

GOSAT-1/2 レベル1 プロダクト読み出しツール

取扱説明書(C 言語編)

2019年8月

宇宙航空研究開発機構

改訂履歴

版数	日付	改訂頁	改訂内容
初版	2019/8	—	

目次

1.	はじめに	1
1. 1	サポートするプロダクト.....	1
1. 2	アプリケーションプログラムインタフェース (API) 一覧.....	2
2.	動作環境	3
3.	インストール	4
4.	パラメータファイル.....	5
5.	サンプルプログラム.....	6
6.	API 仕様.....	8
6. 1	HDF ファイル操作.....	8
6. 1. 1	gtk_com_hdfopen.....	8
6. 1. 2	gtk_com_hdfclose.....	9
6. 1. 3	gtk_com_dimsize.....	10
6. 1. 4	gtk_com_read.....	11
6. 1. 5	gtk_com_dump.....	13
6. 2	GOSAT-1 FTS Level 1 プロダクト API.....	14
6. 2. 1	gtk_fts1_getigm.....	14
6. 2. 2	gtk_fts1_getspc.....	16
6. 2. 3	gtk_fts1_rad2bt.....	18
6. 2. 4	gtk_fts1_readcam.....	20
6. 3	GOSAT-2 FTS-2 Level 1 プロダクト API.....	21
6. 3. 1	gtk_fts2_getigm.....	21
6. 3. 2	gtk_fts2_getspc.....	23
6. 3. 3	gtk_fts2_rad2bt.....	25
6. 3. 4	gtk_fts2_readcam.....	27
6. 4	GOSAT-1 CAI Level 1 プロダクト API.....	28
6. 4. 1	gtk_com_readstruct.....	28
6. 4. 2	gtk_cai1_geointerpol.....	30
6. 4. 3	gtk_cai1_dn2rad.....	31
6. 4. 4	gtk_cai1_cutimage.....	32
6. 5	GOSAT-1 CAI-2 Level 1 プロダクト API.....	34
6. 5. 1	gtk_cai2_geointerpol.....	34
6. 5. 2	gtk_cai2_dn2rad.....	35

6. 5. 3	gtk_cai2_cutimage.....	37
6. 5. 4	gtk_cai2_geocalc.....	39
6. 5. 5	gtk_cai2_resampimage.....	41
6. 5. 6	gtk_cai2_create_sensor_param.....	43
6. 5. 7	gtk_cai2_delete_sensor_param.....	44
6. 5. 8	gtk_cai2_create_processed_image.....	45
6. 5. 9	gtk_cai2_delete_image.....	49
6. 5. 10	gtk_cai2_create_resample_info.....	50
6. 5. 11	gtk_cai2_get_resample_position.....	51
6. 5. 12	gtk_cai2_delete_resample_info.....	53

1. はじめに

この文書は、GOSAT レベル 1 プロダクト読み出しツールキット（以下 GTK）の取扱説明書です。GTK は、GOSAT-1/GOSAT-2 のレベル 1 プロダクトを読み込むプログラムライブラリです。本書では、C 言語版の利用方法について示します。

1. 1 サポートするプロダクト

GTK で読み出しの対象となるプロダクトを表 1. 1-1 に示します。

表 1. 1-1 対象プロダクト一覧

No	データ名称	対象バージョン
1	TANSO-FTS レベル 1A プロダクト	V2XX
2	TANSO-FTS レベル 1B プロダクト	V2XX
3	TANSO-CAI レベル 1A プロダクト	V130
4	TANSO-FTS-2 レベル 1A プロダクト	V1XX
5	TANSO-FTS-2 レベル 1B プロダクト	V1XX
6	TANSO-CAI-2 レベル 1A プロダクト	V1XX

プロダクトの説明は以下の文書をご参照ください。

- GOSAT / TANSO Level 1 Product Description Document TANSO-FTS Section
- レベル 1 データフォーマット説明書 TANSO-CAI 編
- ミッション運用系システム_レベル 1 データフォーマット説明書 (TANSO-FTS-2 編)
- ミッション運用系システム_レベル 1 データフォーマット説明書 (TANSO-CAI-2 編)

1. 2 アプリケーションプログラムインタフェース(API)一覧

表 1. 2-1 に API 一覧を示します。

表 1. 2-1 API 一覧

No.	関数名	機能
1.	gtk_com_hdfopen	GOSAT-1/2 HDF ファイルオープン
2.	gtk_com_hdfclose	HDF ファイルクローズ
3.	gtk_com_dimsize	ディメンジョンサイズ取得
4.	gtk_com_readi1s	1byte 符号あり整数型データ取得
5.	gtk_com_readi1u	1byte 符号なし整数型データ取得
6.	gtk_com_readi2s	2bytes 符号あり整数型データ取得
7.	gtk_com_readi2u	2bytes 符号なし整数型データ取得
8.	gtk_com_readi4s	4bytes 符号あり整数型データ取得
9.	gtk_com_readi4u	4bytes 符号なし整数型データ取得
10.	gtk_com_readr4	4bytes 実数型データ取得
11.	gtk_com_readr8	8bytes 実数型データ取得
12.	gtk_com_readchar	文字型データ取得
13.	gtk_com_readstruct	GOSAT-1 時刻構造体データ取得
14.	gtk_com_dump	データセットダンプ
15.	gtk_fts1_getigm	FTS/L1A インタフェログラム取得
16.	gtk_fts1_getspc	FTS/L1B スペクトル取得
17.	gtk_fts1_rad2bt	FTS/TIR 輝度温度取得
18.	gtk_fts1_readcam	FTS/視野確認カメラデータ出力
19.	gtk_fts2_getigm	FTS-2/L1A インタフェログラム取得
20.	gtk_fts2_getspc	FTS-2/L1B スペクトル取得
21.	gtk_fts2_rad2bt	FTS-2/TIR 輝度温度取得
22.	gtk_fts2_readcam	FTS-2/視野確認カメラデータ出力
23.	gtk_cai1_geointerpol	CAI 緯度経度取得(補間)
24.	gtk_cai1_dn2rad	CAI 輝度取得
25.	gtk_cai1_cutimage	CAI 輝度/DN 値取得
26.	gtk_cai2_geointerpol	CAI-2 緯度経度取得(補間)
27.	gtk_cai2_dn2rad	CAI-2 輝度取得
28.	gtk_cai2_cutimage	CAI-2 指定領域取得
29.	gtk_cai2_geocalc	CAI-2 緯度経度算出
30.	gtk_cai2_resampimage	CAI-2 画像リサンプリング(バンド間レジストレーション)
31.	gtk_cai2_create_sensor_param	CAI-2 センサパラメータ読込
32.	gtk_cai2_delete_sensor_param	CAI-2 センサパラメータ破棄
33.	gtk_cai2_create_prcessed_image	CAI-2 画像処理済データ生成
34.	gtk_cai2_delete_image	CAI-2 画像データ破棄
35.	gtk_cai2_create_resample_info	CAI-2 バンド間レジストレーション情報生成
36.	gtk_cai2_get_resample_position	CAI-2 基準バンドの画素番号・ライン番号に該当する参照画像の画素番号・ライン番号を取得
36.	gtk_cai2_delete_resample_position	CAI-2 バンド間レジストレーション情報破棄

