

SGLI Product I/O Tool Kit (SGTK)
Operation manual

First Edition

December, 2018



Table of Contents

1 HDF Library and SGLI I/O Tool Kit.....	1-1
1.1 SGLI Product Information	1-1
1.2 About SGLI	1-2
2 Installing HDF Library and SGLI.....	2-1
2.1 Installing HDF Library.....	2-1
2.1.1 Getting HDF5 Library.....	2-1
2.1.2 How to Install the HDF5 Library (for Linux).....	2-1
2.1.3 How to Install the HDF5 Library (for Windows)	2-2
2.2 Installing SGLI.....	2-3
2.2.1 Installing SGLI for Linux.....	2-3
2.2.2 Installing SGLI for Windows	2-4
3 Programming using SGLI	3-1
3.1 Flow of Programming.....	3-2
4 Function Composition	4-1
5 Function of SGLI	5-1
5.1 C- Language.....	5-1
5.1.1 Common function	5-1
5.1.2 Data get function	5-2
5.1.3 Data Set Function.....	5-12
5.1.4 Other Function.....	5-21
5.2 Fortran Language	5-24
5.2.1 Common Function	5-24
5.2.2 Data Get Function	5-25
5.2.3 Data Set Function.....	5-34
5.2.4 Other function.....	5-43
6 File Definition.....	6-1
6.1 Configuration File	6-1
6.2 Leap Second File.....	6-3
6.3 Access Label Parameter File	6-4
7 In/Out-Put Data.....	7-1
7.1 Data Definition	7-1
7.1.1 Dataset Access Label	7-1
8 Error Number ID	8-1
9 Appendix.....	9-1
9.1 Sample Program (sample_01)	9-1
9.1.1 Linux	9-1
9.1.2 Windows.....	9-2

1 HDF Library and SGLI I/O Tool Kit

SGLI I/O Tool kit (SGTK) is offered to use SGLI product data, which is the data of Global Change Observation Mission 1st –Climate, with a software program developed by C-language or FORTRAN. SGLI product is classified into Level 1 to 3 and the data stored in the data format HDF (Hierarchical Data Format). SGTK can be easily accessed to the SGLI product data of HDF.

1.1 SGLI Product Information

SGLI product is provided as a file in HDF5 format.

HDF (Hierarchical Data Format) is the data format which doesn't depend on the user's computer composition developed by NCSA (The National Center for Supercomputing Applications: Illinois University) to use the information. The HDF file has the HDF4 format and the HDF5 format, and HDF5 completely reviews the problem of HDF4 (data type of two or more kinds of that there is a limitation in the data size etc.) and is being offered.

SGTK accesses the file of HDF5, and inputs and outputs information specified by the user program.

The HDF file is divided into Attribute part containing attribute information, Dataset part containing product information, and Group part including them. The Attribute part of SGLI product contains the metadata and the Dataset part contains the product data such as observation data and latitude longitude data. Group part is a similar elements in the directory of the file system, multiple classification such as "geometric information" metadata will be used as a group together the datasets.

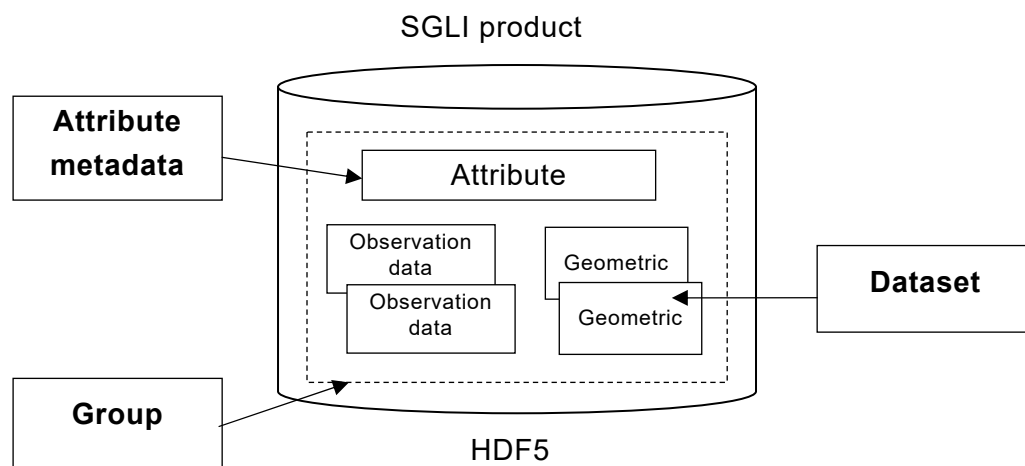


Figure 1-1 HDF File Composition

Access to the HDF, open the HDF file, Attribute by specifying the ID returned by this open, and output of Dataset of information. In order to identify a plurality of Dataset in SGTK, to identify the Dataset using the path to any element or access label, (provided as specified alias of the path. Identifier on each programming language).

1.2 About SGTK

SGTK is a toolkit developed to access the SGLI product data, the DEM file, and the weather file easily on C language and the Fortran language program. SGTK is composed of the I/O function group of SGLI product information.

The way to access to HDF is to open the HDF file, specify identifier (hid_t) returned by this opening, and input and output information on Attribute and each Dataset. Dataset is specified by using the HDF identification number (Please refer to Chapter 6.1.1 for details) to identify two or more Dataset in SGTK

(1) Requirements

SGTK runs under the environment shown in Table 1-1.

Table 1-1 Requirements

Item	DELL		DELL		DELL	
OS	Red Hat Enterprise Linux 6.3 Linux 2.6 (32bit, 64bit)		Windows 8.1 (32bit)		Windows 8.1 (64bit)	
compiler	C/C++ Language	Fortran	C/C++ Language	Fortran	C/C++ Language	Fortran
	gcc, g++ Intel C++ Compiler	g77, g95, gfortran f77, f95 Intel Fortran Compiler	Visual Studio 2013, 2015	G95	Visual Studio 2013, 2015	G95
RAM	12GB		4GB		4GB	
HDF Library	HDF5-1.8.12					

(*1).

SGTK using g95 compiler is shown below website.

Linux x86_64/EMT64 (32 bit D.I.) Default integer of 32 bits, compatible with older programs

(2) Platform

SGTK supports both 32 / 64 bits machines.

(3) Language and Compiler

SGTK is used with language and compiler to show in Table 1-2.

Table 1-2 Language and Compiler

OS	C/C++ Language	Fortran Language	
		Fortran77	Fortran90/95
RedHat	gcc, g++	gfortran	gfortran, G95
Enterprise Linux	Intel C++ Compiler	Intel Fortran Compiler	
Windows8	Visual Studio 2015 (cl.exe)	-	G95

(4) Language and Compiler

Real effective accuracy is responsible for verification of value to significant digits that depends on the floating-point type precision indicated by the data set. Specifically, single precision: 7-digit, double precision: 15-digit.

(5) Byte order

The handling of byte order of HDF5 that is input and output by SGTK is shown below.

- Input : It follows the byte order that is stored in HDF5.
- Output : If it overwrites, it follows the byte order that is stored in HDF5. If it creates a new, it uses the little-endian as the default. (It is also possible to change to the big-endian by setting SGTK.)

2 Installing HDF Library and SGTK

The SGLI product data is made by using Release 1.8.12 Patch1 in the HDF5 library. This section explains the installation of Release 1.8.12 Patch1 in the HDF5 library.

2.1 Installing HDF Library

Compile the source code of the HDF library to install it.
Use the source file prepared for each platform.

2.1.1 Getting HDF5 Library

The HDF5 library can be downloaded from the website of the HDF group. You can get the HDF5 library Release 1.8.21 (released day is June 5 the 2018) from the following URL.
<http://www.hdfgroup.org/HDF5/release/obtain51812.html>

The HDF5 library downloaded as of August 2018 is shown in Table 2-1. Please obtain the HDF5 library corresponding to your computer and OS.

Table 2-1 HDF5 Library

Platform	Download File name	Remark
Linux	hdf5-1.8.12.tar.gz	Source code
Windows	CMake-hdf5-1.8.12.zip	Source code (CMake version) For Visual Studio 2013/2015/2017, installing batch files for 32bit and 64bit are prepared.

2.1.2 How to Install the HDF5 Library (for Linux)

The installation procedure is described below.

(1) Getting library

Download the source code to compile and install.

(2) Installing the HDF5 library (source code)

Decompress the hdf5-1.8.2.tar.gz file to extract it.

```
$ gunzip hdf5-1.8.21.tar.gz  
$ tar xzf hdf5-1.8.21.tar
```

When the above command is executed, a directory of hdf5-1.8.21 is created. The example of specifying the directory of "/HDF5/shared" as the installation destination and installing the HDF5 library is shown below.

```
$ cd hdf5-1.8.21
$ mkdir build
$ cd build
$ ../configure --prefix=/HDF5/shared --enable-cxx
$ make
$ make check
$ make install
```

Specify the installation directory (/HDF5/shared) as the argument of Configure.
Specify c++.

2.1.3 How to Install the HDF5 Library (for Windows)

The installation procedure is described below.

(1) Installing the HDF5 library (source code)

(a) Install the cmake.

(Build system for cross platform. <http://www.cmake.org/>)

(b) Decompress the CMake-hdf5-1.8.21.zip and extract it.

(c) The following batch files for installation are prepared in the directory CMake-hdf5-1.8.21 where the file is extracted.

Select the combination of Visual Studio's version and Windows 32 bit/64bit, and execute it.

