GST-190020

GOSAT-1/2 Level 1 product reading toolkit

User's manual (C Language)

Aug, 2019

Japan Aerospace Exploration Agency

Changing history

Version	Date	Page	Changes
	2019/08	—	

Table of Contents

1. Introduction

This document is the user’s manual for GOSAT Level 1 product reading Toolkit (GTK). GTK is a programming library to read GOSAT-1/GOSAT-2 Level 1 product. This document describes about C language version.

1. 1 Supported product formats

GTK supports product formats described in Table 1.1-1.

Table 1.1 -1 Supported product format list

No	Product file	Version.
1	TANSO-FTS Level 1A product	V2XX
2	TANSO-FTS Level 1B product	V2XX
3	TANSO-CAI Level 1A product	V130
4	TANSO-FTS-2 Level 1A product	V1XX
5	TANSO-FTS-2 Level 1B product	V1XX
6	TANSO-CAI-2 Level 1A product	V1XX

Following document describes for more detail about the format.

- GOSAT / TANSO Level 1 Product Description Document TANSO-FTS Section
- Level 1 Data Format Description Document TANSO-CAI Version
- GOSAT-2 Mission Operations System_Level 1 Data Format Description Document (TANSO-FTS-2 Version)
- GOSAT-2 Mission Operations System_Level 1 Data Format Description Document (TANSO-CAI-2 Version)

1. 2 List of Application Programming Interface (API)

Table 1.2-1 shows the Application Programing Interface (API) supported by GTK.

Table 1.2-1 Application Programing Interface

No	API	Description
1.	gtk_com_hdfopen	Open GOSAT-1/2 product file.
2.	gtk_com_hdfclose	Close GOSAT-1/2 product file.
3.	gtk_com_dimsize	Get dimension size of dataset
4.	gtk_com_readi1s	Read 1byte signed integer data.
5.	gtk_com_readi1u	Read 1byte unsigned integer data.
6.	gtk_com_readi2s	Read 2bytes signed integer data.
7.	gtk_com_readi2u	Read 2bytes unsigned integer data.
8.	gtk_com_readi4s	Read 4bytes signed integer data.
9.	gtk_com_readi4u	Read 4bytes unsigned integer data.
10.	gtk_com_readr4	Read 4bytes floating point data.
11.	gtk_com_readr8	Read 8bytes floating point data.
12.	gtk_com_readchar	Read string data.
13.	gtk_com_readstruct	Read GOSAT-1 time structure data.
14.	gtk_com_dump	Dump dataset
15.	gtk_fts1_getigm	Get FTS/L1A interferogram
16.	gtk_fts1_getspc	Get FTS/L1B spectrum
17.	gtk_fts1_rad2bt	Get FTS/TIR brightness temperature
18.	gtk_fts1_readcam	Read FTS/CAM data and export to JPEG file.
19.	gtk_fts2_getigm	Get FTS-2/L1A interferogram
20.	gtk_fts2_getspc	Get FTS-2/L1B spectrum
21.	gtk_fts2_rad2bt	Convert FTS-2/TIR radiance to brightness temperature
22.	gtk_fts2_readcam	Read FTS-2/CAM data and export to JPEG file.
23.	gtk_cai1_geointerpol	Get CAI latitude/longitude (by interpolation)
24.	gtk_cai1_dn2rad	Convert CAI observation data to radiance
25.	gtk_cai1_cutimage	Get CAI data about specified region
26.	gtk_cai2_geointerpol	Get CAI-2 latitude/longitude (by interpolation)
27.	gtk_cai2_dn2rad	Convert CAI-2 observation data to radiance
28.	gtk_cai2_cutimage	Get CAI-2 data about specified region
29.	gtk_cai2_geocalc	Calculate CAI-2 latitude/longitude
30.	gtk_cai2_resampimage	Get CAI-2 resampled image by band-to-band registration
31.	gtk_cai2_create_sensor_param	Create CAI-2 sensor parameter
32.	gtk_cai2_delete_sensor_param	Delete CAI-2 sensor parameter
33.	gtk_cai2_create_prcessed_image	Create CAI-2 processed image
34.	gtk_cai2_delete_image	Delete CAI-2 image
35.	gtk_cai2_create_resample_info	Create information instance for CAI-2 band-to-band registration
36.	gtk_cai2_get_resample_position	Get pixel number/line number corresponding to a specified position on the base band.
36.	gtk_cai2_delete_resample_position	Delete information instance for CAI-2 band-to-band registration

2. System requirements

Table 2-1 shows system requirements.

Table 2-1 System requirements

Environment	Requirements
RAM	8GB or more
OS	Linux, Windows10(Cygwin), macOS
C compiler	support C99 (ISO/IEC 9899:1999)
Required software	HDF5 library (version 1.8) JPEG library (version 9) TIFF library (version 4) OpenCV library (version 2.4)

All of above development environment includes C compiler and required software must be ready for user's development environment.

Table 2-2 shows the system environment this software was tested.

Table 2-2 Test environment

OS	Hardware	Software
Linux	OS: Linux RHEL7.2 (Kernel 3.10.0-327.el7.x86_64) CPU: Intel(R) Xeon(R) CPU E3-1271 v3 @ 3.60GHz RAM: 16 GB	HDF5-1.8.18 libjpeg 9.0 libtiff 4.0.9 opencv 2.4.13 gcc4.8.3
Mac	OS: macOS X10.11.4 CPU: Intel(R) Core(TM) i5-4570R CPU @ 2.70GHz RAM: 8 GB	HDF5-1.8.16 libjpeg 9.0 libtiff 4.0.9 opencv 2.4.13 gcc5.3
Windows	OS : Windows10 (cygwin2.5.1-1) CPU : Intel(R) Xeon(R) CPU E3-1226 V3 @3.30GHz RAM : 32 GB	HDF5-1.8.16 libjpeg 9.0 libtiff 4.0.9 opencv 2.4.13 gcc4.9.3

3. Install

The GTK source files are archived in tar.gz file. Download the tar.gz file from download site and type following command to extract files.

```
$ tar zxvf gosat_tk-C-version.tar.gz
```

Table 3-1 shows the contents in archived file.

Table 3-1 Contents

Contents	Description
VERSION	Description about version
src/	Library source code
include/	Header files
sample/	Source code of sample program

Enter to following folder and type “make” command to build library.

```
$ cd gosat_tk-C-version/
$ make
```

The toolkit library(“libgosat_tk.a”) will be built in “src” folder.

User can change one or more required software path by following command.

```
$ make HDF5_DIR=”path” JPEG_DIR=”path” ZLIB_DIR=”path”
TIFF_DIR=”path” OPENCV_DIR=”path”
```

Type the following command to install library on the user’s development environment.

```
$ make install
```

By default, the files will be installed in /usr/local/include and /usr/local/lib.
User can change install prefix by following command.

```
$ make install PREFIX=”path”
```

The files will be installed in *path*/include and *path*/lib.

