

AMSR Data Input Toolkit (ADIT)

User's Guide

(日本語版)

Version 3.00

H Edition: Nov. 16, 2012

Japan Aerospace Exploration Agency (JAXA)

 **MITSUBISHI SPACE SOFTWARE CO., LTD.**

Change Record Page

Document Title: AMSR Data Input Toolkit (ADIT) User's Guide			
Document Date: June 26, 2002			
Issue	Date	Page Affected	Description
Original	26/6/2002	All	Baseline
A	20/12/2002	p1-1	表 1.2-1 に LINUX の確認環境を追加した。
		P2-4	今回のバージョンでは、LINUX の C プログラムについて対応しているため、ADIT を解凍展開後のディレクトリ構成に、LINUX 用のメイクファイルを追加した。
		"	今回のバージョンでは、SCAN_TIME を UNIX SystemTime(UTC)に変換する際に閏秒補正を行う機能を追加した。これに伴い、ADIT を解凍展開後のディレクトリ構成に、閏秒補正時に使用する閏秒情報ファイルの格納先ディレクトリを追加した。
		P2-5	閏秒補正に関して、環境設定手順を記述した。
		P3-3	C プログラムのコンパイルに関し、LINUX での利用例を記述した。
		P3-4	"
		p4-18	SPC_temp_calc 及び SPS_temp_calc を 4byte real から 8byte real へ変更した。
		P4-24	L2 の品質フラグを全面的に見直した。
		P4-29	Table 4.4.1-1 に相対レジストレーションの補正係数を追記した。
B	28/5/2003	p5-1~4	サンプルプログラムを追加した。
C	24/2/2004	p4-24,25	レベル 2 プロダクトの品質フラグを変更した。
		P4-30	Table 4.4.1-1 に次の 4 つのメタデータを追加した。 <ul style="list-style-type: none"> ・ CalibrationMethod ・ HTSCorrectionParameterVersion ・ SpillOverParameterVersion ・ MoonLightEffectParameterVersion
D	1/5/2005	p2-4	ADIT のインストール方法を v1.07 に対応した。
		P3-1,2 p4-1,3,4,6,7	89GHz 低周波相当データ算出関数についての記述を追加した。
		P4-15	Table 4.3.4-1 に次の項目の変更を対応させた <ul style="list-style-type: none"> ・ pos_orbit
		p 5-4	サンプルプログラムの追記及びサンプルデータを変更した。
E	5/3/2008	P4-35,36	指摘事項修正。
F	2/3/2010	P1-1	表 1.2-1 の HP 及び DEC の項目を削除した。

Issue	Date	Page Affected	Description
		P2-4	ADIT のインストール方法を v2.02 に対応した。
		P3-4, P3-9	DEC 及び HP の項目を削除
		P3-13~17	F77 プログラミング(Linux 版)を追記
G	26/9/2011	P3-1~P3-4, P3-8, P3-14, P3-15, P4-1~P4-10, P5-3~P5-4	複数走査読み込み関数を追記
H	16/11/2012	P1-1	表 1.2-1 の試験実施環境を更新
		P2-4	ADIT のインストール方法を v3.00 に対応した。

目次

1	HDFライブラリとADIT	1-1
1.1	HDFとは?	1-1
1.2	ADITとは?	1-1
2	HDFライブラリとADITのインストール	2-1
2.1	HDFライブラリのインストール.....	2-1
2.1.1	HDF4.2r5 (バイナリ) のインストール.....	2-1
2.1.2	HDF4.2r5 (ソース) のインストール.....	2-2
2.2	ADITのインストール.....	2-4
2.3	ADITの環境設定.....	2-5
3	ADITを利用したプログラミング	3-1
3.1	プログラムの流れ	3-1
3.2	Cプログラミング	3-3
3.2.1	プログラムの記述.....	3-3
3.2.2	コンパイル	3-4
3.2.3	Cサンプルプログラム	3-5
3.3	Fortranプログラミング (SunOS版、SGI版).....	3-8
3.3.1	プログラムの記述.....	3-8
3.3.2	コンパイル	3-10
3.3.3	Fortranサンプルプログラム	3-10
3.4	Fortranプログラミング (Linux版)	3-14
3.4.1	プログラムの記述.....	3-14
3.4.2	コンパイル	3-15
3.4.3	Fortranサンプルプログラム	3-17
4	APPENDIX	4-1
4.1	Routines defined in ADIT.....	4-1
4.2	User routine interface in ADIT.....	4-3
4.3	Structure definition in ADIT.....	4-11
4.3.1	L1B, L2 common structure.....	4-12
4.3.2	AMSRL1B_SWATH (for L1B)	4-12
4.3.3	SUN_EARTH (for L1B)	4-15
4.3.4	STATUS_L1B (for L1B).....	4-18
4.3.5	CAL (for L1B).....	4-20
4.3.6	NAVI (for L1B)	4-26
4.3.7	AMSRL2_SWATH (for L2)	4-26
4.3.8	STATUS_L2 (for L2)	4-28

4.3.9	L3 Science data	4-31
4.4	Metadata	4-32
4.4.1	L1B Metadata	4-32
4.4.2	L2 Metadata	4-37
4.4.3	L3 Metadata	4-38
5	サンプルプログラム一覧	5-1

1 HDFライブラリとADIT

1.1 HDFとは

AMSR/AMSR-E データは、National Center for Supercomputing Applications (NCSA) で開発された HDF (Hierarchical Data Format) フォーマットで記録されています。HDF フォーマットのデータを C や Fortran のプログラムで読むためには、NCSA でフリー配布している HDF ライブラリを利用する計算機環境にあらかじめインストールしておく必要があります。HDF ライブラリのインストール方法については、2 章で説明します。

1.2 ADITとは

AMSR/AMSR-E データを C や Fortran のプログラムで読み込んで処理をする場合は、HDF ライブラリを直接使用するか、AMSR/AMSR-E データ用に開発された AMSR Data Input Toolkit (ADIT) を使うのが便利です。

ADIT は、ADIT 内で定義されている構造体にスキャン毎の AMSR/AMSR-E データを格納します。格納されているデータに変換係数が設定されている場合は、変換係数を使用して変換処理を行った結果を格納します。また、Quality 情報など複雑に格納されているデータに関しても、容易に読み出しができるように変換を施した上で格納します。ADIT の動作確認済みの試験環境を表 1.2-1 に示します。

表 1.2-1ADIT 動作確認環境

OS	OS version	C compiler	FORTTRAN compiler	HDF version
SunOS	Solaris 10	Oracle Solaris Studio 12.2 cc	Oracle Solaris Studio 12.2 f77	4.2r5
LINUX	2.6.18-194.el5	gcc-4.1.2	GNU:gfortran 4.0.1 PGI:pgf90 6.1-9 Intel:ifort 11.1	4.2r5

2 HDFライブラリとADITのインストール

ここでは、HDF ライブラリのインストール方法について説明します。AMSR/AMSR-E データは HDF ライブラリバージョン 4.2r5 を使用して作成されています。従って、使用する HDF ライブラリはバージョン 4.2r5 の使用を推奨します。

ここでは、HDF ライブラリバージョン 4.2r5 のインストールについて説明することになります。

2.1 HDFライブラリのインストール

HDF ライブラリのインストール方法は、HDF4.2r5 のコンパイル済みのバイナリデータをインストールする方法と、HDF ライブラリのソースコードをコンパイルする方法の 2通りがあります。

2.1.1 HDF4.2r5 (バイナリ) のインストール

hdfgroup のサイトから、自分の計算機環境および OS のバージョンに対応した HDF4.2r5 のコンパイル済みファイルをダウンロードします。HDF ライブラリのインストールに際し、個人で使う場合にはスーパーユーザーで操作する必要はありません。所望のディレクトリ下にインストールし、自分で作成したプログラムのコンパイル時に参照リンクを張れば使うことができ、他の人と共有したい場合は、ファイルを展開後、展開したディレクトリ構造ごと、マシンの管理者に root の下のディレクトリにコピーしてもらおうとよいでしょう。

Web ツール (Netscape, Internet Explorer など) を使用してダウンロードする場合には、下記の URL を入力します。

```
ftp:// ftp.hdfgroup.org/HDF/HDF_Current/bin
```

自分の計算機環境および OS のバージョンに対応したディレクトリに移動し、インストール先のディレクトリに目的のファイルをダウンロードします。

ftp が終了したら、自分のマシンのインストール先ディレクトリ下で、ダウンロードしたファイルの解凍展開を 2 段階で行います (この例では、SGI の IRIX 用のファイルを解凍展開しています)。

```
% gunzip 4.2r5-irix64-n32.tar.gz
% tar xvf 4.2r5-irix64-n32.tar
```