- build-VS2013-32.bat
- build-VS2013-64.bat
- build-VS2015-32.bat
- build-VS2015-64.bat
- build-VS2017-32.bat
- build-VS2017-64.bat

In case of the installation for Windows (64bit) using Visual Studio 2015 in the directory CMake-hdf5-1.8.21 extracted just below the drive C.
(Directory of C:\CMake-hdf5-1.8.21\build is created and various files are generated.)

```
C:\CMake-hdf5-1.8.21> build-VS2015-64.bat
```

2.2 Installing SGTK

2.2.1 Installing SGTK for Linux

(1) Extracting the file

Decompress the SGTK_SGLI_Ver1.01.tar.gz and extract it.

```
$ tar xzf SGTK_SGLI_Ver1.01.tar.gz
```

When the above command is executed, the directory of SGTK_SGLI is created.

In this directory, the files and directories listed in the Table 2.2-1 are extracted.

Table 2-2 Contents in SGTK

Name	File or Directory	Contents	Remark
include/	Directory	Include file of SGTK is stored.	
lib/	Directory	Library of SGTK is created.	
linux/	Directory	Installer for Linux is stored.	Makefile
msvc/	Directory	Installer for Windows is stored.	VisualStudio project file
src/	Directory	Source code of SGTK is stored.	

(2) Installing library

Move to the directory of "linux" in SGTK to execute the make command.

Please change the following paths listed in MakeFile according to your computer and OS.

- ROOTDIR: HDF5 library is installed in this path. (/HDF5/shared)
- INCDIR: Header file of HDF5 library is stored in this path.

```
$ vi Makefile
ROOTDIR =/HDF5/shared
INCDIR= -I$(ROOTDIR)/include -I../include
$ make
```

Confirm the library. If "libsgtk.a" is created under lib directory, creation of the SGTK library is complete.

2.2.2 Installing SGTK for Windows

(1) Extracting the file

Decompress the SGTK_SGLI_Ver1.01.zip, and extract it.

When the above command is executed, the directory of SGTK_SGLI is created. In this directory, the files and directories listed in the Table 2-2 are extracted.

Table 2.2-1 Contents in SGTK

Name	File or directory	Contents	Remark
include/	Directory	Include file of SGTK is stored.	
lib/	Directory	Library of SGTK is created.	
linux/	Directory	Installer for Linux is stored.	Makefile
msvc/	Directory	Installer for Windows is stored.	VisualStudio project file
src/	Directory	Source code of SGTK is stored.	

(2) Installing library

Move to the directory of “msvc” in SGTK to start VisualStudio from “sgtk.sln”.

Open [sgtk property page] from [Property (P)] in [Project (P)] menu.

Select [VC++ directory].

Specify the following four directories where HDF5 is installed in [include directory]. These directories must be separated by semicolon (;).

```
C:¥CMake-hdf5-1.8.21¥build
C:¥CMake-hdf5-1.8.21¥ hdf5-1.8.21¥src
C:¥CMake-hdf5-1.8.21¥ hdf5-1.8.21¥hl¥src
C:¥CMake-hdf5-1.8.21¥ hdf5-1.8.21¥c++¥src
```

```
$(VC_IncludePath);$(WindowsSDK_IncludePath);
Add the above content.
(There are cases where $(WindowsSDK_IncludePath) does not exist.)
C:¥CMake-hdf5-1.8.21¥build;C:¥CMake-hdf5-1.8.21¥ hdf5-1.8.21¥src;C:¥CMake-hdf5-1.8.21¥ hdf5-1.8.21¥c++¥src;C:¥CMake-hdf5-1.8.21¥ hdf5-1.8.21¥hl¥src
; $(VC_IncludePath);$(WindowsSDK_IncludePath);
```

Confirm the library after completing the build successfully. If “SGTK.lib” is created under the “Release¥Win64 (Win32 in case of 32bit)” folder generated in the same hierarchy as the current folder, creation of the SGTK library is complete.

3 Programming using SGTK

The setting for Linux is listed in Table 3-1.

Table 3-1 Setting for Linux

Item	Name	Description
Installation destination	/HDF5/shared/ /SGTK_SGLI/	Description when the installation destination of HDF5 and IO tool is the directory describing in the left column.
Include path	/HDF5/shared/include /SGTK_SGLI/include	Write to Makefile Example: -I/HDF5/shared/include -I/SGTK_SGLI/include
Library path	/HDF5/shared/lib /SGTK_SGLI/lib	Write to Makefile Example: -L/HDF5/shared/lib -L/SGTK_SGLI/lib
Library name	sgtk hdf5 hdf5_cpp hdf5_hl hdf5_hl_cpp	Write to Makefile Example: -lsgtk -lhdf5 -lhdf5_cpp -lhdf5_hl -lhdf5_hl_cpp
Environmental variable required at the execution	LD_LIBRARY_PATH	Add library path export LD_LIBRARY_PATH=/HDF5/shared/lib :/SGTK_SGLI/lib:\$LIBRARY_PATH

The setting for Linux is listed in Table 3-2.

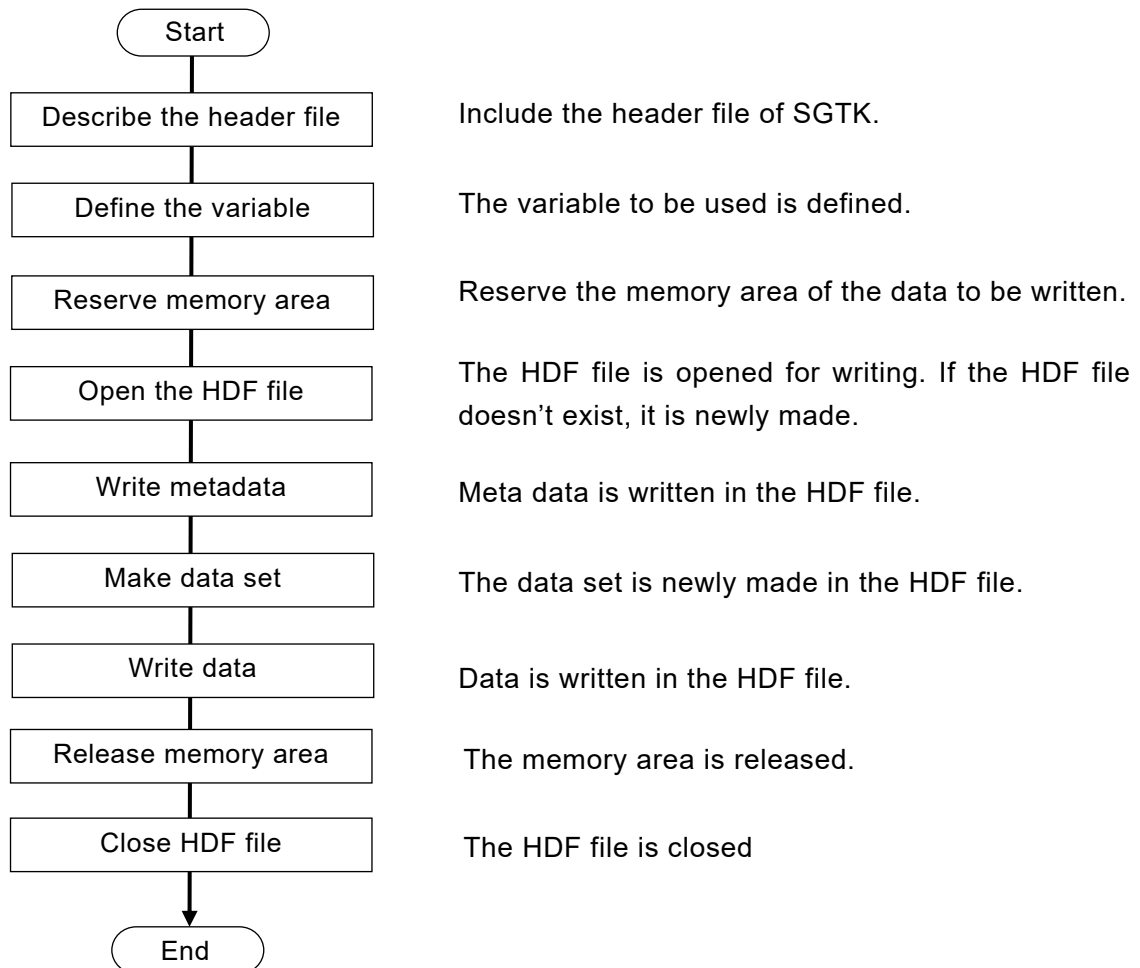
Table 3-2 Setting for Windows

Item	Name	Description
Installation destination	C:\%CMake-hdf5-1.8.21 C:\%SGTK_SGLI	
Include path	C:\%CMake-hdf5-1.8.21\%build C:\%CMake-hdf5-1.8.21\%hdf5-1.8.21\%src C:\%CMake-hdf5-1.8.21\%hdf5-1.8.21\%hl\%src C:\%CMake-hdf5-1.8.21\%hdf5-1.8.21\%c++\%src C:\%SGTK_SGLI\%include	Add these paths in the left column by separating with semicolon to the include directory in the property page.
Library path	C:\%CMake-hdf5-1.8.21\%build\%bin\%Release C:\%SGTK_SGLI\%lib	Add these paths in the left column by separating with semicolon to the library directory in the property page.
Library name	sgtk.lib hdf5.lib hdf5_cpp.lib hdf5_hl.lib hdf5_hl_cpp.lib	Add these paths to the dependent file of addition of linker in the property page.
Environmental variable required at the execution	PATH Set PATH=%PATH%;C:\%CMake-hdf5-1.8.21\%build\%bin\%Release;C:\%SGTK_SGLI\%lib	Set the library path to read DLL at the execution.