4. Parameter files

The APIs for CAI/CAI-2 requires the parameter files to configure its settings. Refer to the following document for more detail about parameter.

“GOSAT-1/2 Level 1 product reading toolkit user’s manual (Parameter file)”

User can download the archived parameter files from download site. Type the following command to extract files.

```
$ tar zxvf gosat_tk-parameter-version.tar.gz
```

Table 4-1 shows the directory structure after extracting parameter files.

Table 4-1 Directory structure of parameter file

Directory	DESCRIPTIONS
parameter/	Contains parameter files.

5. Sample programs

The source archive includes sample program for testing function quickly.

Table 5-1 shows the list of sample program.

Table 5-1 List of sample program

No.	Program	Description
1.	sample_com	Open GOSAT-1/2 product file and read data from dataset.
2.	sample_fts1_getigm	Get FTS/L1A interferogram.
3.	sample_fts1_getspc	Get FTS/L1B spectrum.
4.	sample_fts1_rad2bt	Get FTS/L1B brightness temperature.
5.	sample_fts1_readcam	Read FTS/CAM data and export to JPEG file.
6.	sample_fts2_getigm	Get FTS-2/L1A interferogram.
7.	sample_fts2_getspc	Get FTS-2/L1B spectrum.
8.	sample_fts2_rad2bt	Get FTS-2/L1B(TIR) brightness temperature.
9.	sample_fts2_readcam	Read FTS-2/CAM Data and export to JPEG file.
10.	sample_cai1_geointerpol	Get CAI latitude/longitude.
11.	sample_cai1_dn2rad	Convert CAI observation data to radiance.
12.	sample_cai1_cutimage	Get CAI-2 data about specified region.
13.	sample_cai2_geointerpol	Get CAI-2 latitude/longitude.
14.	sample_cai2_dn2rad	Convert CAI-2 observation data to radiance.
15.	sample_cai2_cutimage	Get CAI-2 data about specified region.
16.	sample_cai2_geocalc	Get CAI-2 latitude/longitude
17.	sample_cai2_resampimage	Get CAI-2 resampled image by band-to-band registration
18.	sample_cai2_create_processed_image	Create CAI-2 L1A processed image.
19.	sample_cai2_get_resample_position	Get CAI-2 resampled position by band-to-band registration

To build sample program, enter “sample” directory and type following command.

```
$ cd sample
$ make
```

Executable program files will be built in sample directory.

Sample programs require the specific test product for reading test. Download test product from download site and type following command to extract files.

```
$ tar zxvf gosat_tk-data-version.tar.gz
```

Table 5-2 shows the directory structure to run sample program. Move the product files and the parameter files like Table 5-2.

Table 5-2 Directory structure to run sample program

Contents	Description
sample/	Contains sample programs
parameter/	Contains parameter files
data/	Contains product files

Move to sample directory and run executables listed in Table 5-1 like below.

```
$./sample_com
```

6. API Specification

6．1 Handling HDF files

6．1．1 gtk_com_hdfopen

NAME

gtk_com_hdfopen – Open GOSAT-1/2 product file

SYNTAX

```
#include "gosat_tk.h"

 hid_t gtk_com_hdfopen(const char* file);
```

PARAMETERS

Parameter	I/O	Description
file	in	file name of GOSAT-1/2 product hdf5

RETURN VALUE

Return value	Description
0≦	valid hdf5 file identifier
Negative number	hdf5 open failed or the files is not supported.

NOTE: hid_t is equal to signed integer.

DESCRIPTIONS

gtk_com_hdfopen opens GOSAT-1/2 product file (hdf5) and read “Metadata” dataset.

If hdf5 file is GOSAT-1/2 product, the gtk_com_hdfopen returns with a valid hdf5 file identifier, otherwise a negative number.

NOTE

Application needs to call gtk_com_hdfclose to close file.

SAMPLES

sample_com

6．1．2　　gtk_com_hdfclose

NAME

gtk_com_hdfclose – Close GOSAT-1/2 product file

SYNTAX

```
#include "gosat_tk.h"

GTK_RET gtk_com_hdfclose(const hid_t file_id);
```

PARAMETERS

Parameter	I/O	Description
file	in	hdf5 file identifier

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_com_hdfclose closes the specified file.

SAMPLES

sample_com

6. 1. 3 gtk_com_dimsize

NAME

gtk_com_dimsize – Get dimension size of dataset

SYNTAX

```
#include "gosat_tk.h"

GTK_RET gtk_com_dimsize(int* ndims, hsize_t dims[H5S_MAX_RANK],
    const hid_t file_id, const char* path);
```

PARAMETERS

Parameter	I/O	Description
ndims	out	Pointer to the location in which the number of dimensions to be returned.
dims	out	Pointer to the location in which element size of dimensions to be returned.
file_id	in	hdf5 file identifier
path	in	Dataset path　for dimension. (example :"/Metadata/granuleID")

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_com_dimsize reads the dimension size of the specified dataset.

SAMPLES

sample_com

6. 1. 4 gtk_com_read

NAME

gtk_com_read – Read the specified dataset

SYNTAX

```
#include "gosat_tk.h"
```

- Signed 1byte integer

```
GTK_RET gtk_com_readi1s(int* ndims, hsize_t dims[H5S_MAX_RANK],
                  signed char** data,   const hid_t file_id, const char* path);
```

- Unsigned 1byte integer

```
GTK_RET gtk_com_readi1u(int* ndims, hsize_t dims[H5S_MAX_RANK],
            unsigned char** data, const hid_t file_id, const char* path);
```

- Signed 2bytes integer

```
GTK_RET gtk_com_readi2s(int* ndims, hsize_t dims[H5S_MAX_RANK],
            short** data,   const hid_t file_id, const char* path);
```

- Unsigned 2bytes integer

```
GTK_RET gtk_com_readi2u(int* ndims, hsize_t dims[H5S_MAX_RANK],
          unsigned short** data, const hid_t file_id, const char* path);
```

- Signed 4bytes integer

```
GTK_RET gtk_com_readi4s(int* ndims, hsize_t dims[H5S_MAX_RANK],
            int** data, const hid_t file_id, const char* path);
```

- Unsigned 4bytes integer

```
GTK_RET gtk_com_readi4u(int* ndims, hsize_t dims[H5S_MAX_RANK],
          unsigned int** data, const hid_t file_id, const char* path);
```

- 4bytes floating point data

```
GTK_RET gtk_com_readr4(int* ndims, hsize_t dims[H5S_MAX_RANK],
          float** data, const hid_t file_id, const char* path);
```

- 8bytes floating point data

```
GTK_RET gtk_com_readr8(int* ndims, hsize_t dims[H5S_MAX_RANK],
          double** data, const hid_t file_id, const char* path);
```

- String

```
GTK_RET gtk_com_readchar(int* ndims, hsize_t dims[H5S_MAX_RANK],
          int *slength, char** str, const hid_t file_id, const char* path);
```

PARAMETERS

Parameter	I/O	Description
ndims	out	Pointer to the location in which the number of dimensions to be returned.
dims	out	Pointer to the location in which element size of dimensions to be returned.
data	out	Pointer to the location in which obtained data to be returned.
slength	out	Pointer to the location in which the string length to be returned.
str	out	Pointer to the location in which obtained string data to be returned.
file_id	in	hdf5 file identifier
path	in	Dataset path (example :"/Metadata/granuleID")

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_com_read reads the specified dataset. The obtained data will be stored into the new memory address *data which the API allocated.