解凍展開が完了すると、カレントディレクトリの下に、4.2r5-irix64-n32/ というディレクトリができます。この時のディレクトリの構造は、

4.2r5-irix64-n32/	COPYING	Copyright
	README	簡単な使い方
	bin/	HDF のユーティリティ (ツール) のディレクトリ

include/	インクルードファイルのディレクトリ
lib/	ライブラリのディレクトリ
man/	ツールのマニュアルのディレクトリ
release_notes/	HDF ライブラリの説明のディレクトリ
sample/C	サンプルプログラム (C プログラム用)
sample/Fortran	サンプルプログラム (Fortran プログラム用)

となっています。これでインストールは完了です。

2.1.2 HDF4.2r5 (ソース) のインストール

コンパイル済みの HDF4.2r5 が正常に動作しない場合には、HDF4.2r5 のソースをコンパイルして使用するのが良いでしょう。

HDF ライブラリのコンパイルには、ANSI C のコンパイラが必要です。ANSI C のコンパイラが使えない場合には、フリーで配布されている GNUANSI コンパイラ gcc を使用してください。HDF4.2r5 ライブラリのソースコードを Web ツール (Netscape, InternetExplorer など) を使用してダウンロードする場合には、下記の URL を入力します。

```
ftp:// ftp.hdfgroup.org/HDF/HDF_Current/bin
```

インストール先のディレクトリに HDF4.2r5.tar.gz をダウンロードした後、ファイルを以下の操作により解凍展開します。

```
% gunzip HDF4.2r5.tar.gz
% tar xvf HDF4.2r5.tar
```

解凍展開が完了すると、カレントディレクトリの下に HDF4.2r5/ というディレクトリができます。この下のディレクトリ構造は、以下のようになります。

HDF4.2r5/	COPYING	Copyright
	INSTALL	インストール手順の説明
	MAKEVMS.COM*	
	Makefile.in	
	README	ディレクトリの説明など
	Win32.nofortran.zip*	
	Win32.zip*	
	config/	マシン毎の Makefile のディレクトリ
	configure*	マシン毎の Makefile を作る configure
	configure.in	
	hdf/	HDF のソースコードのディレクトリ
	install-sh*	
	lib/	
	man/	HDF のマニュアルのディレクトリ
	mfhdf/	netCDF のディレクトリ
	mkinstalldirs	
	move-if-change*	
	release_notes/	HDF ライブラリの説明のディレクトリ

HDF ファイルをコンパイルする前に、configure を実行し、インストールする計算機環

境のチェックや、インストール先のディレクトリを設定を行います。`configure` をそのまま実行すると、デフォルトではインストール先が `/usr/local` です。その下のディレクトリに、`HDF ライブラリ/usr/local/lib`、`ユーティリティ/usr/local/bin`、`マニュアル/usr/local/man`、`インクルードファイル/usr/local/include` が作成されます。同名のディレクトリには上書きする可能性がありますのでご注意ください。

ここでは、自分のホームディレクトリ (`/home/amsr/work`) 下の `HDF4.2r5/` にインストールする例を示します。以下のように `configure` を実行します。

```
% ./configure -v --prefix=/home/amsr/work/HDF4.2r5
```

`configure` が実行されると、計算機環境に合った `Makefile` が自動的に作成されます。また、`--prefix=` で設定したディレクトリがインストール先のディレクトリになります。これでコンパイルの準備は完了です。

`HDF ライブラリ` のコンパイルを以下のようにして行います。

```
% make
```

コンパイルには少し時間がかかります。`ライブラリ` のコンパイルが正常に終了したことを確認する場合は、コンパイル終了後に、

```
% make test
```

とすると、一通りのテストが行えます。結果が標準出力に出力されますので、

```
% make test >& make.test.out
```

として標準出力される情報をファイルにセーブして確認すると便利です。上のファイルの内容を確認して問題ないようでしたら、

```
% make install
```

として、`configure` 実行時に設定したディレクトリに `ライブラリ` をインストールしてください。これでインストールは完了です。

2.2 ADITのインストール

ここでは、インストール方法について ADITv3.00 の例で説明します。ADIT は、EORC のサイトで入手できます。EORC のサイトの URL を以下に示します。

```
ftp:// ftp.hdfgroup.org/HDF/HDF_Current/bin
```

ADITv3.00 のファイルをダウンロードした後、インストール先のディレクトリにて、以下の操作によりファイルを解凍展開します。

```
% gunzip ADITv3.00.tar.gz
% tar xvf ADITv3.00.tar
```

解凍展開が完了すると、カレントディレクトリに ADITv3.00/ というディレクトリができます。この時のディレクトリの構造は、

ADITv3.00/	Makefile.SGI	SGI に対応する ADIT ライブラリ作成用 Makefile
	Makefile.SunOS	SUN に対応する ADIT ライブラリ作成用 Makefile
	Makefile.LINUX	LINUX に対応する ADIT ライブラリ作成用 Makefile
	allmake	マシンに対応した Makefile を実行するシェルスクリプト
	install*	ADIT のインストーラ
	include/	ADIT のインクルードファイルのディレクトリ
	lib/	ADIT ライブラリのインストール先ディレクトリ
	src/	ADIT のソースコードのディレクトリ
	etc/	閏秒 (TAI-UTC) 情報ファイルの格納ディレクトリ
	sample/	ADIT を利用したサンプルプログラムのディレクトリ

のようになっています。

次に、以下のように、ディレクトリ ADITv3.00/ に移動した後、ADIT のインストーラを実行します。

```
% cd ADITv3.00/
% ./install
```

ADIT のインストーラ install を実行すると、以下のようなメッセージが出てきますので、①～④でインストールの設定入力操作を行います。

```
% ./install
### Start installing AMSR Data Input Toolkit (Ver 3.00) ###
Input the directory of ADIT. (/home/amsr/work/ADITv3.00) ==> ①
Input the directory of included files of the HDF library. ==>/home/amsr/work/HDF4.1r2/include ②
Input the directory of library files of the HDF library. ==>/home/amsr/work/HDF4.1r2/lib ③
Input the directory storing a library of ADIT. (/home/amsr/work/ADITv3.00/lib) ==> ④

(コンパイル中のメッセージ)

### Finished installing ADIT. ###
### Created a library of ADIT. (/home/amsr/work//ADITv3.00/lib/libADIT.a) ###
*** Press Enter ==>
%
```

①～④では、以下の内容を設定することになります。

① インストールディレクトリの設定

ADIT 一式が格納されているディレクトリを指定する。括弧内は、デフォルトディレクトリで、インストーラが実行されたディレクトリである。デフォルトディレクトリでよければエンターを入力し、変更する場合、ここで、ディレクトリを入力する。

- ② HDF ライブラリのインクルードファイル格納ディレクトリの設定
HDF ライブラリのインクルードファイルが格納されているディレクトリを指定する。
- ③ HDF ライブラリのライブラリファイル格納ディレクトリの設定
HDF ライブラリのライブラリファイルが格納されているディレクトリを指定する。
- ④ ADIT のライブラリファイル格納ディレクトリの設定
make の結果作成される ADIT のライブラリファイルの格納ディレクトリを設定する。
括弧内は、デフォルトディレクトリで、インストーラが実行されたディレクトリ内の「lib」ディレクトリである。デフォルトディレクトリでよければエンターを入力し、変更する場合、ここで、ディレクトリを入力する。

インストーラ `install` の実行で①～④の設定が終了すると、自動的に ADIT のコンパイルが行われ、④で設定したディレクトリに ADIT のライブラリがインストールされます。ADIT のインストールが正常に終了しているかは、以下の操作で確認できます。

```
% ls -l <④で設定したディレクトリ>
```

「libADIT.a」が作成できていれば、インストール完了です。

2.3 ADITの環境設定

本ライブラリでは、時刻情報に関して閏秒 (TAI-UTC) 補正 (4.3.1 に示します SCAN_TIME に対して時刻補正を行います。)を行うため、閏秒情報ファイルを使用しています。本ライブラリの使用の際には、環境変数“LEAP_DATA”に閏秒情報ファイル名を絶対パスで指定する必要があります。以下に設定例を示します。

シェルに `csch`、`tcsh` をご利用の場合は以下のように環境変数を設定します。例えば `csch` 利用者は `.cschrc` に以下のように環境設定します。

```
setenv LEAP_DATA <①で設定したディレクトリ/etc/tai-utc.dat>
```

シェルに `bsh`、`ksh`、`bash` をご利用の場合は以下のように環境変数を設定します。例えば `bash` 利用者は `.bashrc` に以下のように環境設定します。

```
export LEAP_DATA=<①で設定したディレクトリ/etc/tai-utc.dat>
```

なお、閏秒情報ファイルは通常、数年に一度更新されますので、ブラウザから以下に示す URL を指定して、本ファイルを取得してください。

```
ftp://maia.usno.navy.mil/ser7/tai-utc.dat
```

内容が更新されている場合は<①で設定したディレクトリ/etc/tai-utc.dat>と入れ替えてご利用ください。

3 ADITを利用したプログラミング

3.1 プログラムの流れ

ADIT を使って、AMSR/AMSR-E データを C や Fortran のプログラムで読み込むためには、下記の要領でプログラムを作成します。（但し、89GHz 低周波相当データ算出関数を除く）