2. 動作環境

表 2-1 に、動作環境に必要な要件を示します。

表 2-1 動作環境要件

実行環境	要件
メモリ	8GB 以上
OS	Linux、Windows10(Cygwin)、macOS
C コンパイラ	C99 (ISO/IEC 9899:1999) 対応コンパイラ
必要なソフトウェア	HDF5 ライブラリ (バージョン 1.8) JPEG ライブラリ (バージョン 9) TIFF ライブラリ (バージョン 4) OpenCV ライブラリ (バージョン 2.4)

C コンパイラを含む開発環境、必要なソフトウェアについては、あらかじめインストールされている必要があります。

表 2-2 に、確認済みの動作環境を示します。

表 2-2 動作確認環境一覧

名称	ハードウェア	ソフトウェア
Linux	OS: Linux RHEL7 .2 (Kernel 3.10.0-327.el7.x86_64) CPU: Intel(R) Xeon(R) CPU E3-1271 v3 @ 3.60GHz メモリ: 16 GB	HDF5-1.8.15-patch1 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 メモリ: 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 メモリ : 32 GB	HDF5-1.8.16 libjpeg 9.0 libtiff 4.0.9 opencv 2.4.13 gcc4.9.3

3. インストール

GTK のソースコードは、tar.gz で圧縮されています。以下のコマンドで展開します。

```
$ tar zxvf gosat_tk-C-(バージョン).tar.gz
```

表 3-1 に、展開後のディレクトリ構成を示します。

表 3-1 展開後のディレクトリ構成

内容	説明
VERSION	バージョンが記載されています。
src/	ライブラリのソースコードが格納されています。
include/	ヘッダファイルが格納されています。
sample/	サンプルプログラムのソースコードが格納されています。

下記フォルダに移動し、“make” を実行してライブラリを作成します。

```
$ cd gosat_tk-C-(バージョン)
$ make
```

src フォルダにツールキットのライブラリ(libgosat_tk.a) が作成されます。

以下のコマンドで必要なソフトウェアのパスを指定できます。

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

以下のコマンドで、作成したライブラリとヘッダをインストールします。

```
$ make install
```

デフォルトでは、/usr/local/include, /usr/local/lib にヘッダとライブラリがインストールされます。

インストール先のディレクトリを変更する場合は、以下のコマンドで指定します。


```
$ make install PREFIX=" path"
```

path/include、path/lib にインストールされます。

4. パラメータファイル

CAI/CAI-2 用の API は、動作のためにパラメータファイルが必要です。パラメータファイルについての詳細は、以下を参照ください。

「GOSAT-1/2 レベル 1 プロダクト読み出しツール取扱い説明書(パラメータファイル編)」

パラメータファイルは、tar.gz で圧縮されています。以下のコマンドで展開します。

```
$ tar zxvf gosat_tk-parameter-(バージョン).tar.gz
```

表 4-1 に、展開後のディレクトリ構成を示します。

表 4-1 パラメータファイルのディレクトリ構成

内容	説明
parameter/	パラメータファイル一式が格納されています。

5. サンプルプログラム

動作確認を行うため、パッケージにはサンプルプログラムを同梱しています。
表 5-1 にサンプルプログラムの一覧を示します。

表 5-1 サンプルプログラム一覧

No.	サンプルプログラム	内容
1.	sample_com	GOSAT-1/2 プロダクトファイルをオープンし、データセットからデータを読み出す。
2.	sample_fts1_getigm	FTS/L1A プロダクトからインタフェログラムデータを取得する。
3.	sample_fts1_getspc	FTS/L1A プロダクトから、スペクトルデータを取得する。
4.	sample_fts1_rad2bt	FTS/L1B プロダクトから輝度温度を取得する。
5.	sample_fts1_readcam	FTS/L1B プロダクトから視野確認カメラデータを取得する。
6.	sample_fts2_getigm	FTS-2/L1A プロダクトからインタフェログラムデータを取得する。
7.	sample_fts2_getspc	FTS-2/L1B プロダクトからスペクトルデータを取得する。
8.	sample_fts2_rad2bt	FTS-2/L1B(TIR)プロダクトから輝度温度を取得する。
9.	sample_fts2_readcam	FTS-2 共通プロダクトから視野確認カメラデータを取得する。
10.	sample_cai1_geointerpol	CAI L1A プロダクトから各画素の緯度経度を内挿して求める。
11.	sample_cai1_dn2rad	CAI L1A プロダクトから1バンド分の輝度変換後データを取得する。
12.	sample_cai1_cutimage	CAI L1A プロダクトから1バンド分の画像を切り出す。
13.	sample_cai2_geointerpol	CAI-2 L1Aプロダクトから各画素の緯度経度を内挿して求める。
14.	sample_cai2_dn2rad	CAI-2 L1Aプロダクトから1バンド分の輝度変換後データを取得する。
15.	sample_cai2_cutimage	CAI-2 L1A プロダクトから1バンド分の画像を切り出す。
16.	sample_cai2_geocalc	CAI-2 L1A プロダクトから各画素の緯度経度を算出する。
17.	sample_cai2_resampimage	CAI-2 L1Aプロダクトから基準バンドに位置合わせした画像を取得する。
18.	sample_cai2_create_processed_image	CAI-2 L1A プロダクトから、各種画像処理を行い基準バンドへの位置合わせした全バンド分の画像データを生成する。
19.	sample_cai2_get_resample_position	CAI-2 L1A プロダクトから、基準バンドに対応する参照バンドの位置情報を取得する。

“sample” フォルダで以下のコマンドを実行し、サンプルプログラムをビルドします。

```
$ cd sample
$ make
```

sample フォルダにサンプルプログラムの実行ファイルが作成されます。

サンプルプログラムを実行するためには、テスト用のプロダクトが必要です。

テスト用のプロダクトをダウンロードします。

テスト用のプロダクトは tar. gz で圧縮されています。以下のコマンドで展開します。

```
$ tar zxvf gosat_tk-data-(バージョン).tar.gz
```

表 5-2 に、サンプルプログラム実行のためのディレクトリ構成を示します。テスト用のプロダクトファイルとパラメータファイルを表 5-2 のフォルダに格納します。

表 5-2 サンプルプログラム実行時のディレクトリ構成

内容	説明
sample/	サンプルコードが格納されています。
parameter/	パラメータファイルを格納します。
data/	サンプルプロダクトを格納します。

サンプルディレクトリに移動し、表 5-1 の一覧にある実行ファイルを実行します。

```
./sample_com
```

6. API 仕様

6.1 HDF ファイル操作

6.1.1 gtk_com_hdfopen

名前

gtk_com_hdfopen - GOSAT-1/2 プロダクトファイル(hdf)をオープンする。

書式

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

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

引数

変数名	I/O	説明
file	in	GOSAT-1/2 プロダクト(hdf5)のファイル名

戻り値

値	説明
0 ≤	hdf5 ファイル識別子
負	hdf5 ファイルオープン失敗/GOSAT-1/2 のファイルではない

注意: 戻り値の hid_t は、符号付き整数

説明

gtk_com_hdfopen は、GOSAT-1/2 のプロダクトファイル(hdf5)をオープンし、メタ情報のデータセットを読み込む。hdf5 ファイルが、GOSAT-1/2 のファイルであった場合、hdf5 のファイル識別子を返す。