If the dataset is multidimensional, the obtained data is stored as follows.

Dimension of dataset	Expression of multidimensional data in L1 data format specification	Reference to data
2-Dimention	y,x	(*data)[y*M+x] M: dims[1]
3-Dimention	z,y,x	(*data)[z*(M*N)+y*N+x] M : dims[2] N : dims[1]

NOTE

Application needs to call free() to release memory *data.

SAMPLES

sample_com

6. 1. 5 gtk_com_dump

NAME

gtk_com_dump – Dump dataset to a file

SYNTAX

```
#include "gosat_tk.h"

 GTK_RET gtk_com_dump(int* ndims, hsize_t dims[H5S_MAX_RANK],
                  const hid_t file_id, const char* path, const char* fout,
                  const GTK_FORMAT format);
```

PARAMETERS

Parameter	I/O	Description
ndims	out	Pointer to the location in which the number of dimensions to be returned.
dims	out	Pointer to the location in which element size of dimensions to be returned.
file_id	in	hdf5 file identifier
path	in	Dataset path (example :"/Metadata/granuleID")
fout	in	Output file name
format	in	GTK_FORMAT_BIN : binary mode GTK_FORMAT_TXT : text mode

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_com_dump outputs dataset to the specified file. The text or binary format are available.

SAMPLES

sample_com

6. 2 GOSAT-1 FTS Level 1 product API

6. 2. 1 gtk_fts1_getigm

NAME

gtk_fts1_getigm – Get interferogram

SYNTAX

```
#include "gosat_tk.h"

int gtk_fts1_getigm(int* ndata, float** datax, unsigned short** datay,
         const hid_t file_id, GTK_FLAG_IN flag_in,
         int num, const double lat, const double lon,
         const int band, FTS_FLAG_PS ps);
```

PARAMETERS

Parameter	I/O	Description
ndata	out	Pointer to the location in which the number of samples to be returned.
datax	out	Pointer to the location in which the optical path difference [cm] for each sample to be returned.
datay	out	Pointer to the location in which the interferogram [V] to be returned.
file_id	in	hdf5 file id of GOSAT-1 FTS L1A product file.
flag_in	in	OBSNO ：The sounding is specified by index in the file. LATLON：The sounding is specified by latitude and longitude.
num	in	The index of soundings to be read. The parameter is available only if flag_in is OBSNO. 1 ≦ num ≦ numPoints (*) (*) numPoints：Number of the soundings in the file.
lat, lon	in	The latitude and longitude to specify the sounding. [Degree] The parameter is available only if flag_in is LATLON. -90 ≦ lat ≦ 90 (degree) -180 ≦ lon ≦ 180 (degree)
band	in	The band number. 1 ≦ band ≦ 4
ps	in	Polarization. Band 1 to 3: P：P polarization mode S：S polarization mode Band 4： P

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_fts1_getigm reads interferogram of the specified sounding.

The sounding can be specified by either the index in the file or latitude/longitude. If the sounding is specified by latitude/longitude, the nearest sounding in file will be returned.

The obtained data will be stored into the new memory address *datax and *datay which the API allocated.

NOTE

Application needs to call free() to release memory *datax and *datay.

SAMPLES

sample_fts1_getigm

6. 2. 2 gtk_fts1_getspc

NAME

gtk_fts1_getspc – Get spectrum

SYNTAX

```
#include "gosat_tk.h"

int gtk_fts1_getspc(int* ndata, double** datax, float** datay, const hid_t file_id,
        GTK_FLAG_IN flag_in, int num, const double lat, const double lon, const int
    band, FTS_FLAG_PS ps);
```

PARAMETERS

Parameter	I/O	Description
ndata	out	Pointer to the location in which the number of samples to be returned.
datax	out	Pointer to the location in which the wave number for each sample to be returned. [cm-1]
datay	out	Pointer to the location in which the spectrum to be returned. [V/cm-1]
file_id	in	hdf5 file id of GOSAT-1 FTS L1B product file.
flag_in	in	OBSNO ：The sounding is specified by index in the file. LATLON：The sounding is specified by latitude and longitude.
num	in	The index of soundings to be read. The parameter is available only if flag_in is OBSNO. 1 ≦ num ≦ numPoints (*) (*) numPoints：Number of the soundings in the file.
lat, lon	in	The latitude and longitude to specify the sounding. [Degree] The parameter is available only if flag_in is LATLON. -90 ≦ lat ≦ 90 (degree) -180 ≦ lon ≦ 180 (degree)
band	in	The band number. 1 ≦ band ≦ 4
ps	in	Polarization. Band 1 to 3: P：P polarization mode S：S polarization mode Band 4： P

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_fts1_getspc reads spectrum data of the specified sounding.

The sounding can be specified by either the index in the file or latitude/longitude. If the sounding is specified by latitude/longitude, the nearest sounding in file will be returned.

The obtained data will be stored into the new memory address *datax and *datay which the API allocated.

NOTE

Application needs to call free() to release memory datax and datay.

SAMPLES

sample_fts1_getspc

6. 2. 3 gtk_fts1_rad2bt

NAME

gtk_fts1_rad2b - Convert FTS/TIR radiance to brightness temperature

SYNTAX

```
#include "gosat_tk.h"

int gtk_fts1_rad2bt(int* ndata, double** datax, double** datay, const hid_t file_id,
         GTK_FLAG_IN flag_in, int num, const double lat, const double lon);
```

PARAMETERS

Parameter	I/O	Description
ndata	out	Pointer to the location in which the number of samples to be returned.
datax	out	Pointer to the location in which the wave number [cm-1] for each sample to be returned.
datay	out	Pointer to the location in which the brightness temperature [K] to be returned.
file_id	in	hdf5 file id of GOSAT-1 FTS L1B product file.
flag_in	in	OBSNO : The sounding is specified by index in the file. LATLON : The sounding is specified by latitude and longitude
num	in	The index of soundings to be read. The parameter is available only if flag_in is OBSNO. 1 ≦ num ≦ numPoints (*) (*) numPoints : Number of the soundings in the file.
lat, lon	in	The latitude and longitude to specify the sounding. [Degree] The parameter is available only if flag_in is LATLON. -90 ≦ lat ≦ 90 (degree) -180 ≦ lon ≦ 180 (degree)

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_fts1_rad2bt calculates brightness temperature of the specified sounding.