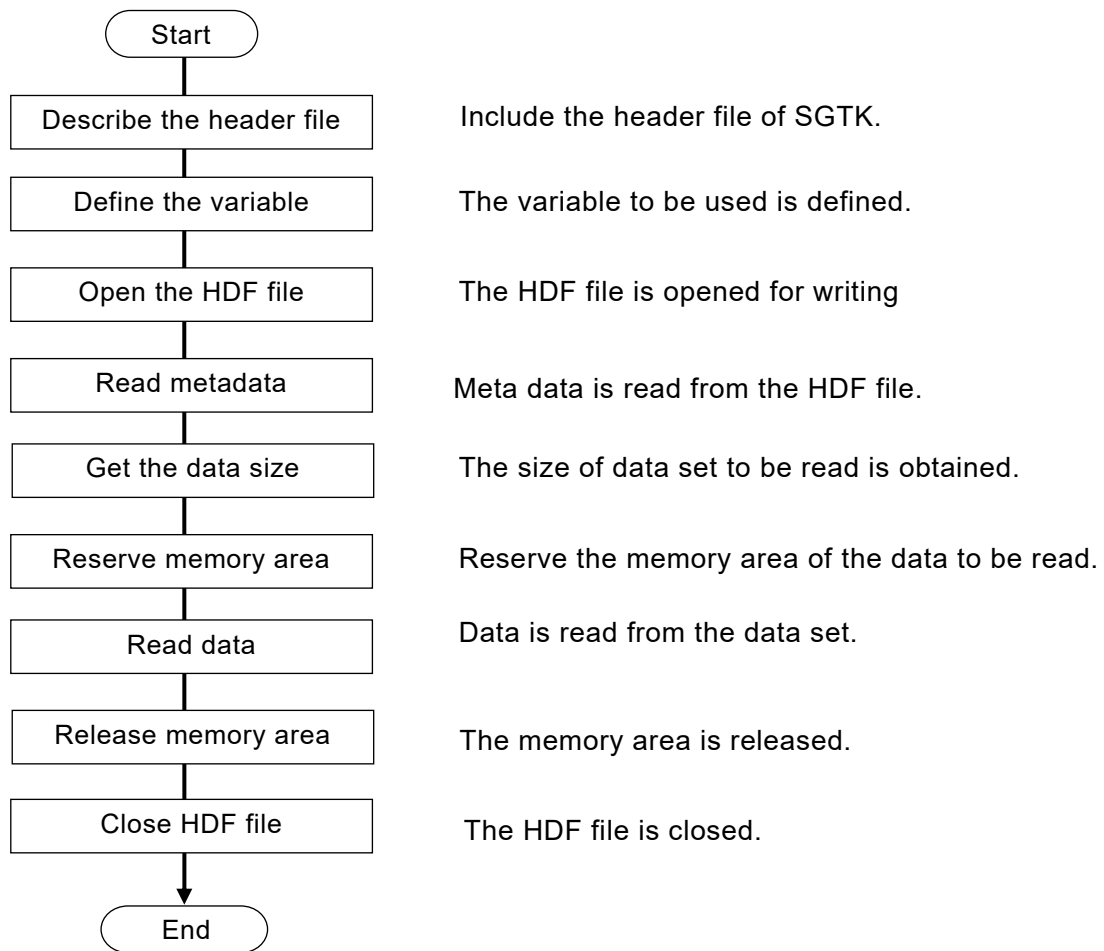
3.1 Flow of Programming

The flow that SGLI data is read and written in C and Fortran by using SGK is shown below.

(1) Making new HDF file



(2) Reading the data from HDF file



4 Function Composition

Function composition of SGK is shown in Figure 4-1.

SGTK provides input function and output function.

(1) Input function

(a) Level 1 product (L1A, L1B) input function

(b) Level 2 product (L2) input function

(c) Level 3 product (L3) input function

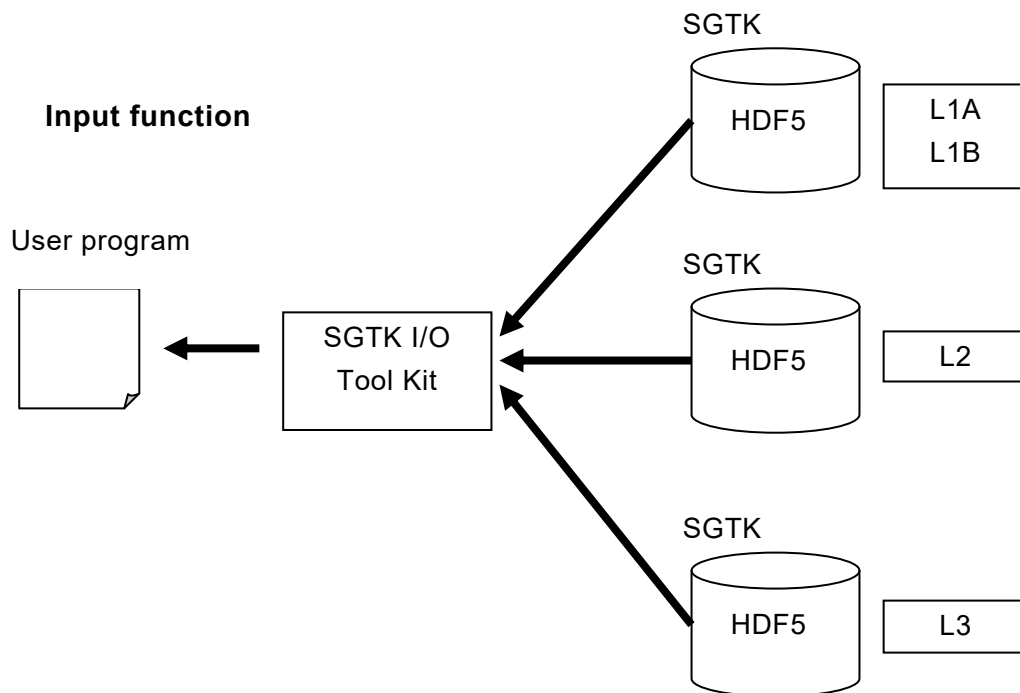


Figure 4-1 SGK Product Input Function

- (2) Output function
 - (a) Level 1 product (L1A, L1B) output function
 - (b) Level 2 product (L2) output function
 - (c) Level 3 product (L3) output function

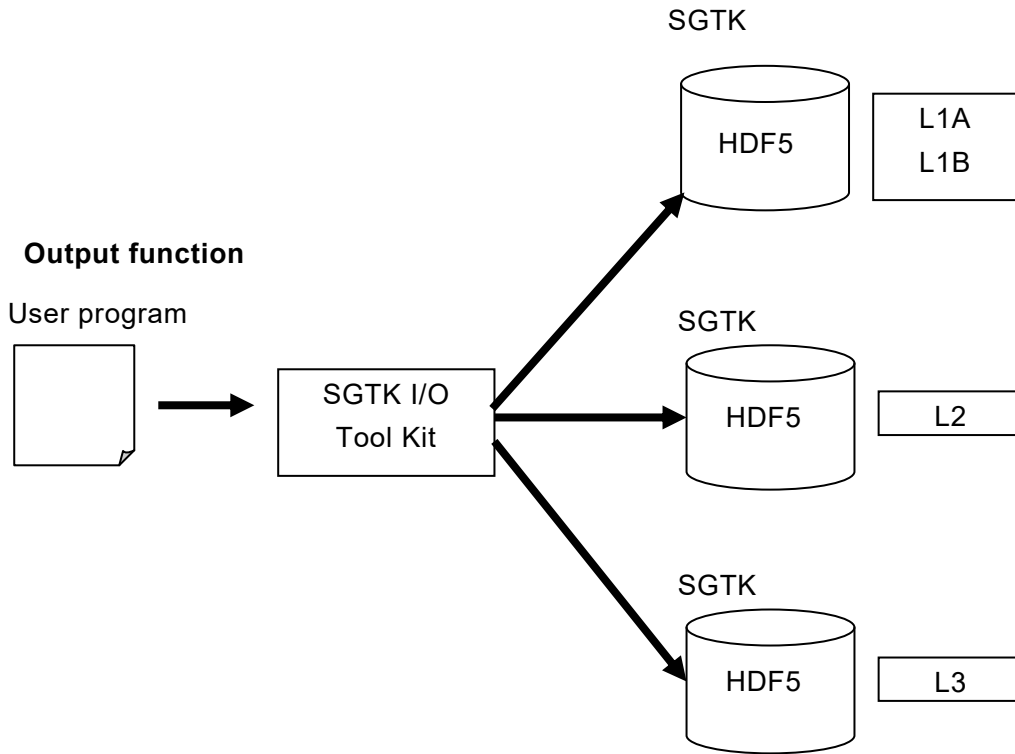


Figure 4-2 SGK Product Output Function

5 Function of SGTK

5.1 C- Language

5.1.1 Common function

(1) L1A, L1B, L2, L3 input functions

Read processing of parameter file that defines access label.				
Read access label definition parameter to create parameter table.				
int SGTK_make_parameter_table (const std::string parameter_file)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal: 0 Error: A negative value returns.
parameter				
parameter_file	const std::string	input	1	Parameter file name

Product file opening (reading only)				
The HDF product file is opened with read only.				
hid_t SGTK_open_file(char *file_name)				
Name	Type	IN/OUT	Size	Descriptions
return value				
HDF_file_id	hid_t	output	1	Normal:HDF access file id is returned Error:A negative value returns when it is not possible to acquire it.
parameter				
file_name	char *	input	1	SGLI HDF file name

Product file close				
The HDF product file is closed.				
int SGTK_close_file(hid_t HDF_file_id)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id