① ヘッダファイルの記述

プログラムで利用する ADIT 用のヘッダ情報ファイル(include ファイル)を指定します。AMSR データの L 1 B プロダクト、L2、L3 プロダクトの入出力構造体定義や、パラメータなどが記述されています。

② 入出力構造体の宣言

①でのヘッダファイル (include ファイル) に記述されている入出力構造体を「任意の名前に宣言します。以降、プログラムの中では、宣言された名前で参照されます。

③ HDF ファイルのオープン

読み込む HDF ファイルをオープンします。

④ メタデータの読み込み

HDF ファイルのうち、読み込むメタデータの各要素を変数に読み込みます。

⑤ スキャン毎又は複数スキャンのデータの読み込み

AMSR/AMSR-E の L 1 B 及び L 2、L 3 プロダクトは、1 シーン分の観測データなどが SD データや V データで格納されています。これらの 1 シーン分の SD データや V データを読み込みます。

この時、プロダクトに格納されているデータに変換係数が設定されているものがありますが、ADIT は変換係数を使用して、変換処理を行った結果をユーザが読み込むように処理しています。

⑥ HDF ファイルのクローズ

HDF ファイルをクローズし、データの操作を終了します。

89GHz 低周波相当データ算出関数は、スキャン毎ではなく 1 シーン分の観測データを 1 度で作成するため、プログラムの流れが異なります。

① **HDF ファイルのオープン**

読み込む HDF ファイルをオープンします。

② **89GHz 低周波相当データの算出**

読み込む HDF ファイル、パラメータファイルを入力し、89GHz 低周波相当データを算出します。

③ **HDF ファイルのクローズ**

HDF ファイルをクローズし、データの操作を終了します。

3.2 Cプログラミング

ここでは、簡単なCプログラミングについて説明します。

3.2.1 プログラムの記述

① ヘッダファイルの記述

#include <AMSR.h>	ADIT ライブラリのヘッダファイル
-------------------	--------------------

② 入出力構造体の宣言

AMSRL1B_SWATH *swath1b;	左側:ヘッダファイル AMSR.h で定義されている名称 右側:ユーザが定義する任意の名称
-------------------------	--

③ HDF ファイルのオープン

file_id=openV(HDF ファイル名);	HDF ファイルの V データのオープンを行うための記述。 file_id は、V データのアクセスに必要な file_id。 1 番目の引数は、HDF のファイル名
sd_id=openSD(HDF ファイル名);	HDF ファイルの SD データのオープンを行うための記述。 sd_id は、SD データのアクセスに必要な sd_id 1 番目の引数は、HDF のファイル名

④ メタデータの読み込み

status = getATTRIBUTE_NAME_AMSR(sd_id,"LocalGranuleID",granuleID);	HDF ファイルのメタデータを「メタデータ名」"LocalGranuleID"で読み込むための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な sd_id。 2 番目の引数は、HDF ファイルのメタデータ名である LocalGranuleID。 3 番目の引数は、ユーザが定義の変数名。
--	---

⑤ スキャン毎又は複数スキャンのデータの読み込み

スキャン毎に読み込む場合

for(i=0;i<scanno;i++){ status = getAMSRL1B_SWATH(sd_id,file_id,swath1b,i); }	スキャン毎のデータ構造体への入力を行うための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な sd_id。 2 番目の引数は、③でオープンした V データのアクセスに必要な file_id。 3 番目の引数は、ユーザ定義の入出力構造体宣言名。 4 番目の引数は、読み込むスキャン番号 (0 開始)
--	---

複数スキャンを一度に読み込む場合

<code>status = getAMSRL1B_SWATH_line(sd_id,file_id,swath1b,start_scan,end_scan);</code>	
	<p>スキャン毎のデータ構造体への入力を行うための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な <code>sd_id</code>。 2 番目の引数は、③でオープンした V データのアクセスに必要な <code>file_id</code>。 3 番目の引数は、ユーザ定義の入出力構造体宣言名。 4 番目の引数は、読み込むスキャンの開始番号 (0 開始) 5 番目の引数は、読み込むスキャンの終了番号 (0 開始)</p>

⑥ HDF ファイルのクローズ

<code>status = closeV(file_id);</code>	<p>HDF ファイルの V データのクローズを行うための記述。 1 番目の引数は、③でオープンした V データのアクセスに必要な <code>file_id</code>。</p>
<code>status = closeSD(sd_id);</code>	<p>HDF ファイルの SD データのオープンを行うための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な <code>sd_id</code>。</p>

3.2.2 コンパイル

ここでは、代表的な UNIX 計算機である SUN、SGI や LINUX における C プログラムのコンパイルの例を示します。

① SUN (Solaris 8)

コンパイルオプションとして、“`-DSUN -Xc -lnsl -lm`” を付けます。

<pre>cc -DSUN -Xc -lnsl -o sample1 sample1.c ¥ -I/home/amsr/work/ADITv3.00/include -I/home/amsr/work/HDF4.2r5/include¥ -L/home/amsr/work/ADITv3.00/lib -L/home/amsr/work/HDF4.2r5/lib¥ -lADIT -lmfhdf -ldf [-ljpeg] -lz -lm</pre>

② SGI (IRIX6.5)

コンパイルオプションとして、“`-DSGI -xansi -O -s -lm`” を付けます。

<pre>cc -DSGI -xansi -O -s -o sample1 sample1.c ¥ -I/home/amsr/work/ADITv3.00/include -I/home/amsr/work/HDF4.2r5/include¥ -L/home/amsr/work/ADITv3.00/lib -L/home/amsr/work/HDF4.2r5/lib¥ -lADIT -lmfhdf -ldf [-ljpeg] -lz -lm</pre>
--

③ LINUX (2.2.13-33)

コンパイルオプションとして“`-DLINUX -ansi -lm`” を付けます。

<pre>gcc -DLINUX -ansi -o sample1 sample1.c ¥ -I/home/amsr/work/ADITv3.00/include -I/home/amsr/work/HDF4.2r5/include¥ -L/home/amsr/work/ADITv3.00/lib -L/home/amsr/work/HDF4.2r5/lib¥ -lADIT -lmfhdf -ldf [-ljpeg] -lz -lm</pre>
--

3.2.3 C サンプルプログラム

ここでは、sample1.c について説明します。

```
#include <stdio.h>
#include <AMSR.h>

int main(int argc,char *argv[])
{
    int32 i,scanno;
    int32 file_id,sd_id;

    char scannoC[10],granuleID[40];
    char beginD[20],beginT[20];
    char endD[20],endT[20];

    AMSRL1B_SWATH *swath1b;
    SCAN_TIME *scantime;
    SUN_EARTH *sunearth;
    STATUS_L1B *status1b;
    CAL *cal;
    NAVI *navi;

    /* Argument check */
    if(argc!=2) {
        printf("USAGE : sample1 <AMSR/L1B filename>¥n");
        exit(1);
    }

    /* V&SD HDF open */
    if( (file_id=openV(argv[1]))==FAIL ) {
        fprintf(stderr,"Vdata open error(%s)¥n",argv[1]);
        exit(1);
    }

    if( (sd_id=openSD(argv[1])) ==FAIL ) {
        fprintf(stderr,"SDdata open error(%s)¥n",argv[1]);
    }
}
```

ADIT を使用するために必要なヘッダファイルの取り込み

ADIT で定義されている構造体の変数宣言

プログラムの引数を確認する記述

HDF の SD データ、V データをオープンするための記述


```

        exit(1);
    }
}

/* coremeta read by name call */
    if( (getATTRIBUTE_NAME_AMSR(sd_id,"LocalGranuleID",granuleID))==FAIL)
exit(1);
    printf("GRANULE ID(call by NAME) : %s¥n",granuleID);

/* coremeta read by attr_index call */
    if( (getATTRIBUTE_AMSR(sd_id,3,granuleID))==FAIL) exit(1);
    printf("GRANULE ID(call by INDEX) : %s¥n",granuleID);

    if( (getATTRIBUTE_AMSR(sd_id,28,scannoC))==FAIL) exit(1);
    scanno=atoi(scannoC);
    printf("SCANNO : %d¥n",scanno);

    if (getATTRIBUTE_AMSR(sd_id,7,beginT)==FAIL) exit(1);
    if (getATTRIBUTE_AMSR(sd_id,8,beginD)==FAIL) exit(1);
    if (getATTRIBUTE_AMSR(sd_id,9,endT) ==FAIL) exit(1);
    if (getATTRIBUTE_AMSR(sd_id,10,endD) ==FAIL) exit(1);
    printf("OBS. TIME : %s %s - %s %s¥n",beginD,beginT,endD,endT);

/* memory allocation */
    swath1b=(AMSRL1B_SWATH *)calloc(1,sizeof(AMSRL1B_SWATH));
    scantime=(SCAN_TIME *)calloc(1,sizeof(SCAN_TIME));
    sunearth=(SUN_EARTH *)calloc(1,sizeof(SUN_EARTH));
    status1b=(STATUS_L1B *)calloc(1,sizeof(STATUS_L1B));
    cal=(CAL *)calloc(1,sizeof(CAL));
    navi=(NAVI *)calloc(1,sizeof(NAVI));

/* data read every scan */
    for(i=0;i<scanno;i++)
    {
        printf("SCAN NO. %04d/%d¥n",i+1,scanno);
        if (getAMSRL1B_SWATH(sd_id,file_id,swath1b,i)==FAIL) exit(1);
        if (getSCANTIME_AMSR1(file_id,scantime,i) ==FAIL) exit(1);
    }
}