The sounding can be specified by either the index in the file or latitude/longitude. If the sounding is specified by latitude/longitude, the nearest sounding in file will be returned.

The calculate data will be stored into the new memory address *datax and *datay which the API allocated.

NOTE

Application needs to call free() to release memory *datax and *datay.

SAMPLES

sample_fts1_rad2bt

6．2．4　gtk_fts1_readcam

NAME

gtk_fts1_readcam – Read CAM data and export to JPEG file

SYNTAX

```
#include "gosat_tk.h"

int gtk_fts1_readcam(const hid_t file_id, const int num);
```

PARAMETERS

Parameter	I/O	Description
file_id	in	hdf5 file id of GOSAT-1 FTS L1 product file.
num	in	The index of camera data in the file. 1 ≦ num ≦ numPoints(*) (*) numPoints: The number of camera data

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_fts1_readcam reads the specified camera data and export it to the JPEG file in current working directory. The JPEG file is saved as the following file name.

File name: yymmddHHMMSSXXXX.jpg

SAMPLES

sample_fts1_readcam

6．3　　GOSAT-2 FTS-2 Level 1 product API

6．3．1　　gtk_fts2_getigm

NAME

gtk_fts2_getigm – Get interferogram

SYNTAX

```
#include "gosat_tk.h"

int gtk_fts2_getigm(int* ndata, double** datax, float** datay, const hid_t file_id,
        GTK_FLAG_IN flag_in, int num, const double lat, const double lon, const int
    band, FTS_FLAG_PS ps);
```

PARAMETERS

Parameter	I/O	Description
ndata	out	Pointer to the location in which the number of samples to be returned.
datax	out	Pointer to the location in which the optical path difference [cm] for each sample to be returned.
datay	out	Pointer to the location in which the interferogram [V] to be returned.
file_id	in	hdf5 file id of GOSAT-2 FTS-2 L1A SWIR/TIR product file.
flag_in	in	OBSNO　: The sounding is specified by index in the file. LATLON : The sounding is specified by latitude and longitude.
num	in	The index of soundings to be read. The parameter is available only if flag_in is OBSNO. 1 ≦ num ≦ numPoints (*) (*) numPoints : Number of the soundings in the file.
lat, lon	in	The latitude and longitude to specify the sounding. [Degree] The parameter is available only if flag_in is LATLON. -90 ≦ lat ≦ 90 (degree) -180 ≦ lon ≦ 180 (degree)
band	in	The band number. 1 ≦ band ≦ 5
ps	in	Polarization. Band 1 to 3: P : P polarization mode S : S polarization mode Band 4,5 P

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_fts2_getigm reads interferogram of the specified sounding.

The sounding can be specified by either the index in the file or latitude/longitude. If the sounding is specified by latitude/longitude, the nearest sounding in file will be returned.

The obtained data will be stored into the new memory address *datax and *datay which the API allocated.

NOTE

Application needs to call free() to release memory datax and datay.

SAMPLES

sample_fts2_getigm

6．3．2　　gtk_fts2_getspc

NAME

gtk_fts2_getspc - Get spectrum

SYNTAX

```
#include "gosat_tk.h"

int gtk_fts2_getspc(int* ndata, double** datax, float** datay, const hid_t file_id,
        GTK_FLAG_IN flag_in, int num, const double lat,
        const double lon, const int band, FTS_FLAG_PS ps);
```

PARAMETERS

Parameter	I/O	Description
ndata	out	Pointer to the location in which the number of samples to be returned.
datax	out	Pointer to the location in which the wave number [cm-1] for each sample to be returned.
datay	out	Pointer to the location in which the spectrum [V/cm-1] be returned.
file_id	in	hdf5 file id of GOSAT-2 FTS-2 L1B SWIR/TIR product file.
flag_in	in	OBSNO　：The sounding is specified by index in the file. LATLON：The sounding is specified by latitude and longitude.
num	in	The index of soundings to be read. The parameter is available only if flag_in is OBSNO. 1 ≦ num ≦ numPoints (*) (*) numPoints：Number of the soundings in the file.
lat, lon	in	The latitude and longitude to specify the sounding. [Degree] The parameter is available only if flag_in is LATLON. -90 ≦ lat ≦ 90 (degree) -180 ≦ lon ≦ 180 (degree)
band	in	The band number. 1 ≦ band ≦ 5
ps	in	Polarization. Band 1 to 3: P：P polarization mode S：S polarization mode Band 4,5 P

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_fts2_getspc reads interferogram of the specified sounding.

The sounding can be specified by either the index in the file or latitude/longitude. If the sounding is specified by latitude/longitude, the nearest sounding in file will be returned.

The obtained data will be stored into the new memory address *datax and *datay which the API allocated.

NOTE

Application needs to call free() to release memory *datax and *datay by calling free() when no longer necessary.

SAMPLES

sample_fts2_getspc

6. 3. 3 gtk_fts2_rad2bt

NAME

gtk_fts2_rad2bt – Convert FTS-2/TIR radiance to brightness temperature

SYNTAX

```
#include "gosat_tk.h"

int gtk_fts2_rad2bt(int* ndata, double** datax, double** datay, const hid_t file_id,
          GTK_FLAG_IN flag_in, int num, const double lat, const double lon,
         const int band);
```

PARAMETERS

Parameter	I/O	Description
ndata	out	Pointer to the location in which the number of samples to be returned.
datax	out	Pointer to the location in which the wave number [cm-1] for each sample to be returned.
datay	out	Pointer to the location in which the brightness temperature [cm-1] to be returned.
file_id	in	hdf5 file id of GOSAT-2 FTS-2 L1B TIR product file.
flag_in	in	OBSNO ：The sounding is specified by index in the file. LATLON：The sounding is specified by latitude and longitude.
num	in	The index of soundings to be read. The parameter is available only if flag_in is OBSNO. 1 ≦ num ≦ numPoints (*) (*) numPoints：Number of the soundings in the file.
lat, lon	in	The latitude and longitude to specify the sounding. [Degree] The parameter is available only if flag_in is LATLON. -90 ≦ lat ≦ 90 (degree) -180 ≦ lon ≦ 180 (degree)
band	in	The band number band = 4 or 5

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_fts2_rad2bt calculates brightness temperature of the specified sounding.

The sounding can be specified by either the index in the file or latitude/longitude. If the sounding is specified by latitude/longitude, the nearest sounding in file will be returned.

The calculated data will be stored into the new memory address *datax and *datay which the API allocated.

NOTE

Application needs to call free( ) to release memory datax and datay.

SAMPLES

sample_fts2_rad2bt

6. 3. 4 gtk_fts2_readcam

NAME

gtk_fts2_readcam – Read CAM data and export to JPEG file

SYNTAX

```
#include "gosat_tk.h"

int gtk_fts2_readcam(const hid_t file_id, const int num);
```

PARAMETERS

Parameter	I/O	Description
file_id	in	hdf5 file id of GOSAT-2 FTS-2 L1B Common product file.
num	in	The index of CAM data to be read. 1 ≦ num ≦ numPoints (*) (*) numPoints: Number of CAM data

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_fts2_readcam reads the specified camera data and export it to the JPEG file in current working directory. The JPEG file is saved as the following file name.