注意

オープンしたファイルは、gtk_com_hdfclose でクローズすること。

サンプル

```
sample_com
```

6. 1. 2 gtk_com_hdfclose

名前

gtk_com_hdfclose - GOSAT-1/2 プロダクトファイルをクローズする。

書式

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

```
GTK_RET gtk_com_hdfclose(const hid_t file_id);
```

引数

変数名	I/O	説明
file_id	in	hdf5 ファイル識別子

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_com_hdfclose は、GOSAT-1/2 のプロダクトファイル(hdf5)をクローズする。

サンプル

```
sample_com
```

6. 1. 3 gtk_com_dimsize

名前

gtk_com_dimsize -データセットのディメンジョンサイズを取得する。

書式

```
#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);
```

引数

変数名	I/O	説明
ndims	out	ディメンジョン数を格納するポインタ
dims	out	ディメンジョンの要素数を格納するポインタ
file_id	in	hdf5 ファイル識別子
path	in	ディメンジョンを取得するデータセットのパス (例: "/Metadata/granuleID")

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_com_dimsize は、指定されたパスのデータセットのディメンジョンサイズを取得する。

サンプル

```
sample_com
```

6. 1. 4 gtk_com_read

名前

gtk_com_read - 指定したデータセットを読み込む

書式

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

【符号付き 1byte 整数型データ】

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

【符号なし 1byte 整数型データ】

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

【符号あり 2byte 整数型データ】

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

【符号なし 2byte 整数型データ】

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

【符号付き 4byte 整数型データ】

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

【符号なし 4byte 整数型データ】

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

【4byte 実数型データ】

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

【8byte 実数型データ】

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

【文字列型データ】

```
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);
```

引数

変数名	I/O	説明
ndims	out	ディメンジョン数を格納するポインタ
dims	out	ディメンジョンの要素数を格納するポインタ
data	out	読み込んだデータを格納するポインタ
slength	out	読み込んだデータの文字列長を格納するポインタ
str	out	読み込んだ文字列データを格納するポインタ
file_id	in	hdf5 ファイル識別子
path	in	読み込むデータセットのパス (例: " /Metadata/granuleID")

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

`gtk_com_read` は、指定されたパスのデータセットを読み込む。読み込んだデータは、API が割り当てた新規メモリアドレス `*data` に格納される。

多次元のデータセット読み込んだ場合、以下のように格納される。

データセットの次元	フォーマット説明書での多次元要素数の記述	data の参照先
2 次元	y, x	<code>*data[y*M+x]</code> M: <code>dims[1]</code>
3 次元	z, y, x	<code>*data[z*(M*N)+y*N+x]</code> M : <code>dims[2]</code> N : <code>dims[1]</code>

注意

アプリケーションは、利用しなくなった `*data` を `free()` で解放すること

サンプル

`sample_com`

6. 1. 5 gtk_com_dump

名前

gtk_com_dump - データセットの内容をファイルにダンプする。

書式

```
#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);
```

引数

変数名	I/O	説明
ndims	out	ディメンジョン数を格納するポインタ
dims	out	ディメンジョンの要素数を格納するポインタ
file_id	in	hdf5 ファイル識別子
path	in	ダンプするデータセットのパス (例: "/Metadata/granuleID")
fout	in	出力先のファイル名
format	in	GTK_FORMAT_BIN : binary mode GTK_FORMAT_TXT : text mode

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_com_dump は、指定したファイルにデータセットを出力する。出力形式は、テキストまたはバイナリを指定する。

サンプル

```
sample_com
```

6. 2 GOSAT-1 FTS Level 1 プロダクト API

6. 2. 1 gtk_ftsl_getigm

名前

gtk_ftsl_getigm- インタフェログラムデータを取得する。

書式

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

```
int gtk_ftsl_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);
```

引数

変数名	I/O	説明
ndata	out	インタフェログラムのサンプル数を格納するポインタ
datax	out	各サンプルの光路差長[cm]を格納するポインタ
datay	out	各サンプルのインタフェログラム[V]を格納するポインタ
file_id	in	GOSAT-1 FTS L1A プロダクトの hdf5 ファイル識別子
flag_in	in	出力するデータの指定方法 OBSNO : ファイル内の連番で指定する。 LATLON : 緯度・経度で指定する(最も近い観測点を選ばれる)
num	in	インタフェログラムデータを取得する観測データのインデックス。 flag_in が OBSNO の場合に有効 $1 \leq \text{num} \leq \text{numPoints} (*)$ (*) numPoints : ファイル内の観測データ数
lat, lon	in	インタフェログラムデータを取得する観測点の緯度経度[°] flag_in が LATLON の場合に有効。 $-90 \leq \text{lat} \leq 90$ (degree) $-180 \leq \text{lon} \leq 180$ (degree)
band	in	出力するバンド $1 \leq \text{band} \leq 4$
ps	in	偏波 バンド 1~3 の場合 : P : P polarization mode S : S polarization mode バンド 4 の場合 P

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

`gtk_fts1_getigm` は、指定した観測点のインターフェログラムデータを取得する。
観測点は、ファイル内のインデックス又は緯度経度で指定する。緯度経度を指定した場合は最も近い観測番号が選ばれる。

読み込んだデータは、新規に割り当てられたメモリアドレス `*datax`, `*datay` に格納される。

注意

アプリケーションは、利用しなくなった `*datax`, `*datay` を `free()` で解放すること。

サンプル

```
sample_fts1_getigm
```

6. 2. 2 gtk_fts1_getspc

名前

gtk_fts1_getspc - スペクトルデータを取得する

書式

```
#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);
```

引数

変数名	I/O	説明
ndata	out	スペクトルのサンプル数を格納するポインタ
datax	out	各サンプルの波数[cm-1]を格納するポインタ
datay	out	各サンプルのスペクトル[V/cm-1]を格納するポインタ
file_id	in	GOSAT-1 FTS L1B プロダクトの hdf5 ファイル識別子
flag_in	in	OBSNO : 観測点番号で指定する。 LATLON : 緯度・経度で指定する。
num	in	スペクトルデータを取得する観測データのインデックス。 flag_in が OBSNO の場合に有効 $1 \leq \text{num} \leq \text{numPoints}(*)$ (*) numPoints: 観測データ数
lat, lon	in	スペクトルデータを取得する観測点の緯度経度[°]。 flag_in が LATLON の場合に有効。 $-90 \leq \text{lat} \leq 90$ (degree) $-180 \leq \text{lon} \leq 180$ (degree)
band	in	出力するバンド $1 \leq \text{band} \leq 4$
ps	in	偏波 バンド 1~3 の場合 : P : P polarization mode S : S polarization mode バンド 4 の場合 P