5.1.2 Data get function

(1) L1A, L1B, L2, L3 data get function

Dimension information of data set getting				
The data set dimension information is got from product.				
- When the access label intended for plural Datasets is specified, the error is returned.				
int SGK_get_dims (hid_t HDF_file_id, char *label, int **num)				
Name	Type	IN/OUT	Size	Descriptions
return value				
num	int	output	1	The array number of dimension returns.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Please refer to "6.1.1 the HDF access label".
num	int **	output	N	Dimension Information (1 ~ 6 dimension information)

Acquisition of meta data (meta data name)				
The meta data (string type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_string(hid_t HDF_file_id, char *label, char *meta, char *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	char *	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (int type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_int(hid_t HDF_file_id, char *label, char *meta, int *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	int *	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (int8 type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_int8(hid_t HDF_file_id, char *label, char *meta, int8 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	int8 *	output	1	meta data value

Acquisition of meta data (meta data name)

The meta data (int16 type) is acquired by specifying the meta data name of the product.
 - When the memory area where the meta data value is stored is NULL, the error is returned.

int SGKTK_get_meta_int16(hid_t HDF_file_id, char *label, char *meta, int16 *value)

Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	int16 *	output	1	meta data value

Acquisition of meta data (meta data name)

The meta data (int32 type) is acquired by specifying the meta data name of the product.
 - When the memory area where the meta data value is stored is NULL, the error is returned.

int SGKTK_get_meta_int32(hid_t HDF_file_id, char *label, char *meta, int32 *value)

Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	int32 *	output	1	meta data value

Acquisition of meta data (meta data name)

The meta data (uint type) is acquired by specifying the meta data name of the product.
 - When the memory area where the meta data value is stored is NULL, the error is returned.

int SGKTK_get_meta_uint(hid_t HDF_file_id, char *label, char *meta, uint *value)

Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	uint *	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (uint8 type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_uint8(hid_t HDF_file_id, char *label, char *meta, uint8 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	uint8 *	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (uint16 type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_uint16(hid_t HDF_file_id, char *label, char *meta, uint16 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	uint16 *	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (uint32 type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_uint32(hid_t HDF_file_id, char *label, char *meta, uint32 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	uint32 *	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (float type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_float(hid_t HDF_file_id, char *label, char *meta, float *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	float *	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (float32 type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_float32(hid_t HDF_file_id, char *label, char *meta, float *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	float *	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (float64 type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
int SGK_get_meta_float64(hid_t HDF_file_id, char *label, char *meta, double *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	meta data name
value	double *	output	1	meta data value

Acquisition of Integer Type data				
The data of integer Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
Int SGK_get_dataset_int(hid_t HDF_file_id, char *label, int **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
get_value	int **	output	N	Memory area for storing data

Acquisition of Integer Type data				
The data of int8 Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
Int SGK_get_dataset_int8(hid_t HDF_file_id, char *label, int8 **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
get_value	int8 **	output	N	Memory area for storing data

Acquisition of Integer Type data				
The data of int16 Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
Int SGK_get_dataset_int16(hid_t HDF_file_id, char *label, int16 **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	Input	1	HDF access file id
label	char *	input	1	Data set access label
get_value	int16 **	output	N	Memory area for storing data

Acquisition of Integer Type data				
The data of int32 Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
Int SGK_get_dataset_int32(hid_t HDF_file_id, char *label, int32 **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
get_value	int32 **	output	N	Memory area for storing data

Acquisition of Integer Type data				
The data of uint Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
Int SGK_get_dataset_uint(hid_t HDF_file_id, char *label, uint **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
get_value	uint **	output	N	Memory area for storing data

Acquisition of Integer Type data				
The data of uint8 Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
Int SGK_get_dataset_uint8(hid_t HDF_file_id, char *label, uint8 **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
get_value	uint8 **	output	N	Memory area for storing data

Acquisition of Integer Type data				
The data of uint16 Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
Int SGK get_dataset_uint16(hid_t HDF_file_id, char *label, uint16 **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	Input	1	HDF access file id
label	char *	input	1	Data set access label
get_value	uint16 **	output	N	Memory area for storing data

Acquisition of Integer Type data				
The data of uint32 Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
Int SGK get_dataset_uint32(hid_t HDF_file_id, char *label, uint32 **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
get_value	uint32 **	output	N	Memory area for storing data

Acquisition of Integer Type data				
The data of float Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK get_dataset_float(hid_t HDF_file_id, char *label, int kind, float **get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
kind	int	input	1	Brightness / reflectance / other identification
get_value	float **	output	N	Memory area for storing data

Acquisition of Integer Type data

The data of float32 Type is acquired specifying the HDF access label.
 - Memory area for storing data, the user using the function to ensure.

int SGKTK_get_dataset_float32(hid_t HDF_file_id, char *label, int kind, float **get_value)

Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
kind	int	input	1	Brightness / reflectance / other identification
get_value	float **	output	N	Memory area for storing data

Acquisition of Integer Type data

The data of float64 Type is acquired specifying the HDF access label.
 - Memory area for storing data, the user using the function to ensure.

int SGKTK_get_dataset_float64(hid_t HDF_file_id, char *label, int kind, double **get_value)

Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
Kind	int	input	1	Brightness / reflectance / other identification
get_value	double **	output	N	Memory area for storing data

Acquisition of latitude and longitude data

It gets the latitude and longitude data in the range of the specified scan number

- Scan number is specified and end scan number start scan number.
Exit scan number, if it is less than the start scan number, an error is returned.
If greater than the scan numbers ending scan number is stored in the specified HDF data set is, I get to data that is stored in the HDF data set.
- Memory area for storing data, the user using the function to ensure.

int SGTK_get_dataset_latlon(hid_t HDF_file_id, char *label_lat, char *label_lon, int kind, int start_no, int end_no, float **latlon)

Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error : A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label_lat	char *	input	1	Data set access label (latitude)
label_lon	char *	input	1	Data set access label (longitude)
kind	int	input	1	Memory area for storing data
start_no	int	input	1	Acquisition start the scan number
end_no	int	input	1	Acquisition end the scan number
latlon	float **	output	1	Memory area for storing data

5.1.3 Data Set Function

(1) L1A, L1B, L2, L3 data set function

HDF Dataset making				
HDF Dataset of data dimension size is made.				
— When generating it, HDF Dataset is initialized by the missing value.				
int SGTK_set_dims(int HDF_file_id, char *label, int *dims, int rank, int dat_type, char **dim_lbl)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:SUCCESS(0) returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char*	input	1	Data set access label
dims	int*	input	1	Array of dimension
rank	int	input	1	Number of dimension
dat_type	int	input	1	HDF5 data type
dim_lbl	char **	input	1	Dimension label

Meta data setting				
The meta data of string type is set by the meta data name of the HDF product.				
int SGTK_set_meta_string(hid_t HDF_file_id, char *label, char *meta, char *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	char *	input	1	Buffer address storing meta data value

Meta data setting				
The meta data of int type is set by the meta data name of the HDF product.				
int SGTK_set_meta_int(hid_t HDF_file_id, char *label, char *meta, int *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	int *	input	1	Setting value

Meta data setting				
The meta data of int8 type is set by the meta data name of the HDF product.				
int SGK_set_meta_int8(hid_t HDF_file_id, char *label, char *meta, int8 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	int8 *	input	1	Setting value

Meta data setting				
The meta data of int16 type is set by the meta data name of the HDF product.				
int SGK_set_meta_int16(hid_t HDF_file_id, char *label, char *meta, int16 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	Int16 *	input	1	Setting value

Meta data setting				
The meta data of int32 type is set by the meta data name of the HDF product.				
int SGK_set_meta_int32(hid_t HDF_file_id, char *label, char *meta, int32 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	Int32 *	input	1	Setting value

Meta data setting				
The meta data of uint type is set by the meta data name of the HDF product.				
int SGK_set_meta_uint(hid_t HDF_file_id, char *label, char *meta, uint *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	uint *	input	1	Setting value

Meta data setting				
The meta data of uint8 type is set by the meta data name of the HDF product.				
int SGK_set_meta_uint8(hid_t HDF_file_id, char *label, char *meta, uint8 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	uint8 *	input	1	Setting value

Meta data setting				
The meta data of uint16 type is set by the meta data name of the HDF product.				
int SGK_set_meta_uint16(hid_t HDF_file_id, char *label, char *meta, uint16 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	uint16 *	input	1	Setting value

Meta data setting				
The meta data of uint32 type is set by the meta data name of the HDF product.				
int SGK_set_meta_uint32(hid_t HDF_file_id, char *label, char *meta, uint32 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	uint32 *	input	1	Setting value

Meta data setting				
The meta data of float type is set by the meta data name of the HDF product.				
int SGK_set_meta_float(hid_t HDF_file_id, char *label, char *meta, float *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	float *	input	1	Setting value

Meta data setting				
The meta data of float32 type is set by the meta data name of the HDF product.				
int SGK_set_meta_float32(hid_t HDF_file_id, char *label, char *meta, float *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	float *	input	1	Setting value

Meta data setting				
The meta data of float64 type is set by the meta data name of the HDF product.				
int SGK_set_meta_float64(hid_t HDF_file_id, char *label, char *meta, double *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
meta	char *	input	1	Meta data name
value	double *	input	1	Setting value

Data set setting				
The data of int Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_int(hid_t HDF_file_id, char *label, int *dims, int rank, int *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	int *	input	1	Buffer address storing data

Data set setting				
The data of int8 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_int8(hid_t HDF_file_id, char *label, int *dims, int rank, int8 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	int8 *	input	1	Buffer address storing data