File name: YYYY-MM-DDThh.mm.ss.ffffffZ.jpg

SAMPLES

sample_fts2_readcam

6．4 GOSAT-1 CAI Level 1 product API

6．4．1 gtk_com_readstruct

NAME

gtk_com_readstruct – Get CAI time structure data

SYNTAX

```
#include "gosat_tk.h"

GTK_RET gtk_com_readstruct(
        int* ndims, hsize_t dims[H5S_MAX_RANK],
        GTK_USER_DATE_YMDHMS** data,
        const hid_t file_id, const char* path);

typedef struct gtk_user_date_ymdhms{
          int year;
          char month;
          char day;
          char hour;
          char min;
          float sec;
}GTK_USER_DATE_YMDHMS;
```

PARAMETERS

Parameter	I/O	Description
ndims	out	Pointer to the location in which the number of dimensions to be returned.
dims	out	Pointer to the location in which element size of dimensions to be returned.
data	out	Pointer to the location in which obtained data to be returned.
file_id	in	hdf5 file identifier of GOSAT-2 CAI L1A product.
path	in	Dataset

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_com_readstruct reads the specified dataset as CAI time structure data. The obtained data will be stored into the new memory address *data which the API allocated.

NOTE

Application needs to call free() to release memory *data.

SAMPLES

sample_com

6．4．2　　gtk_cai1_geointerpol

NAME

gtk_cai1_geointerpol – Get CAI latitude/longitude (by interpolation).

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai1_geointerpol( int* data_dims, GTK_GEO_DATA** data,
                                    hid_t file_id, int band );

typedef struct{
          double latitude_deg;      /* degree */
          double longitude_deg;   /* degree */
}GTK_GEO_DATA;
```

PARAMETERS

Parameter	I/O	Description
data_dims	out	Pointer to the location in which the number of dimensions to be returned. The following dimension will be returned. data_dims[0]: The number of lines data_dims[1]: The number of pixels
data	out	Pointer to the location in which latitude/longitude data to be returned.
file_id	in	hdf5 file identifier of GOSAT-1 CAI L1A product
band	in	The band number 1 ≦ band ≦ 4

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

CAI product contains the latitude/longitude dataset for subset of pixel/lines. gtk_cai1_geointerpol reads the dataset and interpolates them for all pixels/lines.

The interpolated data will be stored into the new memory address *data which the API allocated.

The latitude/longitude at pixel x, line y is stored to (*data)[y*X+x] where X is data_dims[1] corresponding to the number of pixel per line.

NOTE

Application needs to call free() to release memory *data.

SAMPLES

sample_cai1_geointerpol

6．4．3　gtk_cai1_dn2rad

NAME

gtk_cai1_dn2rad – Convert CAI observation data to radiance

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai1_dn2rad(int* data_dims, float** data, const hid_t file_id,
        const int band, const char* config_folder_name);
```

PARAMETERS

Parameter	I/O	Description
data_dims	out	Pointer to the location in which the number of dimensions to be returned. The following dimension will be returned. data_dims[0]: The number of lines data_dims[1]: The number of pixels
data	out	Pointer to the location in which the radiance to be returned.　[W/m^2/str/um]
file_id	in	hdf5 file identifier of GOSAT-1 CAI L1A product
band	in	The band number 1 ≦ band ≦ 4
config_folder_name	in	The folder name which the parameter files are stored in. (example: "parameter/")

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_cai1_dn2rad reads image data storing as digital number(DN) and convert it to the radiance.

The converted data will be stored into the new memory address *data which the API allocated.

The radiance at pixel x, line y is stored to (*data)[y*X+x] where X is data_dims[1] corresponding to the number of pixel per line.

NOTE

Application needs to call free() to release memory *data.

SAMPLES

sample_cai1_dn2rad

6．4．4 gtk_cai1_cutimage

NAME

gtk_cai1_cutimage – Get image data about specific region

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai1_cutimage(int* data_dims, float** data, GTK_GEO_DATA** latlon,
    GTK_USER_DATE_YMDHMS** time, const hid_t file_id, GTK_FLAG_IN flag_in,
    const double line_lat1, const double pixel_lon1, const double line_lat2, const
    double pixel_lon2, const int band, CAI_FLAG_OUT flag_out, const char*
    config_folder_name);
```

GTK_USER_DATE_YMDHMS: Refer to 6．4．1.

GTK_GEO_DATA : Refer to 6．4．2

PARAMETERS

Parameter	I/O	Description
data_dims	out	Pointer to the location in which the number of dimensions to be returned. The following dimension will be returned. data_dims[0]: The number of lines data_dims[1]: The number of pixels
data	out	Pointer to the location in which the image to be returned. If flag_out is DN, data is digital number. If flag_out is RAD, data is radiance[W/m2/str/um].
latlon	out	Pointer to the location in which the latitude/longitude to be returned.
time	out	Pointer to the location in which the observation time to be returned.
file_id	in	hdf5 file identifier of GOSAT-1 CAI L1A product
flag_in	in	LNPX: The region is specified by line and pixel number LATLON : The region is specified by latitude and longitude
line_lat1,pixel_lon1 line_lat2,pixel_lon2	in	If flag_in is LNPX: Specify the region by line and pixel number. 1 ≦ band ≦ 3 1 ≦ line_lat1＜line_lat2≦ lines 1 ≦ line_lon1＜line_lon2≦ 2056 band=4 1 ≦ line_lat1＜line_lat2 ≦ lines 1 ≦ line_lon1＜line_lon2 ≦ 512 If flag_in is LATLON Specify the region by latitude/longitude [degree] -90 ≦line_lat1＜line_lat2≦ 90 -180 ≦ line_lon1＜line_lon2 ≦ 180

band	in	The band number 1 ≦ band ≦ 4
flag_out	in	Specify output data type. DN ：digital number RAD：radiance
config_folder_name	in	The folder name which the parameter files are stored in. (example: "parameter/")

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_cai1_cutimage trims image data of specific region.

The region can be specified by either the pixels/lines or latitude/longitude. If the region is specified by latitude/longitude, the nearest pixel/line will be selected for trimming the region.

The obtained data will be stored into the new memory address *data, *latlon and *time which the API allocated.

The image data at pixel x, line y is stored to (*data)[y*X+x] where X is data_dims[1] corresponding to the number of pixel per line.

The latitude/longitude at pixel x, line y is stored to (*latlon)[y*X+x] where X is data_dims[1] corresponding to the number of pixel per line.

The observation time at line y is stored to (*time)[y].

NOTE

Application needs to call free() to release *data, *latlon,*time.

SAMPLES

sample_cai1_cutimage

6．5 GOSAT-1 CAI-2 Level 1 product API

6．5．1 gtk_cai2_geointerpol

NAME

gtk_cai2_geointerpol – Get CAI-2 latitude/longitude (by interpolation).

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai2_geointerpol(int* data_dims, GTK_GEO_DATA** data, const hid_t file_id);

typedef struct{
        double latitude_deg;    /* degree */
        double longitude_deg; /* degree */
}GTK_GEO_DATA;
```

PARAMETERS

Parameter	I/O	Description
data_dims	out	Pointer to the location in which the number of dimensions to be returned. The following dimension will be returned. data_dims[0]: The number of lines data_dims[1]: The number of pixels
data	out	Pointer to the location in which latitude/longitude data to be returned.
file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A FWD/BWD product

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

CAI-2 product contains the latitude/longitude dataset for the subset of pixel/lines. gtk_cai2_geointerpol reads the dataset and interpolates them for all pixels/lines.