戻り値

値	説明
0	正常
0 以外	異常

説明

gtk_fts1_getspc は、指定した観測点のスペクトルデータを取得する。

観測点はファイル内のインデックス又は緯度経度で指定する。緯度経度を指定した場合は最も近い観測番号が選ばれる。

読み込んだデータは、新規に割り当てられたメモリアドレス*datax, *datay に格納される。

注意

アプリケーションは、利用しなくなった *datax, *datay を free() で解放すること。

サンプル

sample_fts1_getspc

6. 2. 3 gtk_fts1_rad2bt

名前

gtk_fts1_rad2b - FTS/TIR バンドの分光放射輝度を輝度温度[K]に変換する

書式

```
#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);
```

引数

変数名	I/O	説明
ndata	out	輝度温度のサンプル数を格納するポインタ
datax	out	各サンプルの波数 [cm-1]を格納するポインタ
datay	out	各サンプルの輝度温度 [K]を格納するポインタ
file_id	in	GOSAT-1 FTS L1B プロダクトの hdf5 ファイル識別子
flag_in	in	OBSNO : 観測点番号で指定する。 LATLON : 緯度・経度で指定する。
num	in	輝度温度を取得する観測データのインデックス flag_in が OBSNO の場合に有効 $1 \leq \text{num} \leq \text{numPoints}(*)$ (*) numPoints: 観測データ数
lat, lon	in	輝度温度を取得する観測点の緯度経度[°] flag_in が LATLON の場合に有効。 $-90 \leq \text{lat} \leq 90$ (degree) $-180 \leq \text{lon} \leq 180$ (degree)

戻り値

値	説明
0	正常
0 以外	異常

説明

gtk_fts1_rad2bt は、指定した観測点の TIR バンドの輝度温度を取得する。

観測点はファイル内のインデックス又は緯度経度で指定する。緯度経度を指定した場合は最も近い観測番号が選ばれる。

読み込んだデータは、新規に割り当てられたメモリアドレス*datax, *datay に格納される。

注意

アプリケーションは、利用しなくなった datax, datay を free() で解放すること。

サンプル
sample_fts1_rad2bt

6. 2. 4 gtk_fts1_readcam

名前

gtk_fts1_readcam - カメラデータを読み込み、JPEG ファイルに出力する。

書式

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

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

引数

変数名	I/O	説明
file_id	in	GOSAT-1 FTS L1 プロダクトファイルの HDF ファイル識別子
num	in	カメラデータを取得する観測データのインデックス $1 \leq \text{num} \leq \text{numPoints}(*)$ (*) numPoints: 観測データ数

戻り値

値	説明
0	正常
0 以外	異常

説明

gtk_fts1_readcam は、指定した視野確認カメラデータを読み込み、カレントディレクトリの JPEG ファイルに出力する。保存する JPEG ファイル名は、以下の通り。

ファイル名: yymmddHHMMSSXXXX. jpg

サンプル

```
sample_fts1_readcam
```


6. 3 GOSAT-2 FTS-2 Level 1 プロダクト API

6. 3. 1 gtk_fts2_getigm

名前

gtk_fts2_getigm - インタフェログラムデータを取得する

書式

```
#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);
```

引数

変数名	I/O	説明
ndata	out	インタフェログラムのサンプル数を格納するポインタ
datax	out	各サンプルの光路差長 [cm]を格納するポインタ
datay	out	各サンプルのインタフェログラム [V]を格納するポインタ
file_id	in	GOSAT-2 FTS-2 LIA SWIR/TIR プロダクトファイルの HDF ファイル識別子
flag_in	in	出力するデータの指定方法 OBSNO : 観測点番号で指定する。 LATLON : 緯度・経度で指定する。
num	in	インタフェログラムを取得する観測データのインデックス。 flag_in が OBSNO の場合に有効 $1 \leq \text{num} \leq \text{numPoints} (*)$ (*) numPoints : 観測データ数
lat, lon	in	インタフェログラムデータを取得する観測点の緯度経度[°]。 flag_in が LATLON の場合に有効。 $-90 \leq \text{lat} \leq 90$ (degree) $-180 \leq \text{lon} \leq 180$ (degree)
band	in	出力するバンド $1 \leq \text{band} \leq 5$
ps	in	偏波 バンド 1~3 の場合 : P : P polarization mode S : S polarization mode バンド 4, 5 の場合 P

戻り値

値	説明
0	正常
0 以外	異常

説明

`gtk_fts2_getigm` は、指定した観測点のインターフェログラムデータを取得する。
観測点は、ファイル内のインデックス又は緯度経度で指定する。緯度経度を指定した場合は最も近い観測番号が選ばれる。

読み込んだデータは、新規に割り当てられたメモリアドレス `*datax`, `*datay` に格納される。

注意

アプリケーションは、利用しなくなった `*datax`, `*datay` を `free()` で解放すること。

サンプル

`sample_fts2_getigm`

6. 3. 2 gtk_fts2_getspc

名前

gtk_fts2_getspc - スペクトルデータを取得する

書式

```
#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);
```

引数

変数名	I/O	説明
ndata	out	スペクトルのサンプル数を格納するポインタ
datax	out	各サンプルの波数 [cm-1] を格納するポインタ
datay	out	各サンプルのスペクトル [V/cm-1]を格納するポインタ
file_id	in	GOSAT-2 FTS-2 L1B SWIR/TIR プロダクトファイルの HDF ファイル識別子
flag_in	in	OBSNO : 観測点番号で指定する。 LATLON : 緯度・経度で指定する。
num	in	スペクトルデータを取得する観測データのインデックス flag_in が OBSNO の場合に有効 $1 \leq \text{num} \leq \text{numPoints} (*)$ (*) numPoints: 観測データ数
lat, lon	in	スペクトルデータを取得する観測点の緯度経度[°]。 flag_in が LATLON の場合に有効。 $-90 \leq \text{lat} \leq 90$ (degree) $-180 \leq \text{lon} \leq 180$ (degree)
band	in	出力するバンド $1 \leq \text{band} \leq 5$
ps	in	偏波 バンド 1~3 の場合 : P : P polarization mode S : S polarization mode バンド 4,5 の場合 P

戻り値

値	説明
0	正常
0 以外	異常

説明

gtk_fts2_getspc は、指定した観測点のスペクトルデータを取得する。

観測点はファイル内のインデックス又は緯度経度で指定する。緯度経度を指定した場合は

最も近い観測番号が選ばれる。

読み込んだデータは、新規に割り当てられたメモリアドレス*datax, *datay に格納される。

注意

アプリケーションは、利用しなくなった*datax, *datay を free() で解放すること。

サンプル

sample_fts2_getspc

6. 3. 3 gtk_fts2_rad2bt

名前

gtk_fts2_rad2bt- FTS-2/TIR バンドの輝度情報から輝度温度[K]へ変換する。

書式

```
#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);
```

引数

変数名	I/O	説明
ndata	out	輝度温度のサンプル数を格納するポインタ
datax	out	各サンプルの波数 [cm-1] を格納するポインタ
datay	out	各サンプルの輝度温度 [K] を格納するポインタ
file_id	in	GOSAT-2 FTS-2 TIR プロダクトファイルの HDF ファイル ID
flag_in	in	OBSNO : 観測点番号で指定する。 LATLON : 緯度・経度で指定する。
num	in	輝度温度を取得する観測データのインデックス。 flag_in が OBSNO の場合に有効 $1 \leq \text{num} \leq \text{numPoints} (*)$ (* numPoints : 観測データ数
lat, lon	in	輝度温度を取得する出観測点の緯度経度[°]。 flag_in が LATLON の場合に有効。 $-90 \leq \text{lat} \leq 90$ (degree) $-180 \leq \text{lon} \leq 180$ (degree)
band	in	取得するバンド $4 \leq \text{band} \leq 5$