```

"}
"LocalGranuleID"をメタデータ名から読み込むための記述

"LocalGranuleID"をメタデータ番号から読み込むための記述

```

    if (getSUN_EARTH(sd_id,sunearth,i)      ==FAIL) exit(1);
    if (getStatus_L1B(sd_id,status1b,i)    ==FAIL) exit(1);
    if (getCALIBRATION(sd_id,cal,i)        ==FAIL) exit(1);
    if (getNAVIGATION(sd_id,navi,i)        ==FAIL) exit(1);
    }

/* V&SD close */
closeV(file_id);
closeSD(sd_id);

free(swath1b);
free(scantime);
free(sunearth);
free(status1b);
free(cal);
free(navi);

return 0;
}

```

宣言した構造体に HDF のデータを各スキャン毎に読み込むための記述

HDF の SD データ、V データのクローズを行うための記述

*各構造体の各メンバへのアクセス方法については、5章 サンプルプログラム一覧を参照のこと。

3.3 Fortranプログラミング (SunOS版、SGI版)

ここでは、簡単な F77 プログラミングについて説明します。

3.3.1 プログラムの記述

① ヘッダファイルの記述

include 'AMSR_f.h'	ADIT ライブラリのヘッダファイル
--------------------	--------------------

② 入出力構造体の宣言

record /AMSR_L1B_SWATH/ swath1b	左側がヘッダファイル AMSR_f.h で定義されている名称。 右側がユーザが定義する任意の名称 *ここでの構造体とは、配列を 2 つ組み合わせた擬似的なものである。
---------------------------------	---

③ HDF ファイルのオープン

file_id=openV(HDF ファイル名)	HDF ファイルの V データのオープンを行うための記述。 file_id は、V データのアクセスに必要な file_id。 1 番目の引数は、HDF のファイル名
sd_id=openSD(HDF ファイル名)	HDF ファイルの SD データのオープンを行うための記述。 sd_id は、SD データのアクセスに必要な sd_id 1 番目の引数は、HDF のファイル名

④ メタデータの読み込み

status=getATTRIBUTE_NAME_AMSR(sd_id,'LocalGranuleID',granuleID)	HDF ファイルのメタデータを「メタデータ名」'LocalGranuleID'で読み込むための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な sd_id。 2 番目の引数は、HDF ファイルのメタデータ名である LocalGranuleID。 3 番目の引数は、ユーザが定義の変数名。
---	---

⑤ スキャン毎又は複数スキャンのデータの読み込み

スキャン毎に読み込む場合

<pre>do 10 i=0,scanno-1,1 status=getAMSRL1B_SWATH(sd_id,file_id,swath1b,i) 10 continue</pre>	<p>スキャン毎のデータ構造体への入力を行うための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な sd_id。 2 番目の引数は、③でオープンした V データのアクセスに必要な file_id。 3 番目の引数は、ユーザ定義の入出力構造体宣言名。 4 番目の引数は、読み込むスキャン番号 (0 開始)</p>
--	--

複数スキャンを一度に読み込む場合

<pre>status = getAMSRL1B_SWATH_line(sd_id,file_id,swath1b,start_scan,end_scan);</pre>	<p>スキャン毎のデータ構造体への入力を行うための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な sd_id。 2 番目の引数は、③でオープンした V データのアクセスに必要な file_id。 3 番目の引数は、ユーザ定義の入出力構造体宣言名。 4 番目の引数は、読み込むスキャンの開始番号 (0 開始) 5 番目の引数は、読み込むスキャンの終了番号 (0 開始)</p>
---	--

⑥ HDF ファイルのクローズ

<pre>status = closeV(file_id)</pre>	<p>HDF ファイルの V データのクローズを行うための記述。 1 番目の引数は、③でオープンした V データのアクセスに必要な file_id。</p>
<pre>status = closeSD(sd_id)</pre>	<p>HDF ファイルの SD データのクローズを行うための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な sd_id。</p>

3.3.2 コンパイル

ここでは、代表的な UNIX 計算機である SUN、SGI における Fortran プログラムのコンパイルの例を示します。なお、FORTRAN については LINUX に対応していません。

① SUN (Solaris 8)

コンパイルオプションとして、“-DSUN -O -lnsl -lm” を付けます。

```
f77 -DSUN -O -lnsl -o sample1f sample1f.f ¥  
-I/home/amsr/work/ADITv3.00/include -I/home/amsr/work/HDF4.2r5/include¥  
-L/home/amsr/work/ADITv3.00/lib -L/home/amsr/work/HDF4.2r5/lib¥  
-lADIT -lmfhdf -ldf -lz [-ljpeg] -lm
```

② SGI (IRIX6.5)

コンパイルオプションとして、“-DSGI -O -s -lm” を付けます。

```
f77 -DSGI -O -s -o sample1f sample1f.f ¥  
-I/home/amsr/work/ADITv3.00/include -I/home/amsr/work/HDF4.2r5/include¥  
-L/home/amsr/work/ADITv3.00/lib -L/home/amsr/work/HDF4.2r5/lib¥  
-lADIT -lmfhdf -ldf -lz [-ljpeg] -lm
```

3.3.3 Fortran サンプルプログラム

ここでは、sample1f.f について説明します。

```
program main
```

```
include 'AMSR_f.h'
```

```
character*30 fname
```

```
data fname/'A2AMS01092011MD_P01B0000000.00/'
```

```
integer status
```

```
integer i,scanno
```

```
integer file_id,sd_id
```

```
character*10 scannoC
```

```
character*40 granuleID
```

```
character*20 beginD,beginT
```

```
character*20 endD,endT
```

ADIT を使用するために必要なヘッダファイルの取り込み

読み込みたい HDF ファイル名の記述

```

record /AMSRL1B_SWATH/ swath1b
record /SCAN_TIME/      scantime
record /SUN_EARTH/     sunearth
record /STATUS_L1B/    status1b
record /CAL/           cal
record /NAVI/          navi

```

ADIT で定義されている構造体
の変数宣言
*ここでいう構造体とは、配列
を 2 つ組み合わせた擬似的なも
のである。

```

C*   V&SD HDF open
file_id=openV(fname)
if(file_id .eq. FAIL) then
    write(6,'(a,a,a)' 'Vdata open error('fname,')'
    stop
end if
sd_id=openSD(fname)
if(sd_id .eq. FAIL) then
    write(6,'(a,a,a)' 'SDdata open error('fname,')'
    stop
end if

```

HDF の SD データ、V データをオー
プンするための記述

```

C*   coremeta read by name  call
status=getATTRIBUTE_NAME_AMSR(sd_id,'LocalGranuleID',granuleID)
if(status .eq. FAIL) stop
write(6,'(a,a27)' 'GRANULE ID(call by NAME) : ',granuleID

```

"LocalGranuleID"をメタデータ名から
読み込むための記述

```

C*   coremeta read by attr_index call
status=getATTRIBUTE_AMSR(sd_id,3,granuleID)
if(status .eq. FAIL) stop
write(6,'(a,a27)' 'GRANULE ID(call by INDEX) : ',granuleID

status=getATTRIBUTE_AMSR(sd_id,28,scannoC)
if(status .eq. FAIL) stop
    write(6,*) ichar(scannoC(1:1))
scanno=(ichar(scannoC(1:1))-48)*1000
+      +(ichar(scannoC(2:2))-48)*100
+      +(ichar(scannoC(3:3))-48)*10

```

"LocalGranuleID"をメタデータ番号から読み
込むための記述

```

+      +(ichar(scannoC(4:4))-48)
write(6,'(a,i4)' 'SCANNO : ',scanno

status=getATTRIBUTE_AMSR(sd_id,7,beginT)
if(status .eq. FAIL) stop
status=getATTRIBUTE_AMSR(sd_id,8,beginD)
if(status .eq. FAIL) stop
status=getATTRIBUTE_AMSR(sd_id,9,endT)
if(status .eq. FAIL) stop
status=getATTRIBUTE_AMSR(sd_id,10,endD)
if(status .eq. FAIL) stop
write(6,'(a,a12,a,a10,a,a12,a,a10)'
*      'OBS. TIME : ',beginD,' ',beginT,
*      ' - ',endD,' ',endT