Data set setting				
The data of int16 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_int16(hid_t HDF_file_id, char *label, int *dims, int rank, int16 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	int16 *	input	1	Buffer address storing data

Data set setting				
The data of int32 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_int32(hid_t HDF_file_id, char *label, int *dims, int rank, int32 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	int32 *	input	1	Buffer address storing data

Data set setting				
The data of uint Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_uint(hid_t HDF_file_id, char *label, int *dims, int rank, uint *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
dims	int *	input	1	Dimension information
rank	int	input	1	Number of diminution
value	uint *	input	1	Buffer address storing data

Data set setting				
The data of uint8 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_uint8(hid_t HDF_file_id, char *label, int *dims, int rank, uint8 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	uint8 *	input	1	Buffer address storing data

Data set setting				
The data of uint16 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_uint16(hid_t HDF_file_id, char *label, int *dims, int rank, uint16 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	uint16 *	input	1	Buffer address storing data

Data set setting				
The data of uint32 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_uint32(hid_t HDF_file_id, char *label, int *dims, int rank, uint32 *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	uint32 *	input	1	Buffer address storing data

Data set setting				
The data of float Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGKTK_set_dataset_float(hid_t HDF_file_id, char *label, int kind, int *dims, int rank, float *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
kind	int	input	1	Data type Radiance (SGTK_KIND_RADIANCE=0) Reflectance (SGTK_KIND_REFLECTANCE=1) Other (SGTK_KIND_OTHER=3)
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	float *	input	1	Buffer address storing data

Data set setting				
The data of float32 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGKTK_set_dataset_float32(hid_t HDF_file_id, char *label, int kind, int *dims, int rank, float *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
kind	int	input	1	Data type Radiance (SGTK_KIND_RADIANCE=0) Reflectance (SGTK_KIND_REFLECTANCE=1) Other (SGTK_KIND_OTHER=3)
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	float *	input	1	Buffer address storing data

Data set setting				
The data of float64 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_float64(hid_t HDF_file_id, char *label, int kind, int *dims, int rank, float *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label	char *	input	1	Data set access label
kind	int	input	1	Data type Radiance (SGTK_KIND_RADIANCE=0) Reflectance (SGTK_KIND_REFLECTANCE=1) Other (SGTK_KIND_OTHER=3)
dims	int *	input	1	Dimension information
rank	int	input	1	Number of dimension
value	double *	input	1	Buffer address storing data

Acquisition of latitude and longitude data				
It gets the latitude and longitude data in the range of the specified scan number - Scan number is specified and end scan number start scan number. Exit scan number, if it is less than the start scan number, an error is returned. If greater than the scan numbers ending scan number is stored in the specified HDF data set is, I get to data that is stored in the HDF data set. - Memory area for storing data, the user using the function to ensure.				
int SGK_set_dataset_latlon(hid_t HDF_file_id, char *label_lat, char *label_lon, int kind, int start_no, int end_no, int set_num, float *latlon)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal : The number of getting scan. Error : A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
label_lat	char *	input	1	Data set access label (latitude)
label_lon	char *	input	1	Data set access label (longitude)
kind	int	input	1	Brightness / reflectance / other type
start_no	int	input	1	Acquisition start the scan number
end_no	int	input	1	Acquisition end the scan number
set_num	int	input	1	Number of elements (the number of number × longitude of latitude)
latlon	float **	input	1	Memory area for storing data

5.1.4 Other Function

Group making				
Empty Group is made.				
int SGTK_make_group(hid_t HDF_file_id, char *group)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to set it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
group	char *	input	1	Group name

Time converting (UTC to TAI93)				
The time is converted UTC to TAI93.				
int SGTK_utc_to_tai93(int year, int month, int day, int hour, int min, int sec, int msec, double *tai93time)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to set it.
parameter				
year	int	input	1	Year
month	int	input	1	Month
day	int	input	1	Day
hour	int	input	1	Time
min	int	input	1	Minute
sec	int	input	1	Second
msec	int	input	1	Milisecond
tai93time	double *	outnput	1	TAI93 time

Time converting (TAI93 to UTC)				
The time is converted TAI93 to UTC.				
int SGK_tai93_to_utc(double tai93time, int *year, int *month, int *day, int *hour, int *min, int *sec, int *msec)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to set it.
parameter				
tai93time	double	input	1	TAI93 time
year	int *	output	1	Year
month	int *	output	1	Month
day	int *	output	1	Day
hour	int *	output	1	Time
min	int *	output	1	Minute
sec	int *	output	1	Second
msec	int *	output	1	Millisecond

Memory allocation				
The memory is ensured from the specified address.				
void *SGTK_malloc(int *dims, int rank, int size)				
Name	Type	IN/OUT	Size	Descriptions
return value				
void*	void *	output	1	Memory address that was secured
parameter				
dims	int *	input	1	Dimension information
rank	int	input	1	Dimension number
size	int	input	1	Dimension information size

Memory release				
The memory is ensured from the specified address.				
int SGK_free(void *buf)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error:A negative value returns when it is not possible to set it.
parameter				
buf	void *	input	1	Buffer address of the data storage area

Acquisition of latitude and longitude information				
Get the latitude and longitude from the specified granule ID, and stored in the buffer				
int SGK_get_global_latlon(hid_t HDF_file_id, float *latlon)				
Name	Type	IN/OUT	Size	Descriptions
return value				
data_num	int	output	1	Normal : Acquisition requirement number (latitude number×longitude number) Error: A negative value returns when it is not possible to set it.
parameter				
HDF_file_id	hid_t	input	1	HDF access file id
latlon	float	output	1	latitude and longitude information

Configuration file reflect				
The contents of the specified SGK configuration file is reflected.				
int SGK_load_config(char *file_name)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal:0 Error: A negative value returns when it is not possible to acquire it.
parameter				
file_name	char *	input	1	SGK setting file path

5.2 Fortran Language

5.2.1 Common Function

(1) L1A, L1B, L2, L3 input functions

Read processing of parameter file that defines access label.				
Read access label definition parameter to create parameter table.				
INTEGER SGK_make_parameter_table(parameter_file)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal: 0 Error: A negative value returns.
parameter				
parameter_file	CHARACTER	input	1	Parameter file name

Product file opening (reading only)				
The HDF product file is opened with read only.				
INTEGER SGK_open_file(file_name)				
Name	Type	IN/OUT	Size	Descriptions
return value				
HDF_file_id	INTEGER	output	1	Normal:HDF access file id is returned Error:A negative value returns when it is not possible to acquire it.
parameter				
file_name	CHARACTER	input	N	SGLI HDFfile name

Product file close				
The HDF product file is closed.				
INTEGER SGK_close_file(HDF_file_id)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id

5.2.2 Data Get Function

(1) L1A, L1B, L2, L3 data get function

Dimension information of data set getting				
The data set dimension information is got from product. - When the access label intended for plural Datasets is specified, the error is returned.				
INTEGER SGKTK_get_dims(HDF_file_id, label, size, num)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of array dimensions returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	Input	N	Please refer to "6.1.1 the HDF access label".
size	INTEGER	output	1	Dimension Information size (1 to 6 dimension information)
num	INTEGER	output	1	Dimension Information (1 to 6 dimension information)

Acquisition of meta data (meta data name)				
The meta data (CHARACTOR type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGKTK_get_meta_string(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	CHARACTER	output	N	meta data value

Acquisition of meta data (meta data name)				
The meta data (int type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGKTK_get_meta_int(HDF_file_id, label, meta, p_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	INTEGER	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (INTEGER type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGKTK_get_meta_int8(HDF_file_id, label, meta, p_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
Status	int	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	INTEGER	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (INTEGER type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGKTK_get_meta_int16(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	INTEGER	output	2	meta data value

Acquisition of meta data (meta data name)				
The meta data (INTEGER type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGKTK_get_meta_int32(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	INTEGER	output	4	meta data value

Acquisition of meta data (meta data name)				
The meta data (INTEGER type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGK_get_meta_uint(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	int	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	INTEGER	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (INTEGER type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGK_get_meta_uint8(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	INTEGER	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (INTEGER type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGK_get_meta_uint16(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	INTEGER	output	2	meta data value

Acquisition of meta data (meta data name)				
The meta data (INTEGER type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGK_get_meta_uint32(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	INTEGER	output	4	meta data value

Acquisition of meta data (meta data name)				
The meta data (REAL type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGK_get_meta_float(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	REAL	output	1	meta data value

Acquisition of meta data (meta data name)				
The meta data (REAL type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGK_get_meta_float32(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	REAL	output	4	meta data value

Acquisition of meta data (meta data name)				
The meta data (REAL type) is acquired by specifying the meta data name of the product. - When the memory area where the meta data value is stored is NULL, the error is returned.				
INTEGER SGK_get_meta_float64(int *HDF_file_id, char *label, char *meta, double *value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of meta data word returns. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	meta data name
value	REAL	output	8	meta data value

Acquisition of Integer Type data				
The data of INTEGER Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_int(HDF_file_id, label, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
get_value	INTEGER	output	1	Memory area for storing data

Acquisition of Integer Type data				
The data of INTEGER Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_int8(HDF_file_id, label, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
get_value	INTEGER	output	1	Memory area for storing data

Acquisition of Integer Type data				
The data of INTEGER Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_int16(HDF_file_id, label, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
get_value	INTEGER	output	2	Memory area for storing data

Acquisition of Integer Type data				
The data of INTEGER Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_int32(HDF_file_id, label, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
get_value	INTEGER	output	4	Memory area for storing data

Acquisition of Integer Type data				
The data of INTEGER Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_uint(HDF_file_id, label, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
get_value	INTEGER	output	1	Memory area for storing data