戻り値

値	説明
0	正常
0 以外	異常

説明

gtk_fts2_rad2bt は、指定した観測点の TIR バンドの輝度情報から輝度温度を算出する。観測点はファイル内のインデックス又は緯度経度で指定する。緯度経度を指定した場合は最も近い観測番号が選ばれる。

読み込んだデータは、新規に割り当てられたメモリアドレス*datax, *datay に格納される。

注意

アプリケーションは、利用しなくなった*datax, *datay を free() で解放すること。

サンプル
sample_fts2_rad2bt

6. 3. 4 gtk_fts2_readcam

名前

gtk_fts2_readcam - カメラデータを読み込み、JPEG ファイルに出力する

書式

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

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

引数

変数名	I/O	説明
file_id	in	GOSAT-2 FTS-2 共通 L1B プロダクトファイルの HDF ファイル識別子
num	in	カメラデータのインデックス $1 \leq \text{num} \leq \text{numPoints} (*)$ (*) numPoints: カメラデータ数

戻り値

値	説明
0	正常
0 以外	異常

説明

gtk_fts2_readcam は、指定した観測点番号の視野確認カメラデータを読み込み、カレントディレクトリに JPEG 画像の出力を行う。ファイル名は、以下の通り。

ファイル名: YYYY-MM-DDThh.mm.ss.ffffffZ.jpg

サンプル

```
sample_fts2_readcam
```

6. 4 GOSAT-1 CAI Level 1 プロダクト API

6. 4. 1 gtk_com_readstruct

名前

gtk_com_readstruct - CAI の時刻データを取得する。

書式

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

```
GTK_RET gtk_com_readstruct(int* ndims, hsize_t dims[], 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;
```

引数

変数名	I/O	説明
ndims	out	ディメンジョン数を格納するポインタ
dims	out	ディメンジョンの要素数を格納するポインタ
data	out	読み込んだ時刻データを格納するポインタ
file_id	in	GOSAT-1 CAI L1A プロダクトの hdf5 ファイル識別子
path	in	読み込むデータセットのパス

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_com_readstruct は、CAI の時刻データを取得する。

読み込んだデータは、新規に割り当てられたメモリアドレス*data に格納される。

注意

アプリケーションは、利用しなくなった*data を free() で解放すること

サンプル
sample_com

6. 4. 2 gtk_cail_geointerpol

名前

gtk_cail_geointerpol - 画素毎の緯度経度情報を取得する。(内挿)

書式

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

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

```
typedef struct{  
    double latitude_deg;  
    double longitude_deg;  
}GTK_GEO_DATA;
```

引数

変数名	I/O	説明
data_dims	out	取得した緯度経度情報のサイズを格納するポインタ。 data_dims[0]: ライン数 data_dims[1]: ピクセル数
data	out	補間後の緯度経度[°]を格納するポインタ
file_id	in	GOSAT-2 CAI L1A プロダクトの hdf5 ファイル識別子
band	in	緯度経度を補間するバンドを指定する。 $1 \leq \text{band} \leq 4$

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

CAI プロダクトは、間引きされた緯度経度のデータセットを格納している。

本関数は、そのデータセットを読み込み、すべてのライン/ピクセルの値を補間して求める。

補間したデータは、新規に割り当てられたメモリアドレス data に格納される。

ピクセル番号 x, ライン番号 y の緯度経度[°]は、(*data)[y*X+x]に格納される。ここで、X は、1 ラインの画素数 data_dims[1]である。

注意

アプリケーションは、利用しなくなった*data を free() を呼び出して解放すること。

サンプル

```
sample_cail_geointerpol
```

6. 4. 3 gtk_cail_dn2rad

名前

gtk_cail_dn2rad - CAI 観測データ (DN 値) を輝度に変換する

書式

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

```
int gtk_cail_dn2rad(int* data_dims, float** data, const hid_t file_id,  
                   const int band, const char* config_folder_name);
```

引数

変数名	I/O	説明
data_dims	out	変換した輝度データのサイズを格納するポインタ。 data_dims[0]: ライン数 data_dims[1]: ピクセル数
data	out	輝度データ [W/m ² /str/um] を格納するポインタ
file_id	in	GOSAT-1 CAI L1A プロダクトの hdf5 ファイル識別子
band	in	輝度を取得するバンド 1 ≤ band ≤ 4
config_folder_name	in	パラメータファイルを格納したフォルダ名 (例: "parameter/")

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_cail_dn2rad は、画像データ (DN 値) を読み込み、輝度に変換する。

変換したデータは、新規に割り当てられたメモリアドレス*data に格納される。

ピクセル番号 x, ライン番号 y の輝度データは、(*data)[y*X+x] に格納される。ここで、X は、1 ラインの画素数 data_dims[1] である。

注意

アプリケーションは、利用しなくなった*data を free() を呼び出して解放すること。

サンプル

```
sample_cail_dn2rad
```

6. 4. 4 gtk_cail_cutimage

名前

gtk_cail_cutimage - 指定範囲の画像データを取得する。

書式

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

```
int gtk_cail_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: 6. 4. 1 参照。

GTK_GEO_DATA : 6. 4. 2 参照

引数

変数名	I/O	説明
data_dims	out	取得したデータのサイズを格納するポインタ。 data_dims[0]: ライン数 data_dims[1]: ピクセル数
data	out	取得した画像データを格納するポインタ。 flag_out = DN の場合 : DN 値 flag_out = RAD の場合 : 輝度 [W/m2/str/um]
latlon	out	取得した緯度経度[°]を格納するポインタ。
time	out	取得した時間情報を格納するポインタ
file_id	in	GOSAT-1 CAI L1A プロダクトの hdf5 ファイル識別子
flag_in	in	取得するデータの条件 LNPX : Line and pixel mode LATLON : Latitude and Longitude mode
line_lat1, pixel_lon1 line_lat2, pixel_lon2	in	取得データの範囲指定 flag_in = LNPX の場合: 取得する範囲をライン数、ピクセル数で指定する。 $1 \leq \text{band} \leq 3$ $1 \leq \text{line_lat1} < \text{line_lat2} \leq \text{lines}$ $1 \leq \text{line_lon1} < \text{line_lon2} \leq 2056$ band=4 $1 \leq \text{line_lat1} < \text{line_lat2} \leq \text{lines}$ $1 \leq \text{line_lon1} < \text{line_lon2} \leq 512$ flag_in = LATLON の場合: 取得する範囲を緯度、経度で指定する[°]。 $-90 \leq \text{line_lat1} < \text{line_lat2} \leq 90$ $-180 \leq \text{line_lon1} < \text{line_lon2} \leq 180$
band	in	輝度を取得するバンド $1 \leq \text{band} \leq 4$

flag_out	in	出力データの指定 DN : digital number mode RAD : Radiance mode
config_folder_name	in	センサパラメータを格納したフォルダ名 (例: "parameter/")

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_cai1_cutimage は、指定した範囲のデータを取得する。

取得する範囲はピクセル・ライン又は緯度経度で指定する。緯度経度を指定した場合は最も近いピクセル・ラインが選ばれる。

取得したデータは、新規に割り当てられたメモリアドレス*data, *latlon and *time に格納される。

ピクセル番号 x, ライン番号 y の画像データは、(*data)[y*X+x]に格納される。ここで、Xは、1ラインの画素数 data_dims[1]である。

ピクセル番号 x, ライン番号 y の緯度経度は、(*latlon)[y*X+x]に格納される。ここで、Xは、1ラインの画素数 data_dims[1]である。

ライン番号 y の時刻データは、(*time)[y]に格納される。

注意

アプリケーションは、利用しなくなった*data, *latlon, *time を free() で解放すること。

サンプル

sample_cai1_cutimage

6. 5 GOSAT-1 CAI-2 Level 1 プロダクト API

6. 5. 1 gtk_cai2_geointerpol

名前

gtk_cai2_geointerpol - 画素毎の緯度経度情報を内挿で取得する。(内挿)

書式