```

```

C*      data read every scan
do 10 i=0,scanno-1,1
  write(6,'(a,i4.4,a,i4.4)' 'SCAN NO. ',i+1,',' ,scanno
  status=getAMSRL1B_SWATH(sd_id,file_id,swath1b,i)
  if(status .eq. FAIL) stop
  status=getSCANTIME_AMSR1(file_id,scantime,i)
  if(status .eq. FAIL) stop
  status=getSUN_EARTH(sd_id,sunearth,i)
  if(status .eq. FAIL) stop
  status=getSTATUS_L1B(sd_id,status1b,i)
  if(status .eq. FAIL) stop
  status=getCALIBRATION(sd_id,cal,i)
  if(status .eq. FAIL) stop
  status=getNAVIGATION(sd_id,navi,i)
  if(status .eq. FAIL) stop
10 continue

```

宣言した構造体に HDF のデータを各スキャン毎に読み込むための記述

```

C*      V&SD close
status=closeV(file_id)
status=closeSD(sd_id)

stop

```

HDF の SD データ、V データのクローズを行うための記述

end

*各構造体の各データへのアクセス方法については、5章 サンプルプログラム一覧を参照のこと。

3.4 Fortranプログラミング(Linux版)

ここでは、簡単な F77 プログラミング(Linux 版)について説明します。

3.4.1 プログラムの記述

① ヘッダファイルの記述

include 'AMSR_Linux_f.h'	ADIT ライブラリのヘッダファイル
--------------------------	--------------------

② 入出力構造体の宣言

TYPE (AMSR_L1B_SWATH) swath1b	左側がヘッダファイル AMSR_Linux_f.h で定義されている名称。 右側がユーザが定義する任意の名称 *ここでの構造体とは、配列を 2 つ組み合わせた擬似的なものである。
-------------------------------	---

③ HDF ファイルのオープン

file_id=openV(HDF ファイル名)	HDF ファイルの V データのオープンを行うための記述。 file_id は、V データのアクセスに必要な file_id。 1 番目の引数は、HDF のファイル名
sd_id=openSD(HDF ファイル名)	HDF ファイルの SD データのオープンを行うための記述。 sd_id は、SD データのアクセスに必要な sd_id 1 番目の引数は、HDF のファイル名

④ メタデータの読み込み

status=getATTRIBUTE_NAME_AMSR(sd_id,'LocalGranuleID',granuleID)	HDF ファイルのメタデータを「メタデータ名」'LocalGranuleID'で読み込むための記述。 1 番目の引数は、③でオープンした SD データのアクセスに必要な sd_id。 2 番目の引数は、HDF ファイルのメタデータ名である LocalGranuleID。 3 番目の引数は、ユーザが定義の変数名。
---	---

⑤ スキャン毎のデータの読み込み

スキャン毎に読み込む場合

do 10 i=0,scanno-1,1 status=getAMSR_L1B_SWATH(sd_id,file_id,swath1b,i) 10 continue
--

	<p>スキャン毎のデータ構造体への入力を行うための記述。</p> <p>1 番目の引数は、③でオープンした SD データのアクセスに必要な <code>sd_id</code>。</p> <p>2 番目の引数は、③でオープンした V データのアクセスに必要な <code>file_id</code>。</p> <p>3 番目の引数は、ユーザ定義の入出力構造体宣言名。</p> <p>4 番目の引数は、読み込むスキャン番号 (0 開始)</p>
--	--

複数スキャンを一度に読み込む場合

<code>status = getAMSRL1B_SWATH_line(sd_id,file_id,swath1b,start_scan,end_scan);</code>	
	<p>スキャン毎のデータ構造体への入力を行うための記述。</p> <p>1 番目の引数は、③でオープンした SD データのアクセスに必要な <code>sd_id</code>。</p> <p>2 番目の引数は、③でオープンした V データのアクセスに必要な <code>file_id</code>。</p> <p>3 番目の引数は、ユーザ定義の入出力構造体宣言名。</p> <p>4 番目の引数は、読み込むスキャンの開始番号 (0 開始)</p> <p>5 番目の引数は、読み込むスキャンの終了番号 (0 開始)</p>

⑥ HDF ファイルのクローズ

<code>status = closeV(file_id)</code>	<p>HDF ファイルの V データのクローズを行うための記述。</p> <p>1 番目の引数は、③でオープンした V データのアクセスに必要な <code>file_id</code>。</p>
<code>status = closeSD(sd_id)</code>	<p>HDF ファイルの SD データのクローズを行うための記述。</p> <p>1 番目の引数は、③でオープンした SD データのアクセスに必要な <code>sd_id</code>。</p>

3.4.2 コンパイル

ここでは、代表的な UNIX 計算機である SUN、SGI における Fortran プログラムのコンパイルの例を示します。

① pgi

コンパイルオプションとして、“-DLINUX -O -lm” を付けます。

<pre>pgf90 -DLINUX -O -o sample1f sample1f.f ¥ -I/home/amsr/work/ADITv3.00/include -I/home/amsr/work/HDF4.2r5/include¥ -L/home/amsr/work/ADITv3.00/lib -L/home/amsr/work/HDF4.2r5/lib¥ -lADIT -lmfhdf -ldf -lz [-ljpeg] -lm</pre>

② gun

コンパイルオプションとして、“-DLINUX -O -lm” を付けます。

<pre>gfortran -DLINUX -O -o sample1f sample1f.f ¥ -I/home/amsr/work/ADITv3.00/include -I/home/amsr/work/HDF4.2r5/include¥ -L/home/amsr/work/ADITv3.00/lib -L/home/amsr/work/HDF4.2r5/lib¥ -lADIT -lmfhdf -ldf -lz [-ljpeg] -lm</pre>
--

③ Intel

コンパイルオプションとして、“-DLINUX -O -lm”を付けます。

```
ifort -DLINUX -O -o sample1f sample1.f ¥  
-I/home/amsr/work/ADITv3.00/include -I/home/amsr/work/HDF4.2r5/include¥  
-L/home/amsr/work/ADITv3.00/lib -L/home/amsr/work/HDF4.2r5/lib¥  
-lADIT-lmfhdf -ldf -lz [-ljpeg] -lm
```

3.4.3 Fortran サンプルプログラム

ここでは、sample1f.f について説明します。

```
program main

include 'AMSR_Linux_f.h'

character*100 fname
character*100 LOCALGRANULEID
integer status
integer i, scanno
integer file_id, sd_id

character*10 scannoC
character*40 granuleID
character*20 beginD, beginT
character*20 endD, endT
character buff*100

data fname /'../data/P1AME030609207MA_P01B0000000.00.sample'/
data LOCALGRANULEID /'LocalGranuleID'/

TYPE (AMSR_L1B_SWATH) swath1b
TYPE (SCAN_TIME)      scantime
TYPE (SUN_EARTH)      sunearth
TYPE (STATUS_L1B)     status1b
TYPE (CAL)             cal_data
TYPE (NAVI)           navi_data

do 100 i=1, 100, 1
  if( fname(i:i).eq.' ') go to 101
100 continue
101 fname(i:i)=char(0)

do 110 i=1, 100, 1
  if( LOCALGRANULEID(i:i).eq.' ') go to 111
```

ADIT を使用するために必要な
ヘッダファイルの取り込み

読み込みたい HDF ファイル名の記述

ADIT で定義されている構造体
の変数宣言
*ここでいう構造体とは、配列
を 2 つ組み合わせた擬似的なも
のである。

```

110 continue
111 LOCALGRANULEID(i:i)=char(0)

```

```

C* V&SD HDF open
file_id=openV(fname)
if(file_id .eq. FAIL) then
  write(6,'(a,a,a)' 'Vdata open error(',fname,')'
  stop
end if
sd_id=openSD(fname)
if(sd_id .eq. FAIL) then
  write(6,'(a,a,a)' 'SDdata open error(',fname,')'
  stop
end if

```

HDF の SD データ、V データをオープンするための記述

```

C* coremeta read by name call
status=getATTRIBUTE_NAME_AMSR(sd_id,'LocalGranuleID',granuleID)
if(status .eq. FAIL) stop
write(6,'(a,a27)' 'GRANULE ID(call by NAME) : ',granuleID

```

"LocalGranuleID"をメタデータ名から読み込むための記述

```

C* coremeta read by attr_index call
status=getATTRIBUTE_AMSR(sd_id,3,granuleID)
if(status .eq. FAIL) stop
write(6,'(a,a27)' 'GRANULE ID(call by INDEX) : ',granuleID

```

"LocalGranuleID"をメタデータ番号から読み込むための記述

```

status=getATTRIBUTE_AMSR(sd_id,28,scannoC)
if(status .eq. FAIL) stop
  write(6,*) ichar(scannoC(1:1))
scanno=(ichar(scannoC(1:1))-48)*1000
+   +(ichar(scannoC(2:2))-48)*100
+   +(ichar(scannoC(3:3))-48)*10
+   +(ichar(scannoC(4:4))-48)
write(6,'(a,i4)' 'SCANNO : ',scanno

status=getATTRIBUTE_AMSR(sd_id,7,beginT)
if(status .eq. FAIL) stop
status=getATTRIBUTE_AMSR(sd_id,8,beginD)