The interpolated data will be stored into the new memory address *data which the API allocated.

The latitude/longitude at pixel x, line y is stored to (*data)[y*X+x] where X is data_dims[1] corresponding to the number of pixel per line.

NOTE

Application needs to call free() to release memory *data.

SAMPLES

sample_cai2_geointerpol

6．5．2　　gtk_cai2_dn2rad

NAME

gtk_cai2_dn2rad – Convert CAI-2 observation data to radiance

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai2_dn2rad(  hid_t    com_file_id,
                      hid_t    band_file_id,
                      const    char*   config_folder_name,
                      int      band, int        start_pixel,
                      int      end_pixel,
                      int      start_line,
                      int      end_line,
                      float*   corr_image,
                      int      corr_image_num_pixels_per_line,
                      int      corr_image_num_lines
);
```

PARAMETERS

Parameter	I/O	Description
com_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A Common product
band_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A FWD/BWD product.
config_folder_name	in	The folder name which the parameter files are stored in. (example: "parameter/")
band	in	The band number 1 ≦ band ≦ 5 (forward) 6 ≦ band ≦ 10 (backward)
start_pixel, end_pixel start_line, end_line	in	Specify region by pixel/line band=1,2,3,4,6,7,8 and 9: 1 ≦ start_pixel＜end_pixel ≦ 2056 1 ≦ start_line ＜end_line ≦ lines_500 band = 5 and 10: 1 ≦ start_pixel＜end_pixel ≦ 1024 1 ≦ start_line ＜end_line ≦ lines_1km
corr_image	out	Pointer to the location in which the radiance to be returned. [W/m^2/μm/str]
corr_image_num_pixels_per_line	in	The number of pixels per line of corr_image
corr_image_num_lines	in	The number of line of corr_image

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_cai2_dn2rad converts image data storing as digital number(DN) to the radiance.

The converted data will be stored into the address corr_image which application specified.

The radiance at pixel x, line y is stored to corr_image[y*X+x] where X is corr_image_num_pixels_per_line. Application must allocate corr_image size with sizeof(float)×corr_image_num_pixels_per_line×corr_image_num_lines.

SAMPLES

sample_cai2_dn2rad

6．5．3　gtk_cai2_cutimage

NAME

gtk_cai2_cutimage – Get image data about specific region

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai2_cutimage(int* data_dims, float** data, GTK_GEO_DATA** latlon,
    char** time, const hid_t file_id, const hid_t file_id_com, GTK_FLAG_IN flag_in,
    const double line_lat1, const double pixel_lon1, const double line_lat2, const
    double pixel_lon2, const int band, CAI_FLAG_OUT flag_out, const char*
    config_folder_name);
```

GTK_GEO_DATA：Refer to 6．5．1

PARAMETERS

Parameter	I/O	Description
data_dims	out	Pointer to the location in which the number of dimensions to be returned. The following dimension will be returned. data_dims[0]: The number of lines data_dims[1]: The number of pixels
data	out	Pointer to the location in which the image to be returned. If flag_out is DN, data is digital number. If flag_out is RAD, data is radiance [W/m2/str/um]
latlon	out	Pointer to the location in which the latitude/longitude to be returned.
time	out	Pointer to the location in which the observation time to be returned.
file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A FWD/BWD product
file_id_com	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A Common product
flag_in	in	LNPX: The region is specified by line and pixel number LATLON：The region is specified by latitude/longitude
line_lat1,pixel_lon1 line_lat2,pixel_lon2	in	Specify region by pixel/line or latitude/longitude If flag_in = LNPX: band=1,2,3,4,6,7,8 and 9: 1 ≦ start_pixel＜end_pixel ≦ 2056 1 ≦ start_line ＜end_line ≦ lines_500 band = 5 and 10: 1 ≦ start_pixel＜end_pixel ≦ 1024 1 ≦ start_line ＜end_line ≦ lines_1km

		If flag_in = LATLON -90 ≦ line_lat1＜line_lat2≦ 90 -180≦ pixel_lon1＜pixel_lon2≦ 180
band	in	The band number. 1 ≦ band ≦ 10
flag_out	in	Specify output data type. DN ：digital number RAD：radiance
config_folder_name	in	The folder name which the parameter files are stored in. (example: "parameter/")

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_cai2_cutimage trimes image data of specific region.

The region can be specified by either the pixels/lines or latitude/longitude.

If the region is specified by latitude/longitude, the nearest pixel/line will be selected for trimming the region.

The obtained data will be stored into the new memory address *data, *latlon and *time which the API allocated.

The image data at pixel x, line y is stored to (*data)[y*X+x] where X is data_dims[1] corresponding to the number of pixel per line.

The latitude/longitude at pixel x, line y is stored to (*latlon)[y*X+x] where X is data_dims[1] corresponding to the number of pixel per line.

The observation time at line y is stored to (*time)[y].

NOTE

Application needs to call free() to release *data, *latlon,*time.