```
#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;
```

引数

変数名	I/O	説明
data_dims	out	補間した緯度経度のサイズを格納するポインタ。 data_dims[0]: ライン数 data_dims[1]: ピクセル数
data	out	補間後の緯度経度[°]を格納するポインタ
file_id	in	GOSAT-2 CAI-2 L1A 前方視/後方視プロダクトの hdf5 ファイル識別子

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

CAI-2 プロダクトは、間引きされた緯度経度のデータセットを格納している。

gtk_cai2_geointerpol は、データセットを読み込み、すべてのライン/ピクセルの値を補間して求める。

補間したデータは、新規に割り当てられたメモリアドレス*data に格納される。

ピクセル番号 x, ライン番号 y の緯度経度は、(*data)[y*X+x] に格納される。ここで、X は、1 ラインの画素数 data_dims[1] である。

注意

アプリケーションは、利用しなくなった data を free() を呼び出して解放すること。

サンプル

```
sample_cai2_geointerpol
```

6. 5. 2 gtk_cai2_dn2rad

名前

gtk_cai2_dn2rad - CAI-2 観測データ (DN 値) を輝度に変換する。

書式

```
#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 );
```

引数

変数名	I/O	説明
com_file_id	in	GOSAT-2 CAI-2 L1A 共通プロダクトの hdf5 ファイル識別子
band_file_id	in	GOSAT-2 CAI-2 L1A 前方視または後方視プロダクトの hdf5 ファイル識別子
config_folder_name	in	パラメータファイルを格納したフォルダ名 (例: "parameter/")
band	in	画像データを取得するバンド。 $1 \leq \text{band} \leq 5$ (forward) $6 \leq \text{band} \leq 10$ (backward)
start_pixel, end_pixel start_line, end_line	in	取得データの範囲 band=1, 2, 3, 4, 6, 7, 8, 9 の場合: $1 \leq \text{start_pixel} < \text{end_pixel} \leq 2056$ $1 \leq \text{start_line} < \text{end_line} \leq \text{lines}_{500}$ band = 5, 10 の場合: $1 \leq \text{start_pixel} < \text{end_pixel} \leq 1024$ $1 \leq \text{start_line} < \text{end_line} \leq \text{lines}_{1km}$
corr_image	out	取得した輝度 [$\text{W}/\text{m}^2/\mu\text{m}/\text{str}$]
corr_image_num_pixels_per_line	in	corr_image のピクセル数
corr_image_num_lines	in	corr_image のライン数

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

`gtk_cai2_dn2rad` は、画像データ (DN 値) を輝度に変換する。

変換した輝度は、アプリケーションの指定するアドレス `corr_image` に出力する。

ピクセル番号 `x`, ライン番号 `y` の輝度データは、`corr_image[y*X+x]` に格納される。ここで、`X` は、1 ラインの画素数 `corr_image_num_pixels_per_line` である。

アプリケーションは、`corr_image` に、`sizeof(float) × corr_image_num_pixels_per_line × corr_image_num_lines` のメモリを確保し、管理すること。

サンプル

`sample_cai2_dn2rad`

6. 5. 3 gtk_cai2_cutimage

名前

gtk_cai2_cutimage - 指定範囲の画像データを取得する。

書式

```
#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 :6. 5. 1 参照

引数

変数名	I/O	説明
data_dims	out	取得したデータサイズを格納するポインタ。 data_dims[0]: ライン数 data_dims[1]: ピクセル数
data	out	取得した画像データを格納するポインタ。 flag_out = DN の場合 : DN 値 flag_out = RAD の場合 : 輝度 [W/m2/str/um]
latlon	out	取得した緯度経度[°]を格納するポインタ
time	out	取得した時間情報を格納するポインタ
file_id	in	GOSAT-2 CAI-2 前方視または後方視プロダクトの hdf5 ファイル識別子
file_id_com	in	GOSAT-2 CAI-2 共通 L1A プロダクトの hdf5 ファイル識別子
flag_in	in	取得するデータの条件 LNPX : Line and pixel mode LATLON : Latitude and Longitude mode
line_lat1, pixel_lon1 line_lat2, pixel_lon2	in	取得データの範囲 flag_in = LNPX の場合: 取得する範囲をライン数、ピクセル数で指定する。 band=1, 2, 3, 4, 6, 7, 8, 9 の場合: 1 ≤ line_lat1 < line_lat2 ≤ lines_500 1 ≤ pixel_lon1 < pixel_lon2 ≤ 2056 band = 5, 10 の場合: 1 ≤ line_lat1 < line_lat2 ≤ lines_1km 1 ≤ pixel_lon1 < pixel_lon2 ≤ 1024 flag_in = LATLON の場合: 取得する範囲を緯度、経度で指定する。 -90 ≤ line_lat1 < line_lat2 ≤ 90

		$-180 \leq \text{pixel_lon1} < \text{pixel_lon2} \leq 180$
band	in	輝度を取得するバンド $1 \leq \text{band} \leq 10$
flag_out	in	出力データを指定する。 DN : digital number mode RAD : Radiance mode
config_folder_name	in	パラメータファイルを格納したフォルダ名 (例: "parameter/")

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

`gtk_cai2_cutimage` は、指定した範囲の画像データを取得する。

取得する範囲はピクセル・ライン又は緯度経度で指定する。緯度経度を指定した場合は最も近いピクセル・ラインが選ばれる。

取得したデータは、新規に割り当てられたメモリアドレス `*data`, `*latlon` and `*time` に格納される。

ピクセル番号 `x`, ライン番号 `y` の画像データは、`(*data)[y*X+x]` に格納される。ここで、`X` は、1 ラインの画素数 `data_dims[1]` である。

ピクセル番号 `x`, ライン番号 `y` の緯度経度は、`(*latlon)[y*X+x]` に格納される。ここで、`X` は、1 ラインの画素数 `data_dims[1]` である。

