLE * SamplingCycle);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. SamplingCycle Sampling period (SG5000 series: 0: 590 khz, 1: 400 khz, 2: 200 khz, 3: 88kHz, 4: 50kHz, 5: 20kHz, 6: 10kHz, 7: 5kHz, 8: 1kHz; SG3000 series: 10: 88kHz, 11: 50kHz, 12: 20kHz, 13: 10kHz, 14: 5kHz, 15: 1kHz). <pre>typedef enum { //SG5000 series SGIF_5000_SAMPLINGCYCLE_590KHZ, // 590kHz SGIF_5000_SAMPLINGCYCLE_400KHZ, // 400kHz SGIF_5000_SAMPLINGCYCLE_200KHZ, // 200kHz SGIF_5000_SAMPLINGCYCLE_88KHZ, // 88kHz SGIF_5000_SAMPLINGCYCLE_50KHZ, // 50kHz SGIF_5000_SAMPLINGCYCLE_20KHZ, // 20kHz SGIF_5000_SAMPLINGCYCLE_10KHZ, // 10kHz SGIF_5000_SAMPLINGCYCLE_5KHZ, // 5kHz SGIF_5000_SAMPLINGCYCLE_1KHZ, // 1kHz+ //SG3000 series SGIF_3000_SAMPLINGCYCLE_88KHZ = 10, // 88kHz SGIF_3000_SAMPLINGCYCLE_50KHZ, // 50kHz SGIF_3000_SAMPLINGCYCLE_20KHZ, // 20kHz SGIF_3000_SAMPLINGCYCLE_10KHZ, // 10kHz SGIF_3000_SAMPLINGCYCLE_5KHZ, // 5kHz SGIF_3000_SAMPLINGCYCLE_1KHZ, // 1kHz } SGIF_SAMPLINGCYCLE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Prevent Mutual Interference Acquisition

Format	RC WINAPI SGIF_GetMutualInterferencePrevention(IN int DeviceID, OUT SGIF_MUTUAL_INTERFERENCE_PREVENTION * OnOff);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. OnOff Switch (0: off, 1: AB-on, 2: ABC-on). typedef enum { SGIF_MUTUAL_INTERFERENCE_PREVENTION_OFF, // OFF SGIF_MUTUAL_INTERFERENCE_PREVENTION_AB_ON, // AB-ON SGIF_MUTUAL_INTERFERENCE_PREVENTION_ABC_ON, // ABC-ON } SGIF_MUTUAL_INTERFERENCE_PREVENTION;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Gating Time Acquisition

Format	RC WINAPI SGIF_GetStrobeTime(IN int DeviceID, OUT SGIF_STOROBETIME * StrobeTime);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. StrobeTime Output form (0: 2 ms, 1: 5 ms, 2: 10ms, 3: 20 ms). typedef enum { SGIF_STOROBETIME_2MS, // 2ms SGIF_STOROBETIME_5MS, // 5ms SGIF_STOROBETIME_10MS, // 10ms SGIF_STOROBETIME_20MS, // 20ms } SGIF_STOROBETIME;
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Data Storage Acquisition

Format	RC WINAPI SGIF_GetDataStorage(IN int DeviceID, OUT int * NumStorage, OUT SGIF_STORAGECYCLE * StorageCycle);
Parameters	DeviceID Specifies which communication device is used for communication, ranging from 0 to 10. NumStorage With stored data number, range: 0- 1200000. StorageCycle Storage period (0: 1x, 1: 2x, 2: 5x, 3: 10x, 4: 20x, 5: 50x, 6: 100x, 7: 200x, 8: 500x, 9: 1000x, 10: synchronous input). <pre>typedef enum { SGIF_STORAGECYCLE_1, // sampling rate x 1 SGIF_STORAGECYCLE_2, // sampling rate x 2 SGIF_STORAGECYCLE_5, // sampling rate x 5 SGIF_STORAGECYCLE_10, // sampling rate x 10 SGIF_STORAGECYCLE_20, // sampling rate x 20 SGIF_STORAGECYCLE_50, // sampling rate x 50 SGIF_STORAGECYCLE_100, // sampling rate x 100 SGIF_STORAGECYCLE_200, // sampling rate x 200 SGIF_STORAGECYCLE_500, // sampling rate x 500 SGIF_STORAGECYCLE_1000, // sampling rate x 1000 SGIF_STORAGECYCLE_TIMING, // Timing sync } SGIF_STORAGECYCLE;</pre>
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

● Active OUT Count Acquisition

Format	EXP RC WINAPI SGIF_GetNumofUsedHeads(IN int DeviceID, OUT int * NumofUsedHeads).
Parameters	DeviceID Device ID number, ranging 0-63. NumofUsedHeads Active OUT/sensor head counting
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Get OUT count value
Version	V1.0

● Active Sensor Head Count Acquisition

Format	EXP RC WINAPI SGIF_GetNumofusedout(IN int DeviceID, OUT int * Numofusedout).
Parameters	DeviceID Device ID number, ranging 0-63. NumofUsedHeads Active OUT/sensor head counting
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Autonomous sensor head count acquisition
Version	V1.0

- **Read Encoder Parameters (only supported by SCI devices)**

Format	RC WINAPI SGIF_GetEncoderParameters(IN int DeviceID, IN int OutNo, OUT int * InputMode, OUT int * InputTime, OUT int * interval, OUT int * OperatingMode, OUT int * DetectionDirection);
Parameters	DeviceID Device ID number, ranging 0-63. OutNo Encoder number (1-3) InputMode Input mode (0-3) InputTime Minimum input time (100, 200, 500, 1000, 2000, 5000, 10000, 20000) (ns) Interval Interval (1-10000) OperatingMode Operation mode (0-2) DetectionDirection Detection direction (0-2)
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	Read the current encoder number settings (1, 2, 3)
Version	V1.0

- **Get Analog Output Channel Number (only supported by SCI devices)**

Format	RC WINAPI SGIF_GetAnalogoutputchannel(IN int DeviceID, IN int OutNo, Our int * OutBuffer);
Parameters	DeviceID Device ID number, ranging 0-63. OutNo Analog number (1-3) OutBuffer Data pointer
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0

- **Get I/O Output Channel Number (only supported by SCI devices)**

Format	RC WINAPI SGIF_GetIooutputchannel(IN int DeviceID,IN int OutNo, Our int *OutBuffer);
Parameters	DeviceID Device ID number, ranging 0-63. OutNo Digital output number (1-4) OutBuffer Data pointer
Return value	> 0: Failure 0: Success, please refer to " 5 Function Return Code Definition " for details.
Description	-
Version	V1.0