SAMPLES

sample_cai2_cutimage

6. 5. 4 gtk_cai2_geocalc

NAME

gtk_cai2_geocalc – Calculate geolocation

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai2_geocalc(
                hid_t   band_file_id,
                const   char* config_folder_name,
                int     band,
                int     start_pixel,
                int     end_pixel,
                int     start_line,
                int     end_line,
                CAI2_GeoData* geo_data,
                int     geo_data_num_pixels_per_line,
                int     geo_data_num_lines );

typedef struct{
        double  latitude;       /* degree */
        double   longitude;     /* degree */
} CAI2_GeoData;
```

PARAMETERS

Parameter	I/O	Description
com_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A Common product
config_folder_name	in	The folder name which the parameter files are stored in. (example: "parameter/")
band	in	The band number. 1 ≦ band ≦ 5 (forward) 6 ≦ band ≦ 10 (backward)
start_pixel, end_pixel start_line, end_line	in	Specify region by pixel/line band=1,2,3,4,6,7,8 and 9: 1 ≦ start_pixel＜end_pixel ≦ 2056 1 ≦ start_line ＜end_line ≦ lines_500 band = 5 and 10: 1 ≦ start_pixel＜end_pixel ≦ 1024 1 ≦ start_line ＜end_line ≦ lines_1km

geo_data	out	Pointer to the location in which latitude/longitude data to be returned.
geo_data_num_pixels_per_line	in	The number of pixels per line of geo_data
geo_data_num_lines	in	The number lines of geo_data

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_cai2_geocalc calculates geolocation within the specific region. The calculated geolocation will be stored into the address geo_data which application specified.
The latitude/longitude at pixel x, line y is stored to geo_data[y*X+x] where X is geo_data_num_pixels_per_line. Application must allocate geo_data size with sizeof(CAI2_GeoData)×geo_data_num_pixels_per_line×geo_data_num_lines.

SAMPLES

gtk_cai2_geocalc

6．5．5 gtk_cai2_resampimage

NAME

gtk_cai2_resampimage – Get CAI-2 resampled image by band-to-band registration

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai2_resampimage(   hid_t    com_file_id,
                            hid_t    band_file_id,
                            const    char* config_folder_name,
                            int      band,
                            int      start_pixel,
                            int      end_pixel,
                            int      start_line,
                            int      end_line,
                            float*   corr_image,
                       int      corr_image_num_pixels_per_line,
                       int      corr_image_num_lines,
                       enum     CAI2_RADIO_METRIC_CORR_ON_OFF
                                radio_metric_correction );
```

PARAMETERS

Parameter	I/O	Description
com_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A Common product
band_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A FWD/BWD product.
config_folder_name	in	The folder name which the parameter files are stored in. (example: "parameter/")
band	in	The band number. 1 ≦ band ≦ 5 (forward) 6 ≦ band ≦ 10 (backward)
start_pixel, end_pixel start_line, end_line	in	Specify region by pixel/line band=1,2,3,4,6,7,8 and 9: 1 ≦ start_pixel＜end_pixel ≦ 2056 1 ≦ start_line ＜end_line ≦ lines_500 band = 5 and 10: 1 ≦ start_pixel＜end_pixel ≦ 1024 1 ≦ start_line ＜end_line ≦ lines_1km
corr_image	out	Pointer to the resampled image data to be returned. If radio_metric_correction is CAI2_RADIO_METRIC_CORR_OFF

		Digital number. If radio_metric_correction is CAI2_RADIO_METRIC_CORR_ON Radiance [W/m2/μ m/str].
corr_image_num_pixels_per_line	in	The number of pixels per line of corr_image
corr_image_num_lines	in	The number of lines of corr_image
radio_metric_correction	in	Specify output data type. CAI2_RADIO_METRIC_CORR_OFF: Digital number CAI2_RADIO_METRIC_CORR_ON:] Radiance [W/m2/μ m/str].

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_cai2_resampimage resamples image to be overlaid on the base band (band-to-band registration). The resampled image will be stored into the address corr_image which application specified.

The image data at pixel x, line y is stored to corr_image[y*X+x] where X is corr_image_num_pixels_per_line. Application must allocate corr_image size with sizeof(float)×corr_image_num_pixels_per_line×corr_image_num_lines.

SAMPLES

sample_cai2_resampimage

6．5．6　gtk_cai2_create_sensor_param

NAME

gtk_cai2_create_sensor_param – Read sensor parameter and create instance.

SYNTAX

```
#include "gosat_tk.h"

CAI2_SensorParameter* gtk_cai2_create_sensor_param(const char* config_folder);
```

PARAMETERS

Parameter	I/O	Description
config_folder	in	The folder name which the parameter files are stored in. (example: "parameter/")

RETURN VALUE

Return value	Description
Not NULL(0)	instanced sensor parameter
NULL(0)	Failure

DESCRIPTIONS

gtk_cai2_create_sensor_param reads the parameter file in specified folder and create instance of sensor parameter.

NOTE

Application needs to call gtk_cai2_delete_sensor_param() to release instance.

SAMPLES

sample_cai2_create_processed_image

6．5．7　gtk_cai2_delete_sensor_param

NAME

gtk_cai2_delete_sensor_param – Delete sensor parameter instance

SYNTAX

```
#include "gosat_tk.h"

void gtk_cai2_delete_sensor_param(CAI2_SensorParameter* sns_parameter);
```

PARAMETERS

Parameter	I/O	Description
sns_parameter	in	Sensor parameter

RETURN VALUE

No return value.

DESCRIPTIONS

gtk_cai2_delete_sensor_param deletes the specified sensor parameter instance which is created by gtk_cai2_create_sensor_param.

SAMPLES

sample_cai2_create_processed_image

6. 5. 8 gtk_cai2_create_processed_image

NAME

gtk_cai2_create_processed_image – Create image processed data.

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai2_create_processed_image(
                                      hid_t   com_file_id,
                                      hid_t   fwd_band_file_id,
                                      hid_t   bwd_band_file_id,
        const CAI2_SensorParameter* sns_parameter,
                                      CAI2_IMAGE image[10] );

typedef struct{
         int      band;     /*  The band number (1 to 10) */
         float*   data;     /*  The radiance [W/m2/μm/str] */
         char*    flg;      /*  The flag of pixel attribute */
         int      pixels_per_line; /* the number of pixels per line */
         int      lines;  /* the number of line */
         unsigned int image_processing;
                          /* The flag indicating one or more image processing which
                             is actually applied. */
}CAI2_IMAGE;
```

PARAMETERS

Parameter	I/O	Description
com_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A Common product
fwd_band_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A FWD product
bwd_band_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A BWD product
sns_parameter	in	The instance of sensor parameter
image	out	Pointer to the location in which image data (band 1 to 10) to be stored.

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_cai2_create_processed_image reads the common, forward and backward product file and create processed image.

The observation data is converted from digital number to radiance and applied stray light corrections. Finally the corrected image data is resampled to the base band. The created image of band 1 to 10 are stored into image[0] to image[9] for each band.

The image[i] contains following data.

band: The band number, 1 to 10.

data: Radiance [W/m^2/μ m/str].
The data is the array of pixels_per_line×lines allocated by the API.
The radiance at pixel x line y is stored to data[y* pixels_per_line +x].
If there are no observation data, data will be NULL.
If there are no observation data about the base band, data will be NULL.

flg: The flag of pixel attribute.
The flg is the array of pixels_per_line×lines allocated by the API.
The value at pixel x line y is stored to flg[y* pixels_per_line +x].

flg[n] indicates following pixel attribute.
0 : normal.
1 : no data. The missing data or the data with no corresponding position obtained by band-to-band registration.
2 : out of range to be applied image processing.
4 : saturated (corrected by saturation correction)

If there are no observation data, flg will be NULL.
If there are no observation data about the base band, flg will be NULL.

pixels_per_line: The number of pixels per line.
If there are no observation data, the value is zero, otherwise the value is same value with the one of the base band. If there are no observation data about the base band, pixels_per_line is zero.

lines: The number of line.
If there are no observation data, the value is zero, otherwise the value is the same value as the one of the base band. If there are no observation data about the base band, pixels_per_line is zero

image_processing: The flag which indicates applied image processing. The flag is one or more combination of following value.

CAI2_IMGPRC_B1B6_SATURATION_CORRECTION	1<<0
CAI2_IMGPRC_B1B6_STRAY_LIGHT_CORRECTION	1<<1
CAI2_IMGPRC_B1B6_OUTBAND_STRAY_LIGHT_CORRECTION	1<<2
CAI2_IMGPRC_B1B6_B1B6_CROSSTALK_CORRECTION	1<<3
CAI2_IMGPRC_B5B10_CH_CROSSTALK_CORRECTION	1<<16
CAI2_IMGPRC_B5B10_STRAY_LIGHT_CORRECTION	1<<17

Example:
The following value indicates that all processes for band 1 to 6 are applied.