ライン番号 `y` の時刻データは、`(*time)[y]` に格納される。

注意

アプリケーションは、利用しなくなった `*data`, `*latlon`, `*time` を `free()` で解放すること。

サンプル

`sample_cai2_cutimage`

6. 5. 4 gtk_cai2_geocalc

名前

gtk_cai2_geocalc -緯度経度情報の算出を行う。

書式

```
#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;
```

引数

変数名	I/O	説明
com_file_id	in	GOSAT-2 CAI-2 L1A 共通プロダクトの hdf5 ファイル識別子
config_folder_name	in	パラメータファイルを格納したフォルダ名 (例: "parameter/")
band	in	画像データを取得するバンドを指定する。 1 ≤ band ≤ 5 (forward) 6 ≤ band ≤ 10 (backward)
start_pixel, end_pixel start_line, end_line	in	取得データの範囲 band=1, 2, 3, 4, 6, 7, 8, 9 の場合: 1 ≤ start_pixel < end_pixel ≤ 2056 1 ≤ start_line < end_line ≤ lines_500 band = 5, 10 の場合: 1 ≤ start_pixel < end_pixel ≤ 1024 1 ≤ start_line < end_line ≤ lines_1km
geo_data	out	緯度経度[°]を格納するポインタ
geo_data_num_pixels_per_line	in	geo_data のピクセル数
geo_data_num_lines	in	geo_data のライン数

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

`gtk_cai2_geocalc` は、指定した範囲の緯度経度情報の算出を行う。

算出した緯度経度は、アプリケーションが指定するアドレス `geo_data` に出力する。

ピクセル番号 `x`、ライン番号 `y` の緯度経度は、`geo_data [y*X+x]` に格納される。ここで、`X` は、1 ラインの画素数 `geo_data_num_pixels_per_line` である。

アプリケーションは、`geo_data` に `sizeof(CAI2_GeoData) × geo_data_num_pixels_per_line × geo_data_num_lines` のメモリを確保し、管理すること。

サンプル

```
gtk_cai2_geocalc
```

6. 5. 5 gtk_cai2_resampimage

名前

gtk_cai2_resampimage

- バンド間レジストレーションによりリサンプルした画像を取得する

書式

```
#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 );
```

引数

変数名	I/O	説明
com_file_id	in	GOSAT-2 CAI-2 L1A 共通プロダクトの hdf5 ファイル識別子
band_file_id	in	GOSAT-2 CAI-2 L1A 前方視または後方視プロダクトの hdf5 ファイル識別子
config_folder_name	in	センサパラメータを格納したフォルダ名 (例: "parameter/")
band	in	画像データを取得するバンド $1 \leq \text{band} \leq 5$ (forward) $6 \leq \text{band} \leq 10$ (backward)
start_pixel, end_pixel start_line, end_line	in	取得データの範囲 band=1, 2, 3, 4, 6, 7, 8, 9 の場合: $1 \leq \text{start_pixel} < \text{end_pixel} \leq 2056$ $1 \leq \text{start_line} < \text{end_line} \leq \text{lines_500}$ band = 5, 10 の場合: $1 \leq \text{start_pixel} < \text{end_pixel} \leq 1024$ $1 \leq \text{start_line} < \text{end_line} \leq \text{lines_1km}$
corr_image	out	radio_metric_correction が CAI2_RADIO_METRIC_CORR_OFF の場合、 DN 値 radio_metric_correction が

		CAI2_RADIO_METRIC_CORR_ON の場合、 輝度 [W/m ² /μm/str]
corr_image_num_pixels_per_line	in	corr_image のピクセル数
corr_image_num_lines	in	corr_image のライン数
radio_metric_correction	in	出力画素値の指定 CAI2_RADIO_METRIC_CORR_OFF:DN 値 CAI2_RADIO_METRIC_CORR_ON:輝度値

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_cai2_resampimage は、指定した条件の画像データを基準バンドに重畳するようにリサンプリングする。リサンプリングした画像は、アプリケーションが指定するアドレス `corr_image` に出力する。

ピクセル番号 `x`, ライン番号 `y` の輝度データは、`corr_image[y*X+x]` に格納される。ここで、`X` は、1 ラインの画素数 `corr_image_num_pixels_per_line` である。

`corr_image` には、`sizeof(float) × corr_image_num_pixels_per_line × corr_image_num_lines` のメモリが確保されていること。

サンプル

`sample_cai2_resampimage`

6. 5. 6 gtk_cai2_create_sensor_param

名前

gtk_cai2_create_sensor_param - センサパラメータ読み込み、インスタンスを生成する

書式

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

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

引数

変数名	I/O	説明
config_folder	in	パラメータファイルを格納したフォルダ

戻り値

値	説明
NULL (0) 以外	読み込まれたセンサパラメータ
NULL (0)	異常終了

説明

gtk_cai2_create_sensor_param は、指定したフォルダから、パラメータファイルを読み込み、センサパラメータのインスタンスを生成する。

注意

アプリケーションは、利用しなくなったパラメータを gtk_cai2_delete_sensor_param() で解放すること。

サンプル

```
sample_cai2_create_processed_image
```

6. 5. 7 gtk_cai2_delete_sensor_param

名前

gtk_cai2_delete_sensor_param -センサパラメータのインスタンスを破棄する。

書式

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

```
void gtk_cai2_delete_sensor_param(CAI2_SensorParameter* sns_parameter);
```

引数

変数名	I/O	説明
sns_parameter	in	破棄するセンサパラメータ

戻り値

なし

説明

gtk_cai2_delete_sensor_param は、gtk_cai2_create_sensor_param で読み込んだセンサパラメータを破棄する。

サンプル

```
sample_cai2_create_processed_image
```


6. 5. 8 gtk_cai2_create_processed_image

名前

gtk_cai2_create_processed_image - 画像処理済データを生成する。

書式

```
#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; /* バンド(1~10) */  
    float* data; /* 輝度[W/m2/μm/str] */  
    char* flg; /* 画素値を識別するフラグ*/  
    int pixels_per_line; /* 1ラインあたりのピクセル数 */  
    int lines; /* ライン数 */  
    unsigned int image_processing; /* 適用された画像処理を識別するフラグ*/  
}CAI2_IMAGE;
```

引数

変数名	I/O	説明
com_file_id	in	GOSAT-2 CAI-2 L1A 共通プロダクトの hdf5 ファイル識別子
fwd_band_file_id	in	GOSAT-2 CAI-2 L1A 前方視プロダクトの hdf5 ファイル識別子
bwd_band_file_id	in	GOSAT-2 CAI-2 L1A 後方視プロダクトの hdf5 ファイル識別子
sns_parameter	in	センサパラメータ
image	out	画像データ(バンド 1~10)を格納するポインタ

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_cai2_create_processed_image は、共通プロダクト、前方視・後方視プロダクトを読み込み、画像データを出力する。

各バンドの観測データは、DN 値から輝度へ変換され、迷光補正処理後、センサパラメータで指定された基準バンドに対して位置合わせされる。
取得したバンド 1～バンド 10 のデータは、image[0]～image[9]に順番に格納される。