Acquisition of Integer Type data				
The data of INTEGER Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_uint8(HDF_file_id, label, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
get_value	INTEGER	output	1	Memory area for storing data

Acquisition of Integer Type data				
The data of INTEGER Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_uint16(HDF_file_id, label, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	Input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
get_value	INTEGER	output	2	Memory area for storing data

Acquisition of Integer Type data				
The data of INTEGER Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_uint32(HDF_file_id, label, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
get_value	INTEGER	output	4	Memory area for storing data

Acquisition of Integer Type data				
The data of REAL Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_float(HDF_file_id, label, kind, get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
kind	INTEGER	input	1	Brightness / reflectance / other identification
get_value	REAL	output	1	Memory area for storing data

Acquisition of Integer Type data				
The data of REAL Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_float32(HDF_file_id, label, p_kind, p_get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
p_kind	INTEGER	input	1	Brightness / reflectance / other identification
get_value	REAL	output	4	Memory area for storing data

Acquisition of Integer Type data				
The data of REAL Type is acquired specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGK_get_dataset_float64(HDF_file_id, label, p_kind, p_get_value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal : The number of acquisition scan returns. Error:A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
p_kind	INTEGER	input	1	Brightness / reflectance / other identification
p_get_value	REAL	output	8	Memory area for storing data

Acquisition of latitude and longitude data

It gets the latitude and longitude data in the range of the specified scan number

- Scan number is specified and end scan number start scan number.
Exit scan number, if it is less than the start scan number, an error is returned.
If greater than the scan numbers ending scan number is stored in the specified HDF data set is, I get to data that is stored in the HDF data set.
- Memory area for storing data, the user using the function to ensure.

INTEGER SGKTK_get_dataset_latlon(HDF_file_id, latitude, longitude, kind, start_no, end_no, latlon)

Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal: The number of acquisition scan returns. Error: A negative value returns when it is not possible to acquire it
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
latitude	CHARACTER	input	N	Data set access label (latitude)
Longitude	CHARACTER	input	N	Data set access label (longitude)
kind	INTEGER	input	1	Memory area for storing data
start_no	INTEGER	input	1	Acquisition start the scan number
end_no	INTEGER	input	1	Acquisition end the scan number
latlon	REAL	output	1	Memory area for storing data

5.2.3 Data Set Function

(1) L1A, L1B, L2, L3 data set function

HDF Dataset making				
HDF Dataset of data dimension size is made. - When generating it, HDF Dataset is initialized by the missing value.				
INTEGER SGK_set_dims(HDF_file_id, label, p_dims, rank, dat_type, dim_lbl)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Array of dimension
rank	INTEGER	input	1	Number of dimension
dat_type	INTEGER	input	1	HDF5 data type
dim_lbl	CHARACTER	input	N	Dimension label

Meta data setting				
The meta data of CHARACTER type is set by the meta data name of the HDF product.				
INTEGER SGK_set_meta_string(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
Label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	CHARACTER	input	N	Buffer address storing meta data value

Meta data setting				
The meta data of INTEGER type is set by the meta data name of the HDF product.				
INTEGER SGK_set_meta_int(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	INTEGER	input	1	Buffer address storing meta data value

Meta data setting				
The meta data of INTEGER type is set by the meta data name of the HDF product.				
INTEGER SGKTK_set_meta_int8(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	INTEGER	input	1	Buffer address storing meta data value

Meta data setting				
The meta data of INTEGER type is set by the meta data name of the HDF product.				
INTEGER SGKTK_set_meta_int16(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	INTEGER	input	2	Buffer address storing meta data value

Meta data setting				
The meta data of INTEGER type is set by the meta data name of the HDF product.				
INTEGER SGKTK_set_meta_int32(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	Output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	Input	1	HDF access file id
label	CHARACTER	Input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	INTEGER	input	4	Buffer address storing meta data value

Meta data setting				
The meta data of INTEGER type is set by the meta data name of the HDF product.				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	INTEGER	input	1	Buffer address storing meta data value

Meta data setting				
The meta data of INTEGER type is set by the meta data name of the HDF product.				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	INTEGER	input	1	Buffer address storing meta data value

Meta data setting				
The meta data of INTEGER type is set by the meta data name of the HDF product.				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	INTEGER	input	2	Buffer address storing meta data value

Meta data setting				
The meta data of INTEGER type is set by the meta data name of the HDF product.				
INTEGER SGKTK_set_meta_uint32(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	INTEGER	input	1	Buffer address storing meta data value

Meta data setting				
The meta data of REAL type is set by the meta data name of the HDF product.				
INTEGER SGKTK_set_meta_float(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	REAL	input	1	Buffer address storing meta data value

Meta data setting				
The meta data of REAL type is set by the meta data name of the HDF product.				
INTEGER SGKTK_set_meta_float32(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	REAL	input	4	Buffer address storing meta data value

Meta data setting				
The meta data of REAL type is set by the meta data name of the HDF product.				
INTEGER SFTK_set_meta_float64(HDF_file_id, label, meta, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
meta	CHARACTER	input	N	Meta data name
value	REAL	input	8	Buffer address storing meta data value

Data set setting				
The data of INTEGER Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGTK_set_dataset_int(HDF_file_id, label, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Dimension information
rank	INTEGER	input	1	Number of dimension
value	INTEGER	input	1	Buffer address storing data

Data set setting				
The data of INTEGER Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGTK_set_dataset_int8(HDF_file_id, label, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Dimension information
rank	INTEGER	input	1	Number of dimension
value	INTEGER	input	1	Buffer address storing data

Data set setting				
The data of INTEGER Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_int16(HDF_file_id, label, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Dimension information
rank	INTEGER	input	1	Number of dimension
value	INTEGER	input	2	Buffer address storing data

Data set setting				
The data of INTEGER Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_int32(HDF_file_id, label, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Dimension information
rank	INTEGER	input	1	Number of dimension
value	INTEGER	input	4	Buffer address storing data

Data set setting				
The data of INTEGER Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_uint(HDF_file_id, label, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Dimension information
rank	INTEGER	input	1	Number of dimension
value	INTEGER	input	1	Buffer address storing data

Data set setting				
The data of INTEGER Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_uint8(HDF_file_id, label, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Dimension information
rank	INTEGER	input	1	Number of dimension
value	INTEGER	input	1	Buffer address storing data

Data set setting				
The data of INTEGER Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_uint16(HDF_file_id, label, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Dimension information
rank	INTEGER	input	1	Number of dimension
value	INTEGER	input	2	Buffer address storing data

Data set setting				
The data of INTEGER Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_uint32(HDF_file_id, label, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
dims	INTEGER	input	1	Dimension information
rank	INTEGER	input	1	Number of dimension
value	INTEGER	input	4	Buffer address storing data

Data set setting				
The data of REAL Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_float(HDF_file_id, label, kind, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	1	Data set access label
kind	INTEGER	input	1	Dimension information
dims	INTEGER	input	1	Number of dimension
rank	INTEGER	input	1	Buffer address storing data
value	REAL	input	1	HDF access file id

Data set setting				
The data of float32 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_float32(HDF_file_id, label, kind, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
kind	INTEGER	input	1	Dimension information
dims	INTEGER	input	1	Number of dimension
rank	INTEGER	input	1	Buffer address storing data
value	REAL	input	4	HDF access file id

Data set setting				
The data of float64 Type is set specifying the HDF access label. - Memory area for storing data, the user using the function to ensure.				
INTEGER SGKTK_set_dataset_float64(HDF_file_id, label, kind, dims, rank, value)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label	CHARACTER	input	N	Data set access label
kind	INTEGER	input	1	Dimension information
dims	INTEGER	input	1	Number of dimension
rank	INTEGER	input	1	Buffer address storing data
value	REAL	input	8	HDF access file id

Acquisition of latitude and longitude data

It gets the latitude and longitude data in the range of the specified scan number

- Scan number is specified and end scan number start scan number.
Exit scan number, if it is less than the start scan number, an error is returned.
If greater than the scan numbers ending scan number is stored in the specified HDF data set is, I get to data that is stored in the HDF data set.
- Memory area for storing data, the user using the function to ensure.

INTEGER SGKTK_set_dataset_latlon(HDF_file_id, label_lat, label_lon, kind, start_no, end_no, set_num, latlon)

Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:The number of getting scan. Error:A negative value returns when it is not possible to acquire it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
label_lat	CHARACTER	input	N	Data set access label (latitude)
label_lon	CHARACTER	input	N	Data set access label (longitude)
kind	INTEGER	input	1	Brightness / reflectance / other type
start_no	INTEGER	input	1	Acquisition start the scan number
end_no	INTEGER	input	1	Acquisition end the scan number
set_num	INTEGER	input	1	Number of elements (the number of number × longitude of latitude)
latlon	REAL	input	1	Memory area for storing data

5.2.4 Other function

Group making				
Empty Group is made.				
INTEGER SGKTK_make_group(HDF_file_id, group)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal: 0 Error: A negative value returns when it is not possible to set it.
parameter				
HDF_file_id	INTEGER	input	1	HDF access file id
group	CHARACTER	input	N	Group name

Time converting (UTC to TAI93)				
The time is converted UTC to TAI93.				
INTEGER SGKTK_utc_to_tai93(year, month, day, hour, min, sec, msec, tai93time)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal:0 Error: A negative value returns when it is not possible to set it.
parameter				
year	INTEGER	input	1	Year
month	INTEGER	input	1	Month
day	INTEGER	input	1	Day
hour	INTEGER	input	1	Time
min	INTEGER	input	1	Minute
sec	INTEGER	input	1	Second
msec	INTEGER	input	1	Millisecond
tai93time	REAL	output	8	TAI93 time