```

```

if(status .eq. FAIL) stop
status=getATTRIBUTE_AMSR(sd_id,9,endT)
if(status .eq. FAIL) stop
status=getATTRIBUTE_AMSR(sd_id,10,endD)
if(status .eq. FAIL) stop
write(6,'(a,a12,a,a10,a,a12,a,a10)')
*      'OBS. TIME : ',beginD,' ',beginT,
*      '- ',endD,' ',endT

```

C* data read every scan

```

do 10 i=0,scanno-1,1
  write(6,'(a,i4.4,a,i4.4)') 'SCAN NO. ',i+1,'/',scanno
  status=getAMSRL1B_SWATH(sd_id,file_id,swath1b,i)
  if(status .eq. FAIL) stop
  status=getSCANTIME_AMSR1(file_id,scantime,i)
  if(status .eq. FAIL) stop
  status=getSUN_EARTH(sd_id,sunearth,i)
  if(status .eq. FAIL) stop
  status=getSTATUS_L1B(sd_id,status1b,i)
  if(status .eq. FAIL) stop
  status=getCALIBRATION(sd_id,cal,i)
  if(status .eq. FAIL) stop
  status=getNAVIGATION(sd_id,navi,i)
  if(status .eq. FAIL) stop
10 continue

```

C* V&SD close

```

status=closeV(file_id)
status=closeSD(sd_id)

stop
end

```

宣言した構造体に HDF のデータを各スキャン毎に読み込むための記述

HDF の SD データ、V データのクローズを行うための記述

*各構造体の各データへのアクセス方法については、5章サンプルプログラム一覧を参照のこと。

4 APPENDIX

4.1 Routines defined in ADIT

Specific routines handle the specific level of AMSR/AMSR-E products. The detailed descriptions of these routines are introduced in a later section.

Table 4.1-1 Routine table in ADIT

Product level	Routine name	Description
L1B	openV()	File open and initialize for HDF/Vdata
L2	closeV()	File close for HDF/Vdata
L3	openSD()	File open and initialize for HDF/SDdata
	closeSD()	File close for HDF/SD data
	getATTRIBUTE_NAME_AMSR()	Read metadata (by "metadata name") (See Section 4.4.)
	getATTRIBUTE_AMSR()	Read metadata (by "attr index") (See Section 4.4.)
L1B	getAMSRL1B_SWATH()	Read HDF and input data to the structure "AMSRL1B_SWATH" (See Section 4.3.2.)
	getSCANTIME_AMSR1()	Read HDF and input data to the structure "SCAN_TIME" (See Section 4.3.1.)
	getSUN_EARTH()	Read HDF and input data to the structure "SUN_EARTH" (See Section 4.3.3.)
	getSTATUS_L1B()	Read HDF and input data to the structure "STATUS_L1B" (See Section 4.3.4.)
	getCALIBRATION()	Read HDF and input data to the structure "CAL" (See Section 4.3.5.)
	getNAVIGATION()	Read HDF and input data to the structure "NAVI" (See Section 4.3.6.)
	getAMSR_89LOW()	The data of 89GHz low frequency corresponding is calculated from 89GHz A horn and B horn.
	getAMSRL1B_SWATH_line()	Read HDF and input data to the structure "AMSRL1B_SWATH" (See Section 4.3.2.)
	getSCANTIME_AMSR1_line()	Read HDF and input data to the structure "SCAN_TIME" (See Section 4.3.1.)
	getSUN_EARTH_line()	Read HDF and input data to the structure "SUN_EARTH" (See Section 4.3.3.)
	getSTATUS_L1B_line()	Read HDF and input data to the structure "STATUS_L1B" (See Section 4.3.4.)
	getCALIBRATION_line()	Read HDF and input data to the structure "CAL" (See Section 4.3.5.)
	getNAVIGATION_line()	Read HDF and input data to the structure "NAVI" (See Section 4.3.6.)
L2	getAMSRL2_SWATH()	Read HDF and input data to the structure "AMSRL2_SWATH" (See Section 4.3.7.)
	getSCANTIME_AMSR2()	Read HDF and input data to the structure "SCAN_TIME" (See Section 4.3.1.)
	getSTATUS_L2()	Read HDF and input data to the structure "STATUS_L2" (See Section 4.3.8.)
	getAMSRL2_SWATH_line()	Read HDF and input data to the structure "AMSRL2_SWATH" (See Section 4.3.7.)
	getSCANTIME_AMSR2_line()	Read HDF and input data to the structure "SCAN_TIME" (See Section 4.3.1.)
	getSTATUS_L2_line()	Read HDF and input data to the structure "STATUS_L2" (See Section 4.3.8.)

Product level	Routine name	Description
L3	getAMSRL3_MAP()	Read L3 science data and input it to data, which you prepared (See Section 4.3.9.)
	getDIMSIZE()	Access L3 data and get information of L3 science data sizes (See Section 4.3.9.)

4.2 User routine interface in ADIT

User routine interface in ADIT is shown in Table 4.2-1 for C program, and Table 4.2-2 for f77 program.

Table 4.2-1 Routine interface for C

Note: int32 means 4byte int.

Routine name	Parameter	Parameter Type	Input/Output	Note
file_id = openV (Filename)				
	file_id	int32	Output	If failed, return value is FAIL (or -1)
	File_name	char *	Input	AMSR/AMSR-E HDF Filename
status = closeV (file_id)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	file_id	int32	Input	HDF/Vdata access file id
sd_id = openSD (Filename)				
	sd_id	int32	Output	If failed, return value is FAIL (or -1)
	File_name	char *	Input	AMSR/AMSR-E Filename
status = closeSD (sd_id)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
status = getATTRIBUTE_NAME_AMSR (sd_id,name,value)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	If failed, return value is FAIL (or -1)
	name	char *	Input	Metadata name (See Section 4.4.)
	value	char *	Output	Metadata values (See Section 4.4.)
status = getATTRIBUTE_AMSR (sd_id,index,value)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	If failed, return value is FAIL (or -1)
	index	int32	Input	Metadata index (See Section 4.4.)
	value	char *	Output	Metadata values (See Section 4.4.)
status = getAMSRL1B_SWATH (sd_id,file_id,amsr1b_swath,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	file_id	int32	Input	HDF/Vdata access file id
	amsr1b_swath	AMSRL1B_SWATH *	Output	information structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status = getSCANTIME_AMSR1 (sd_id,scan_time,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	scan_time	SCAN_TIME *	Output	information structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status = getSUN_EARTH (sd_id,sun_earth,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	sun_earth	SUN_EARTH *	Output	information structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status = getSTATUS_L1B (sd_id,status_l1b,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id

Table 4.2-1 Routine interface for C

Note: int32 means 4byte int.

Routine name	Parameter	Parameter Type	Input/Output	Note
	status_l1b	STATUS_L1B *	Output	information structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status=getCALIBRATION (sd_id,cal,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	cal	CAL *	Output	information structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status = getNAVIGATION (sd_id,navi,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	navi	NAVI *	Output	information structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status = getAMSR_89LOW (sd_id,pol_id,para_name_A,para_name_B,bt_89low)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	pol_id	int32	Input	Polarization id (V-pol : 0, H-pol:1)
	para_name_A	char *	Input	Parameter file for A horn □1
	para_name_B	char *	Input	Parameter file for B horn □1
	bt_89low	float *	Output	equivalent of 89GHz low frequency data
status = getAMSR1B_SWATH_line (sd_id,file_id,amsr1b_swath,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	file_id	int32	Input	HDF/Vdata access file id
	amsr1b_swath	AMSR1B_SWATH *	Output	information structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan	int32	Input	End scan No. (Beginning0)
status = getSCANTIME_AMSR1_line (sd_id,scan_time,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	scan_time	SCAN_TIME *	Output	information structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan	int32	Input	End scan No. (Beginning0)
status = getSUN_EARTH_line (sd_id,sun_earth,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	sun_earth	SUN_EARTH *	Output	information structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan	int32	Input	End scan No. (Beginning0)
status = getSTATUS_L1B_line (sd_id,status_l1b,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	status_l1b	STATUS_L1B *	Output	information structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan	int32	Input	End scan No. (Beginning0)

Table 4.2-1 Routine interface for C

Note: int32 means 4byte int.