画像データ image[i]には、以下の値が格納される。

- band: バンド 1～10 の値が格納される。
- data: 輝度値 ($\text{W/m}^2/\mu\text{m/str}$) が格納される。
pixels_per_line×lines のサイズで API が確保する配列である。
画素番号 x, ライン y の画素値は、data[y* pixels_per_line +x]に格納される。
出力する観測データが 1 ラインも存在しない場合、NULL が格納される。
基準バンドのデータが存在しない場合、NULL が格納される。
- flg: 画素値の属性が格納される。
pixels_per_line×lines のサイズで API が確保する配列である。
画素番号 x, ライン y の画素値の属性は、flg [y* pixels_per_line +x]に格納される。
flg[n]には、以下の値が格納される。
0 : normal
1 : no data. 欠損、位置合わせによる対応画素なし
2 : out of range. 処理対象外
4 : saturated (corrected) 飽和補正された
出力する観測データが 1 ラインも存在しない場合、NULL が格納される。
基準バンドのデータが存在しない場合、NULL が格納される。
- pixels_per_line: 画素数が格納される。
前方視・後方視の基準バンドの画素数と同じ値となる。
基準バンドのデータが存在しない場合、0 が格納される。
出力する観測データが 1 ラインも存在しない場合、0 が格納される。
- lines: ライン数が格納される。
出力する観測データが 1 ラインも存在しない場合、0 が格納される。
前方視・後方視の基準バンドのライン数と同じとなる。
基準バンドのデータが存在しない場合、0 が格納される。

image_processing: 実際に適用された画像処理を識別するフラグが組み合わせで設定される。

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

例: Band1/6 で全ての画像処理が行われた場合、以下の値となる。

```
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
```

画像処理の実施は、パラメータファイルで設定される。
詳細は、取扱説明書の『パラメータ編』を参照。

出力される画像のサイズは、基準バンドのサイズと同じである。

flg[n]は、data[n]の画像が欠損している場合のみ1となる。画像処理において参照する画像が欠損していた場合においても、data[n]の画像が欠損しておらず処理された場合は、flg[n]に0または2が格納される。

data[n]に画像処理できない画素値があった場合、flg[n]には2が設定される。
例えば、バンド1/6のクロストーク補正では、前方視・後方視の両方の観測が必要となる。前方視・後方視の観測時刻の一方しか存在しないラインでは、flg[n]=2となる。この場合、その画素へのバンド1/6クロストーク補正はスキップされ、data[n]は、その画像処理なしで処理が進められる。

fwd_band_file_idおよび,bwd_band_file_idに格納された利用可能なバンドの観測データを全て処理する。1バンドも利用可能なバンドが存在しない場合は、異常終了する。

注意

アプリケーションは、利用しなくなった画像データを、gtk_cai2_delete_image で解放すること。

サンプル

sample_cai2_create_processed_image

6. 5. 9 gtk_cai2_delete_image

名前

gtk_cai2_delete_image -画像データを破棄する。

書式

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

```
void gtk_cai2_delete_image( CAI2_IMAGE image[10] );
```

CAI2_IMAGE : gtk_cai2_create_prcessed_image を参照。

引数

変数名	I/O	説明
image	in	破棄する画像データ

戻り値

なし

説明

gtk_cai2_delete_image は、gtk_cai2_create_processed_image で生成した、画像データを破棄する。

バンド n(n=1~10)に画像データが存在しない場合、image[n-1].data, image[n-1].flg には NULL を指定すること。

画像データを破棄した後は、image の全てのデータは 0 に初期化される。

サンプル

```
sample_cai2_create_processed_image
```

6. 5. 10 gtk_cai2_create_resample_info

名前

gtk_cai2_create_resample_info

- バンド間レジストレーション情報を生成する。

書式

```
#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* /* バンド間レジストレーション情報 */
```

引数

変数名	I/O	説明
fwd_band_file_id	in	GOSAT-2 CAI-2 L1A 前方視プロダクトの hdf5 ファイル識別子
bwd_band_file_id	in	GOSAT-2 CAI-2 L1A 後方視プロダクトの hdf5 ファイル識別子
sns_parameter	in	センサパラメータ

戻り値

値	説明
NULL (0) 以外	バンド間レジストレーション情報のインスタンス
NULL (0)	異常終了

説明

gtk_cai2_create_resample_info は、バンド間レジストレーションのための情報を取得する。

gtk_cai2_create_resample_info は、ライン方向の位置合わせを行う情報として、引数で指定した前方視および後方視のプロダクトから各バンドの観測時刻を取得する。fwd_band_file_id、bwd_band_file_id の何れかまたは両方が有効な場合、有効なファイル識別子の処理を行い正常終了する。両方の fwd_band_file_id、bwd_band_file_id が有効値でない場合は、異常終了する。

注意

アプリケーションは、利用しなくなった CAI2_RESAMPLE_INFO を gtk_cai2_delete_resample_info () で解放すること。

サンプル

```
sample_cai2_get_resample_position
```

```
6. 5. 11 gtk_cai2_get_resample_position
```

名前

```
gtk_cai2_get_resample_position
```

-基準バンドの画素番号・ライン番号に該当する参照画像の画素番号・ライン番号を取得する。

書式

```
#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)
```

引数

変数名	I/O	説明
baseband_pixel_no	in	基準バンドの画素番号(1オリジン)
baseband_line_no	in	基準バンドのライン番号(1オリジン)
refband	in	参照位置を取得するバンド番号(1~10)
refband_pixel_no	out	参照バンドの画素番号(1オリジン)を格納するポインタ
refband_line_no	out	参照バンドのライン番号(1オリジン)を格納するポインタ
resample_info	in	バンド間レジストレーション情報

戻り値

値	説明
0	正常終了
0 以外	異常終了

説明

gtk_cai2_get_resample_position は、引数で指定した基準バンドの `baseband_pixel_no`, `baseband_line_no` に該当する、参照バンドの位置情報を取得する。取得するデータは、以下の通り。

- ・ `refband_pixel_no`: 基準バンドの画素番号 `baseband_pixel_no` に対応する、参照バンドの画素番号(1オリジン)。
求めた値が参照画像の範囲外となる場合、`refband_pixel_no` に

は 0 が設定される

- `refband_line_no`: 基準バンドの画素番号 `baseband_pixel_no` に対応する、参照バンドのライン番号(1 オリジン)。求めた値が参照画像の範囲外となる場合、`refband_line_no` には 0 が設定される。。

`resample_info` の生成については、`gtk_cai2_create_resample_info` を参照。バンド番号、画素番号の範囲が異常である場合、および、`resample_info` に変換対象のバンドの情報が読み込まれていない場合は、異常終了する。

サンプル

```
sample_cai2_get_resample_position
```


6. 5. 12 gtk_cai2_delete_resample_info

名前

gtk_cai2_delete_resample_info

-バンド間レジストレーション情報を破棄する。

書式

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

```
void gtk_cai2_delete_resample_info(CAI2_RESAMPLE_INFO* resample_info)
```

引数

変数名	I/O	説明
resample_info	in	バンド間レジストレーション情報

戻り値

なし

説明

gtk_cai2_create_resample_info で生成した、バンド間レジストレーション情報を破棄する。

サンプル

```
sample_cai2_get_resample_position
```

(白紙)