Time converting (TAI93 to UTC)				
The time is converted TAI93 to UTC.				
INTEGER SGK_tai93_to_utc(tai93time, year, month, day, hour, min, sec, msec)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	Output	1	Normal: 0 Error: A negative value returns when it is not possible to set it.
parameter				
tai93time	REAL	Input	8	TAI93 time
year	INTEGER	output	1	Year
month	INTEGER	output	1	Month
day	INTEGER	output	1	Day
hour	INTEGER	output	1	Time
min	INTEGER	output	1	Minute
sec	INTEGER	output	1	Second
msec	INTEGER	output	1	Millisecond

Acquisition of latitude and longitude information				
Get the latitude and longitude from the specified granule ID, and stored in the buffer				
INTEGER SGK_get_global_latlon(HDF_file_id, latlon)				
Name	Type	IN/OUT	Size	Descriptions
return value				
data_num	INTEGER	output	1	Normal: Acquisition requirement number (latitude number×longitude number) Error: A negative value returns when it is not possible to set it.
parameter				
HDF_file_id	INTEGER	Input	1	HDF access file id
latlon	REAL	output	4	latitude and longitude information

Configuration file reflect				
The contents of the specified SGK configuration file is reflected.				
INTEGER SGK_load_config(file_name)				
Name	Type	IN/OUT	Size	Descriptions
return value				
status	INTEGER	output	1	Normal: 0 Error: A negative value returns when it is not possible to acquire it.
parameter				
file_name	CHARACTER	input	N	SGTK setting file path

6 File Definition

6.1 Configuration File

This is the configuration file to define the SGTK operation. The default configuration file is attached to the SGTK, but users can change the content of the file optionally. When you change the content of the configuration file, you need to use “Configuration file reflect” function in your source code.

Format is a text file, character code: ASCII, line feed code: UNIX (LF).

No.	Key	Contents	Value
1	DATATYPE_COMPRESS_TYPE	Compression type of product	Blank or gzip. Default: gzip
2	DATATYPE_COMPRESS_LEVEL	Compression level of product	Integer value: level 1 to 9. It becomes effective when compression type is not blank. Default: 4
3	DATATYPE_BYTE_ORDER	Byte order of product	LE or BE. Default: LE
4	DATATYPE_CHUNK	Chunk size of product	One or more integer value. Default: 256
5	LEAP_SECOND_FILE_PATH	The path of the leap second file	Default: ../../params/leapsec.dat (Describe the relative path from the sample program)
6	CORRECT_SLOPE_ATTRIBUTE_NAME	HDF5 Attribute name indicating the Slope of correction function by metadata.	If there are multiple names, they are separated by ",". Default: Slope
7	CORRECT_OFFSET_ATTRIBUTE_NAME	HDF5 Attribute name indicating the Offset of correction function by metadata.	If there are multiple names, they are separated by ",". Default: "Offset
8	CORRECT_BASE_ATTRIBUTE_NAME	HDF5 Attribute name indicating the Base of correction function by metadata.	If there are multiple names, they are separated by ",". Default: Base
9	CORRECT_SLOPE_REF_ATTRIBUTE_NAME	HDF5 Attribute name indicating the Slope of correction function by metadata.	If there are multiple names, they are separated by ",". Default: Slope_reflectance
10	CORRECT_OFFSET_REF_ATTRIBUTE_NAME	HDF5 Attribute name indicating the Offset of correction function by metadata.	If there are multiple names, they are separated by ",". Default: Offset_reflectance
11	RANGE_MIN_ATTRIBUTE_NAME	HDF5 Attribute name that indicates the minimum value of the range check function by metadata.	If there are multiple names, they are separated by ",". Default: Minimum_valid_DN
12	RANGE_MAX_ATTRIBUTE_NAME	HDF5 Attribute name that indicates the maximum value of the range checking function by metadata.	If there are multiple names, they are separated by ",". Default: Maximum_valid_DN
13	RANGE_ERROR_VALUE	HDF5 Attribute name indicating the abnormal values in the range check function by metadata.	If there are multiple names, they are separated by ",". Default: Error_DN

6.2 Leap Second File

This is the leap second file that defines the TAI-UTC released from IERS (International Earth Rotation and Reference Systems Service). The path of the leap second file is defined in the configuration file. (Refer to section 6.1)

(1) Providing destination : IERS (<ftp://maia.usno.navy.mil/ser7/leapsec.dat>)

The above URI is as of June 2013.

(2) Sample

```
Checked(unchanged): 2013-06-04T22:26:05Z using USNO tai-utc.dat file of Jun  4  18:26
1961 JAN  1 =JD 2437300.5  TAI-UTC=  1.4228180 S + (MJD - 37300.) X 0.0012960 S  ACTUAL
1961 AUG  1 =JD 2437512.5  TAI-UTC=  1.3728180 S + (MJD - 37300.) X 0.0012960 S  ACTUAL
...
2006 JAN  1 =JD 2453736.5  TAI-UTC= 33.0000000 S + (MJD - 41317.) X 0.0000000 S  ACTUAL
2009 JAN  1 =JD 2454832.5  TAI-UTC= 34.0000000 S + (MJD - 41317.) X 0.0000000 S  ACTUAL
2012 JUL  1 =JD 2456109.5  TAI-UTC= 35.0000000 S + (MJD - 41317.) X 0.0000000 S  ACTUAL
```

(3) Format

- File format: Text
- Encoding: ASCII
- Return code: LF (UNIX)
- Header (The first line):

```
Checked(unchanged): 2013-06-04T22:26:05Z using USNO tai-utc.dat file of Jun  4  18:26
```

- Data (The second and subsequent lines):

```
2009 JAN  1 =JD 2454832.5  TAI-UTC=34.0000000 S + (MJD - 41317.) X 0.0000000 S  ACTUAL
           *1                *2                *3
```

(4) How to use

- As the header section is an updated information, you do not need to refer it. “*n” of the data section is used.
- In case of before 1971, TAI-UTC (*2 + *3) is inserted after JD (*1)
If finding the leap second before 1971, it is necessary to add (*3).
 - After JD:2437300.5 (UTC:1961-01-01T00:00:00) => Leap second:1.422818
 - After JD:2437512.5 (UTC:1961-08-01T00:00:00) => Leap second:1.69757
- In case of after 1972, TAI-UTC (*2) is inserted after JD (*1).
If finding the leap second after 1972, (*3) is not be used.
 - After JD:2454832.5 (UTC:2009-01-01T00:00:00) => Leap second:34
 - After JD:2456109.5 (UTC:2012-07-01T00:00:00) => Leap second:35

6.3 Access Label Parameter File

Access label parameter file is the parameter file that defines the data set access label (Refer to section 7.1.1). Users use this data set access label to access to group/ data set/ attribute of product. The parameter file is created for each product file. When you read the SGLI product, you need to set access label parameter file that coincides with the product using “Read processing of parameter file” function in your source code (Refer to section 5.1.1 or 5.2.1) .

Description of the parameter file is written the format below.

“Data set access label” = “HDF path”

Format sample of the parameter file is shown in Figure 6-1.

```
↓
/** Group Access Labels.↓
*/↓
GC1_GRP_RESERVED=/Reserved↓
GC1_GRP_RAW_DATA_AUX_PACKET=/Raw_data/AUX_packet↓
GC1_GRP_RAW_DATA=/Raw_data↓
↓
/** Dataset Access Labels.↓
*/↓
GC1_DTS_RESERVED_UNUSED_PACKET_RIGHT=/Reserved/Unused_packet_right↓
GC1_DTS_RESERVED_UNUSED_PACKET_NADIR=/Reserved/Unused_packet_nadir↓
GC1_DTS_RESERVED_UNUSED_PACKET_LEFT=/Reserved/Unused_packet_left↓
GC1_DTS_RAW_DATA_VN11=/Raw_data/VN11↓
GC1_DTS_RAW_DATA_VN10=/Raw_data/VN10↓
GC1_DTS_RAW_DATA_VN09=/Raw_data/VN09↓
GC1_DTS_RAW_DATA_VN08=/Raw_data/VN08↓
GC1_DTS_RAW_DATA_VN07=/Raw_data/VN07↓
GC1_DTS_RAW_DATA_VN06=/Raw_data/VN06↓
GC1_DTS_RAW_DATA_VN05=/Raw_data/VN05↓
GC1_DTS_RAW_DATA_VN04=/Raw_data/VN04↓
GC1_DTS_RAW_DATA_VN03=/Raw_data/VN03↓
GC1_DTS_RAW_DATA_VN02=/Raw_data/VN02↓
GC1_DTS_RAW_DATA_VN01=/Raw_data/VN01↓
↓
/** Attribute Access Labels.↓
*/↓
GC1_ATR_PRODUCT_FILE_NAME=Product_file_name↓
GC1_ATR_WORST_ORBIT_SOURCE_DATA_DESCRIPTION=Worst_orbit_source_data_description↓
GC1_ATR_WORST_ORBIT_SOURCE=Worst_orbit_source↓
GC1_ATR_WORST_ATTITUDE_SOURCE_DATA_DESCRIPTION=Worst_attitude_source_data_description↓
GC1_ATR_WORST_ATTITUDE_SOURCE=Worst_attitude_source↓
GC1_ATR_UPPER_RIGHT_LONGITUDE=Upper_right_longitude↓
GC1_ATR_UPPER_RIGHT_LATITUDE=Upper_right_latitude↓
GC1_ATR_UPPER_LEFT_LONGITUDE=Upper_left_longitude↓
GC1_ATR_UPPER_LEFT_LATITUDE=Upper_left_latitude↓
[EOF]
```

Figure 6-1 Format of Access Label Parameter File

Access label parameter file is stored in the following directories.