Routine name	Parameter	Parameter Type	Input/Output	Note
status=getCALIBRATION_line (sd_id,cal,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	cal	CAL *	Output	information structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan	int32	Input	End scan No. (Beginning0)
status = getNAVIGATION_line (sd_id,navi,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	navi	NAVI *	Output	information structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan	int32	Input	End scan No. (Beginning0)
status = getAMSRL2_SWATH (sd_id,file_id,amsrl2_swath,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	file_id	int32	Input	HDF/Vdata access file id
	amsrl2_swath	AMSRL2_SWATH *	Output	structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status = getSCANTIME_AMSR2 (sd_id,scan_time,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	scan_time	SCAN_TIME *	Output	structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status = getSTATUS_L2 (sd_id,status_l2,scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	status_l2	STATUS_L2 *	Output	structure defined in ADIT
	scan	int32	Input	Scan No.(Beginning 0)
status = getAMSRL2_SWATH_line (sd_id,file_id,amsrl2_swath,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	file_id	int32	Input	HDF/Vdata access file id
	amsrl2_swath	AMSRL2_SWATH *	Output	structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan		Input	End scan No. (Beginning0)
status = getSCANTIME_AMSR2_line (sd_id,scan_time,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	scan_time	SCAN_TIME *	Output	structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan		Input	End scan No. (Beginning0)
status = getSTATUS_L2_line (sd_id,status_l2,start_scan,end_scan)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	status_l2	STATUS_L2 *	Output	structure defined in ADIT
	start_scan	int32	Input	Start scan No.(Beginning0)
	end_scan		Input	End scan No. (Beginning0)
status = getAMSRL3_MAP (sd_id, file_id, map_2int,map_float,size)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id

Table 4.2-1 Routine interface for C

Note: int32 means 4byte int.

Routine name	Parameter	Parameter Type	Input/Output	Note
	file_id	int32	Input	HDF/Vdata access file id
	map_2int	short *	Output	L3 science data buffer, which has type of short
	map_float	float *	Output	L3 science data, which has type of float
	size	int	Input	L3 science data size, which has value of n line x n pixel
status = getDIMSIZE (sd_id,ref_no,SIZE)				
	sd_id	int32	Input	HDF/SD access SD id
	ref_no	int32	Input	HDF/SD access SD reference number
	SIZE	int32 *	Output	L3 science data size, which has value of n line x n pixel

※1 The strage directory of parameter files

AMSR

Parameter file for A horn

(install directory)/MAKE_89_LOW_PAM/A2AMS/A289A.prm

Parameter file for B horn

(install directory)/MAKE_89_LOW_PAM/A2AMS/A289B.prm

AMSR-E

Parameter file for A horn

(install directory)/MAKE_89_LOW_PAM/P1AME/P189A.prm

Parameter file for B horn

(install directory)/MAKE_89_LOW_PAM/P1AME/P189B.prm

Table 4.2-2 Routine interface for f77

Note: integer*2 means 2byte int, and real*4 means 4byte real.

Routine name	Parameter	Parameter Type	Input/Output	Note
file_id = openV (Filename)				
	file_id	integer	Output	If failed, return value is FAIL (or -1)
	File_name	character	Input	AMSR/AMSR-E HDF Filename
status = closeV (file_id)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	file_id	integer	Input	HDF/Vdata access file id
sd_id = openSD (Filename)				
	sd_id	integer	Output	If failed, return value is FAIL (or -1)
	File_name	character	Input	AMSR/AMSR-E Filename
status = closeSD (sd_id)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
status = getATTRIBUTE_NAME_AMSR (sd_id,name,value)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	If failed, return value is FAIL (or -1)
	name	character	Input	Metadata name (See Section 4.4.)
	value	character	Output	Metadata values (See Section 4.4.)
status = getATTRIBUTE_AMSR (sd_id,index,value)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	If failed, return value is FAIL (or -1)
	index	integer	Input	Metadata index (See Section 4.4.)
	value	character	Output	Metadata values (See Section 4.4.)
status = getAMSRL1B_SWATH (sd_id,file_id,amsrl1b_swath,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	file_id	integer	Input	HDF/Vdata access file id
	amsrl1b_swath	AMSRL1B_SWATH	Output	information structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status = getSCANTIME_AMSR1 (sd_id,scan_time,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	scan_time	SCAN_TIME	Output	information structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status = getSUN_EARTH (sd_id,sun_earth,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	sun_earth	SUN_EARTH	Output	information structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status = getSTATUS_L1B (sd_id,status_l1b,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	status_l1b	STATUS_L1B	Output	information structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status=getCALIBRATION (sd_id,cal,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)

Table 4.2-2 Routine interface for f77

Note: integer*2 means 2byte int, and real*4 means 4byte real.

Routine name	Parameter	Parameter Type	Input/Output	Note
	sd_id	integer	Input	HDF/SD access SD id
	cal	CAL	Output	information structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status = getNAVIGATION (sd_id,navi,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	navi	NAVI	Output	information structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status = getAMSR_89LOW (sd_id,pol_id,para_name_A,para_name_B,bt_89low)				
	status	int32	Output	If failed, return value is FAIL (or -1)
	sd_id	int32	Input	HDF/SD access SD id
	pol_id	int32	Input	Polarization id (V-pol : 0, H-pol:1)
	para_name_A	char *	Input	Parameter file for A horn □2
	para_name_B	char *	Input	Parameter file for B horn □2
	bt_89low	float *	Output	equivalent of 89GHz low frequency data
status = getAMSR1B_SWATH_line (sd_id,file_id,amsr1b_swath,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	file_id	integer	Input	HDF/Vdata access file id
	amsr1b_swath	AMSR1B_SWATH	Output	information structure defined in ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status = getSCANTIME_AMSR1_line (sd_id,scan_time,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	scan_time	SCAN_TIME	Output	information structure defined in ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status = getSUN_EARTH_line (sd_id,sun_earth,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	sun_earth	SUN_EARTH	Output	information structure defined in ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status = getSTATUS_L1B_line (sd_id,status_l1b,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	status_l1b	STATUS_L1B	Output	information structure defined in ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status=getCALIBRATION_line (sd_id,cal,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	cal	CAL	Output	information structure defined in

Table 4.2-2 Routine interface for f77

Note: integer*2 means 2byte int, and real*4 means 4byte real.

Routine name	Parameter	Parameter Type	Input/Output	Note
				ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status = getNAVIGATION_line (sd_id,navi,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	navi	NAVI	Output	information structure defined in ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status = getAMSRL2_SWATH (sd_id,file_id,amsrl2_swath,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	file_id	integer	Input	HDF/Vdata access file id
	amsrl2_swath	AMSRL2_SWATH	Output	structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status = getSCANTIME_AMSR2 (sd_id,scan_time,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	scan_time	SCAN_TIME	Output	structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status = getSTATUS_L2 (sd_id,status_l2,scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	status_l2	STATUS_L2	Output	structure defined in ADIT
	scan	integer	Input	Scan No.(Beginning 0)
status = getAMSRL2_SWATH_line (sd_id,file_id,amsrl2_swath,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	file_id	integer	Input	HDF/Vdata access file id
	amsrl2_swath	AMSRL2_SWATH	Output	structure defined in ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status = getSCANTIME_AMSR2_line (sd_id,scan_time,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	scan_time	SCAN_TIME	Output	structure defined in ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status = getSTATUS_L2_line (sd_id,status_l2,start_scan,end_scan)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	status_l2	STATUS_L2	Output	structure defined in ADIT
	start_scan	integer	Input	Start scan No.(Beginning0)
	end_scan	integer	Input	End scan No. (Beginning0)
status = getAMSRL3_MAP (sd_id, file_id, map_2int,map_float,size)				
	status	integer	Output	If failed, return value is FAIL (or -1)
	sd_id	integer	Input	HDF/SD access SD id
	file_id	integer	Input	HDF/Vdata access file id
	map_2int	integer*2	Output	L3 science data buffer, which has type of 2byte int

Table 4.2-2 Routine interface for f77

Note: integer*2 means 2byte int, and real*4 means 4byte real.

Routine name	Parameter	Parameter Type	Input/Output	Note
	map_float	real*4	Output	L3 science data, which has type of 4byte real
	size	integer	Input	L3 science data size, which has value of n line x n pixel
status = getDIMSIZE (sd_id,ref_no,SIZE)				
	sd_id	integer	Input	HDF/SD access SD id
	ref_no	integer	Input	HDF/SD access SD reference number
	SIZE	integer	Output	L3 science data size, which has value of n line x n pixel

※2 The strage directory of parameter files

AMSR

Parameter file for A horn

(install directory)/MAKE_89_LOW_PAM/A2AMS/A289A.prm

Parameter file for B horn

(install directory)/MAKE_89_LOW_PAM/A2AMS/A289B.prm

AMSR-E

Parameter file for A horn

(install directory)/MAKE_89_LOW_PAM/P1AME/P189A.prm

Parameter file for B horn

(install directory)/MAKE_89_LOW_PAM/P1AME/P189B.prm

4.3 Structure definition in ADIT

You can read AMSR/AMSR-E data in HDF file using structures defined in ADIT. Using these structures, you can read specific data in the HDF file.