```
image_processing =
      CAI2_IMGPRC_B1B6_SATURATION_CORRECTION
   | CAI2_IMGPRC_B1B6_STRAY_LIGHT_CORRECTION
   | CAI2_IMGPRC_B1B6_STRAY_LIGHT_CORRECTION_OUTBAND
   | CAI2_IMGPRC_B1B6_B1B6_CROSSTALK_CORRECTION
```

The image processing setting is configured by parameter file.
For more detail about parameter file refer to “Parameter part” of user’s manual.

The size of output image is the same as the base band.

The flg[n] is set to 1 if data[n] is missing data. Otherwise flg[n] will be 1 or 2 even if the reference band/data for data[n] is missing.
If the pixel is out of valid range in one or more image processing, flg[n] is set to 2. For example, the band1 and 6 crosstalk correction requires both forward and backward looking observation, so the flg for line[n] which is observed only forward(or backward) looking will be set to 2. In this case, the band 1 and band 6 crosstalk correction for n is skipped and data[n] remains without the crosstalk correction.

gtk_cai2_create_processed_image processes all of available observation data in fwd_band_file_id and bwd_band_file_id. If there are no available observation data in both the file, gtk_cai2_create_processed_image returns failure.

NOTE

Application needs to call gtk_cai2_delete_image () to release image.

SAMPLES

sample_cai2_create_processed_image

6. 5. 9 gtk_cai2_delete_image

NAME

gtk_cai2_delete_image – Delete image data

SYNTAX

```
#include "gosat_tk.h"

void gtk_cai2_delete_image( CAI2_IMAGE image[10] );
CAI2_IMAGE : Refer to gtk_cai2_create_prcessed_image
```

PARAMETERS

Parameter	I/O	Description
image	in	Image data

RETURN VALUE

No return value.

DESCRIPTIONS

gtk_cai2_delete_image deletes image data which was created by gtk_cai2_create_processed_image.

If there are no image data on band n (n=1 to 10), both image[n-1].data, image[n-1].flg must be NULL. After deleting, all of image data is initialized with zero.

SAMPLES

sample_cai2_create_processed_image

6. 5. 10 gtk_cai2_create_resample_info

NAME

gtk_cai2_create_resample_info

– Create information instance for band-to-band registration

SYNTAX

```
#include "gosat_tk.h"

CAI2_RESAMPLE_INFO*   gtk_cai2_create_resample_info(
        hid_t     fwd_band_file_id,
        hid_t     bwd_band_file_id,
        const CAI2_SensorParameter* sns_parameter)

struct CAI2_RESAMPLE_INFO*   /* Band-to-band registration info structure */
```

PARAMETERS

Parameter	I/O	Description
fwd_band_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A FWD product
bwd_band_file_id	in	hdf5 file identifier of GOSAT-2 CAI-2 L1A BWD product
sns_parameter	in	The instance of sensor parameter

RETURN VALUE

Return value	Description
Not NULL(0)	instanced parameter
NULL(0)	Failure

DESCRIPTIONS

gtk_cai2_create_resample_info creates the instance of parameter for band-to-band registration.

gtk_cai2_create_resample_info obtains the observation time from product file for the line registration. If one or both of fwd_band_id and bwd_band_file_id are valid, gtk_cai2_create_resample_info will process the valid file id and return with success. If neither fwd_band_id nor bwd_band_file_id are available, the function will return failure.

NOTE

Application needs to call gtk_cai2_delete_resample_info () to release image.

SAMPLES

sample_cai2_get_resample_position

6. 5. 11 gtk_cai2_get_resample_position

NAME

gtk_cai2_get_resample_position

- Get pixel number/line number corresponding to a specified position on the base band.

SYNTAX

```
#include "gosat_tk.h"

int gtk_cai2_get_resample_position (
                                    int   baseband_pixel_no,
                                    int   baseband_line_no,
                                    int   refband,
                                    int* refband_pixel_no,
                                    int* refband_delta_line_no,
                                    const CAI2_RESAMPLE_INFO* resample_info)
```

PARAMETERS

Parameter	I/O	Description
baseband_pixel_no	in	The pixel number of the base band. (one-based pixel number)
baseband_line_no	in	The line number of the base band. (one-based line number)
refband	in	The band number for asking. (1 to 10)
refband_pixel_no	out	Pointer to the location in which pixel number of reference band to be returned. (one-based pixel number)
refband_line_no	out	Pointer to the location in which line number of reference band to be returned. (one-based line number)
resample_info	in	Band-toband registration information

RETURN VALUE

Return value	Description
0	Success
Otherwise	Failure

DESCRIPTIONS

gtk_cai2_get_resample_position obtains the pixel number and line number corresponding to the specified position on the base band. The followings are description about output data.

- refband_pixel_no: The band’s pixel number corresponding to baseband_pixel_no on the base band.

If the pixel number is out of image of the specified band, refband_pixel_no is zero.

- refband_line_no: The band’s line number corresponding to baseband_line_no on the base band.
  If the line number is out of image of the specified band, refband_line_no is zero.

Refer to gtk_cai2_create_resample_info to create resample_info instance. The API will be failed if one of the specified band number, pixel number and line number are not valid or the information of target is missing in band resample_info.

SAMPLES

sample_cai2_get_resample_position

6. 5. 12 gtk_cai2_delete_resample_info

NAME

gtk_cai2_delete_resample_info

– Delete band-to-band registration information

SYNTAX

```
#include "gosat_tk.h"

void gtk_cai2_delete_resample_info(CAI2_RESAMPLE_INFO* resample_info)
```

PARAMETERS

Parameter	I/O	Description
resample_info	in	Band-to-band registration information

RETURN VALUE

No return value.

DESCRIPTIONS

gtk_cai2_delete_resample_info deletes image data which was created by gtk_cai2_create_resample_info.

SAMPLES

sample_cai2_get_resample_position

(END)