SGTK_SGLI¥sample¥bin¥L1: Access label parameter file for L1

SGTK_SGLI¥sample¥bin¥L2: Access label parameter file for L2

SGTK_SGLI¥sample¥bin¥L3: Access label parameter file for L3

7 In/Out-Put Data

7.1 Data Definition

7.1.1 Dataset Access Label

SGTK executes input/output to the products of SGLI level 1 to 3.

Input/output function of group, dataset, attribute are common to level 1 to 3. SGTK is designed to be unaware of the difference between them. All products are regarded as equivalent in terms of group, dataset, and attribute.

Therefore, when group, dataset, and attribute are input/output in user program, the dataset access label to identify the each element of Level 1 to 3 is specified to input/output function.

However, since this is arbitrarily specified in the user program, SGTK does not execute the processing for each level.

If the specification of group, dataset, and attribute for SGLI product is changed, what users should do is only change the dataset access label.

This label is described in the access label parameter file (Refer to Section 6.3) as key and value.

Even though default configuration file is attached to SGTK, users can change the content of the file optionally.

8 Error Number ID

Error number ID of SGK functions is listed in Table 8-1.

All of ID are negative value and given the number in the following categories.

- 100 to Execution environment error
- 200 to Error due to the user program's bug.
- 500 to Error due to the HDF file
- 600 to Error due to the setting data

Table 8-1 Error Number ID (1/3)

ID	Description	Remark
-100	Fail to open the file, such as leap second file.	
-101	Fail to allocate memory	
-102	Fail to get the environment variable	
-103	Fail to get leap second	
-109	Fail to get geophysical quantity file. File does not exist, or the definition of the file is not correct.	
-201	Specified value is invalid (year of scan time).	After 1993
-202	Specified value is invalid (month of scan time).	Valid data: 1-12
-203	Specified value is invalid (day of scan time).	Valid data: 1-31
-204	Specified value is invalid (hour of scan time).	Valid data: 1-24
-205	Specified value is invalid (minute of scan time).	Valid data: 0-59
-206	Specified value is invalid (second of scan time).	Valid data: 0-60
-207	Specified value is invalid (millisecond of scan time).	Valid data: 0-999
-208	Specified value is invalid (scan time).	
-209	Specified value is invalid (elapsed second time).	
-210	No file name	
-211	The value of the start scan number is bigger than the end scan number.	
-213	HDF file ID is 0 or minus value.	
-214	The index value of meta data is minus value.	
-215	Specified dimension size is invalid.	
-216	No specified writing data	
-217	Access label is 0.	
-218	Specify the invalid value. (Access label is not for writing)	
-219	Specify the invalid value. (Access label is not for reading)	
-220	Data size is zero (0) when checking the memory area.	
-221	Data address specified by parameter is NULL.	
-238	Writing mode for HDF file is invalid.	
-239	Function data type corresponding to the access label is not correct.	
-501	Fail to open HDF file.	
-502	Fail to close HF file.	
-503	Fail to set dimension size.	
-504	Fail to access to data set.	
-505	Fail to open data set.	
-506	Fail to get the space for data set.	
-507	Fail to get the number of array for data set.	
-508	Fail to get the dimension size for data set.	
-509	Fail to create data set.	
-510	Fail to write to data set.	
-511	Fail to read from data set.	
-520	Fail to open metadata name.	
-521	Fail to get the data type of metadata name.	
-522	Fail to read the value of metadata.	
-523	Fail to open with specified name of metadata.	
-524	Fail to create the name of metadata.	

Table 8-1 Error Number ID (2/3)

ID	Description	Remark
-525	Fail to write the value of metadata.	
-526	Fail to get the class of data type.	
-527	Fail to copy the class of data type.	
-528	Fail to modify the size of data type.	
-529	Fail to get data type.	
-540	Fail to get the space of data type.	
-541	Fail to specify the point to read/write for data set.	
-550	Fail to access data set due to the opening error of metadata "ProductName".	Refer to 7.1.1 Dataset Access Label
-551	Fail to access data set because the internal error occurred in the HDF library when reading metadata "ProductName".	Refer to 7.1.1 Dataset Access Label
-552	Fail to access data set due to the opening error of meta data "OverlapScans".	Refer to 7.1.1 Dataset Access Label
-553	Fail to access data set because the internal error occurred in the HDF library when reading metadata "ProductScans".	Refer to 7.1.1 Dataset Access Label
-554	Fail to access data set due to the opening error of metadata "GeophysicalName".	Refer to 7.1.1 Dataset Access Label
-555	Fail to access data set because the internal error occurred in the HDF library when reading metadata "GeophysicalName ".	Reference: 7.1.1 Dataset Access Label
-556	Fail to access data set due to the opening error of metadata "GranuleID"	Refer to 7.1.1 Dataset Access Label
-557	Fail to access data set because the internal error occurred in the HDF library when reading metadata "GranuleID ".	Refer to 7.1.1 Dataset Access Label
-558	Fail to access data set due to the opening error of metadata "CoRegistrationParameterA1"	Refer to 7.1.1 Dataset Access Label
-559	Fail to access data set because the internal error occurred in the HDF library when reading metadata "CoRegistrationParameterA1 ".	Refer to 7.1.1 Dataset Access Label
-560	Fail to access data set due to the opening error of metadata "CoRegistrationParameterA2"	Refer to 7.1.1 Dataset Access Label
-561	Fail to access data set because the internal error occurred in the HDF library when reading metadata "CoRegistrationParameterA2 ".	Refer to 7.1.1 Dataset Access Label
-562	Fail to access data set due to the opening error of metadata "Projection"	Refer to 7.1.1 Dataset Access Label
-563	Fail to access data set because the internal error occurred in the HDF library when reading metadata "Projection ".	Refer to 7.1.1 Dataset Access Label
-564	Fail to access data set due to the opening error of metadata "OrbitDirection".	Refer to 7.1.1 Dataset Access Label
-565	Fail to access data set because the internal error occurred in the HDF library when reading metadata "OrbitDirection ".	Refer to 7.1.1 Dataset Access Label
-566	Fail to access data set due to the opening error of metadata "ProductionDateTime"	Refer to 7.1.1 Dataset Access Label
-567	Fail to access data because the internal error occurred in the HDF library when reading metadata "ProductionDateTime ".	Refer to 7.1.1 Dataset Access Label
-601	Data type for reading/writing is not the same with the	

Table 8-1 Error Number ID (3/3)

ID	Description	Remark
	data type of data set.	
-602	The maximum and minimum value for data writing is invalid.	
9999	Writing data exceeds the maximum value.	

9 Appendix

9.1 Sample Program (sample_01)

9.1.1 Linux

This is the sample program using the following installation directory.
If using another directory, change makefile or the paths of the following commands.

HDF5 installation directory: /HDF5/shared

SGTK installation directory: /SGTK_SGLI

* For library sgtk, it needs to use the library created in the same environment.

```
$ cd /SGTK_SGLI/sample/linux      Move directory to create the sample for Linux.
$ make                             make
$ cd ../bin                       Move to the directory in which execution program is stored.
```

↓Library path is set to environmental variable

```
$ export LD_LIBRARY_PATH=/HDF5/shared/lib:/SGTK_SGLI/lib:$LD_LIBRARY_PATH
$ sample_01                       Execute the sample program.
SGTK_make_parameter_table: make parameter table
pata_ret=0
map_size=408
SGTK_load_config: load configuration file
SGTK_open_file: open product file
SGTK_set_dims: set dimensions
SGTK_malloc: allocate memory to set data
SGTK_set_dataset_int: set dataset value
SGTK_set_meta_string: set string attribute value
SGTK_set_meta_float: set numeric attribute value
SGTK_malloc: allocate memory to get data
SGTK_get_dataset_int: get dataset value
SGTK_get_meta_string: get string attribute value
SGTK_get_meta_float: get numeric attribute value
SGTK_close_file: close product file
SGTK_free: release alloc'd memory
sample program normal end

$
```

9.1.2 Windows

Double click “C:¥SGTK_SGLI¥sample¥msvc¥sample_01.sln” to build “VisualStudio2015”. Built program is copied to the following directory, and execute it.

```
C:¥SGTK_SGLI¥sample¥bin>
```

```
C:¥SGTK_SGLI¥sample¥bin> Specify the library directory by the following command.  
set PATH=%PATH%;C:¥CMake-hdf5-1.8.21¥build¥bin¥Release;C:¥SGTK_SGLI¥lib
```

```
C:¥SGTK_SGLI¥sample¥bin> sample_01
```

```
SGTK_make_parameter_table : make parameter table
```

```
pata_ret=0
```

```
map_size=408
```

```
SGTK_load_config : load configuration file
```

```
SGTK_open_file : open product file
```

```
SGTK_set_dims : set dimensions
```

```
SGTK_malloc : allocate memory to set data
```

```
SGTK_set_dataset_int : set dataset value
```

```
SGTK_set_meta_string : set string attribute value
```

```
SGTK_set_meta_float : set numeric attribute value
```

```
SGTK_malloc : allocate memory to get data
```

```
SGTK_get_dataset_int : get dataset value
```

```
SGTK_get_meta_string : get string attribute value
```

```
SGTK_get_meta_float : get numeric attribute value
```

```
SGTK_close_file : close product file
```

```
SGTK_free : release alloc'd memory
```

```
sample program normal end
```

If installing HDF5 and SGK to the different directory from this sample, change the property of “inlude” directory, library directory of sample_01 in VisualStudio2015.