Table 4.3-1 Structure definitions

Product level	Structure name	Description
L1B,L2	SCAN_TIME	Information structure of the observational scanning time
L1B	AMSRL1B_SWATH	Information structure of swath data. The member of this structure is as follows. 1. structure "SCAN_TIME" 2. Brightness Temperature 3. Latitude and Longitude of the observation point
	SUN_EARTH	Information structure of angle data related to observation point, sun, and platform. The member of this structure is as follows. 1. Sun Azimuth 2. Sun Elevation 3. Earth Incidence 4. Earth Azimuth 5. Ocean/Lanf flag
	STATUS_L1B	Information structure related to status of the observation data. The member of this structure is as follows. 1. Orbit number 2. Observation Supplement 3. Data Quality
	CAL	Information structure of calibration data. The member of this structure is as follows. 1. Hot-load Count 2. Cold Sky Mirror Count 3. Antenna Temperature Coefficient 4. RX Offset/Gain Count 5. SPC Temperature Count 6. SPS Temperature Count 7. SPC Temperature 8. SPS Temperature
	NAVI	Information structure of navigation data. The member of this structure is as follows. 1. platform position(X,Y,Z) in inertial coordinate 2. platform velocity(Vx,Vy,Vz) in inertial coordinate 3. platform attitude(roll,pitch,yaw) in platform coordinate
L2	AMSRL2_SWATH	Information structure of swath data. The member of this structure is as follows. 1. structure "SCAN_TIME" 2. Geophysical data 3. Latitude and Longitude of the observation point
	STATUS_L2	Information structure of status of the observation data. The member of this structure is as follows. 1. Orbit number 2. Data Quality

4.3.1 L1B, L2 common structure

Table 4.3.1-1 L1B, L2 common structures

Name of structure	member	type	size	Description
SCAN_TIME	koyomi	8byte real	1	total second beginning 1970/1/1 0:0
	year	2byte int	1	year (UT)
	month	2byte int	1	month (UT)
	day	2byte int	1	day (UT)
	hour	2byte int	1	hour (UT)
	minute	2byte int	1	minute (UT)
	second	2byte int	1	second (UT)

(1) SCAN_TIME

“SCAN_TIME” is the structure of scanning start time of the observation. This scanning start time corresponds to the first point of observation in a scan. The member “koyomi” is the total seconds from 1970.01.01.00.00 (Unix system time). Though original scanning start time in L1B products is the total seconds from 1993.01.01.00 by UT (TAI time), ADIT converts TAI time into Unix system time for scanning start time.

4.3.2 AMSRL1B_SWATH (for L1B)

Table 4.3.2-1 AMSRL1B_SWATH

Name of structure	member	type	size	Description
AMSRL1B_SWATH	scan_time	SCAN_TIME	20	structure SCAN_TIME
	tb_low	4byte real	12 x 196	TB data for lower frequency channels Dimension: n channel x n pixel Variable numbers are defined as follows. AMSR-E does not have frequency 50GHz and 52GHz band, therefore these two band data are set to <i>zero</i> in every scan and pixel. 1: 6GHz vertical elements TB data [K] 2: 6GHz horizontal elements TB data [K] 3: 10GHz vertical elements TB data [K] 4: 10GHz horizontal elements TB data [K] 5: 18GHz vertical elements TB data [K] 6: 18GHz horizontal elements TB data [K] 7: 23GHz vertical elements TB data [K] 8: 23GHz horizontal elements TB data [K] 9: 36GHz vertical elements TB data [K] 10: 36GHz horizontal elements TB data [K] 11: 50GHz vertical elements TB data [K] 12: 52GHz vertical elements TB data [K]
	tb_high_A	4byte real	2 x 392	TB data for 89GHz channels (A-scan) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz A-horn vertical elements TB data [K] 2: 89GHz A-horn horizontal elements TB data

Table 4.3.2-1 AMSRL1B_SWATH

Name of structure	member	type	size	Description
				[K]
	tb_high_B	4byte real	2 x 392	TB data for 89GHz channels (B-scan) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz B-horn vertical elements TB data [K] 2: 89GHz B-horn horizontal elements TB data [K]
	latlon_low	4byte real	6 x 2 x 196	Geolocation of the observation point for each lower channels Dimension: n channel x n geolocation variable x n pixel. Lower channel variable numbers are defined as follows. AMSR-E does not have frequency 50GHz and 52GHz band, therefore the 6th data are set to <i>-9999.0</i> in every scan and pixel. 1: 6GHz elements data 2: 10GHz elements data 3: 18GHz elements data 4: 23GHz elements data 5: 36GHz elements data 6: 50GHz elements data Geolocation variable numbers are defined as follows. 1: latitude [deg] 2: longitude [deg]
	latlon_low_mean	4byte real	2 x 196	Geolocation of the observation <i>mean</i> point for lower channels (simple mean value) Dimension: n geolocation variable x n pixel. Variable numbers are defined as follows. 1: latitude [deg] 2: longitude [deg]
	latlon_high_A	4byte real	2 x 392	Geolocation of the observation point for 89GHz channels (A-scan). Dimension: n geo-location variable x n pixel Variable numbers are defined as follows. 1: latitude [deg] 2: longitude [deg]
	latlon_high_B	4byte real	2 x 392	Geolocation of the observation point for 89GHz channels (B-scan). Dimension: n geo-location variable x n pixel Variable numbers are defined as follows. 1: latitude [deg] 2: longitude [deg]

(1) **scan_time**

“scan_time” is the structure whose type is the structure “SCAN_TIME.”

(2) **tb_low**

“tb_low,” whose dimensions are 12 x 196, is the brightness temperature (TB data) of the lower frequency channels. The size “12” means the number of lower channel variables. The first element is 6GHz-Vertical data, the second is 6GHz-Horizontal data, the third is

10GHz-Vertical data,...., the eleventh is 50GHz-Vertical data, and the twelfth is 52GHz-Vertical data.

The size “196” is the number of samples for each scan. The unit is [K].

Table 4.3.2-1 Brightness temperature data value table

value of data	meaning of data value
positive	normal data
negative	questionable data
-32768	parity error data
-9999	missing packet data

(3) tb_high_A

“tb_high_A,” whose dimensions are 2 x 392, is the data of Brightness Temperature of 89GHz channels of A-scan. “2” indicates the dimension of the polarization on the A-horn. The first element is the 89.0GHz-Vertical-A data, and the second is the 89.0GHz-Horizontal-A data.

“392” indicates the number of samples for each scan. Channel element values have the same meaning as in “tb_low.” (See Table 4.3.2-1.)

(4) tb_high_B

“tb_high_B,” whose dimensions are 2 x 392, is the data of Brightness Temperature of 89GHz channels of B-scan. “2” indicates the dimension of the polarization on the B-horn. The first element is the 89.0GHz-Vertical-B data, and the second is the 89.0GHz-Horizontal-B data.

“392” indicates the number of samples for each scan. Channel element values have the same meaning as in “tb_low.” (See Table 4.3.2-1.)

(5) latlon_low, latlon_low_mean

“latlon_low” is the latitude and longitude of the observation point in a scan for each lower frequency channels, “latlon_low_mean” has representative value (simple mean) of latitude and longitude for all lower frequency channels. There are 196 points in a scan. “latlon_low” and “latlon_low_mean” are in degrees. The latitude ranges from -90 to 90; positive value is north latitude, and negative value is south latitude. The longitude ranges from -180 to 180. (See Table 4.3.2-2 and Table 4.3.2-3.)

Table 4.3.2-2 Latitude data value table

value of data	meaning of data value
-90 ~ 0	north latitude data
0 ~ 90	south latitude data
-9999	missing packet data

Table 4.3.2-3 Longitude data value table

value of data	meaning of data value
-180 ~ 0	west longitude data
0 ~ 180	east longitude data
-9999	missing packet data

(6) latlon_high_A

“latlon_high_A” is the latitude and longitude of the observation point in a scan for 89GHz A-scan. There are 392 points in a scan. “latlon_high_A” has units of [deg]. The latitude ranges from -90 to 90, positive value is north latitude, and negative value is south latitude. The longitude ranges from -180 to 180. (See Table 4.3.2-2 and Table 4.3.2-3.)

(7) latlon_high_B

“latlon_high_B” is the latitude and longitude of the observation point in a scan for 89GHz Bb-scan. There are 392 points in a scan. “latlon_high_B” has units of [deg]. The latitude ranges from -90 to 90, positive value is north latitude, and negative value is south latitude. The longitude ranges from -180 to 180. (See Table 4.3.2-2 and Table 4.3.2-3.)

4.3.3 SUN_EARTH (for L1B)

Table 4.3.3-1 SUN_EARTH

Name of structure	member	type	size	Description
SUN_EARTH	sun_azimuth	4byte real	196	Sun azimuth angle [deg]
	sun_elev	4byte real	196	Sun elevation angle [deg]
	earth_incid	4byte real	196	Earth incident angle [deg]
	earth_azimuth	4byte real	196	Earth azimuth angle [deg]
	ol_flag	2byte int	7 x 196	Ocean/Land flag

(1) sun_azimuth

“sun_azimuth” is the Sun azimuth angle at an observation point. The definition is shown in Fig 4.7.3-1 and the range is 360 degree. This data is calculated corresponding to the observation points of 6.GHz to 36GHz. This value is calculated for the representative point of the lower frequency channels (e.g. latitudes and longitudes in “latlon_low_mean.”)

(2) sun_elev

“sun_elev” is the Sun elevation angle at an observation point. The definition is shown in Fig 4.7.3-1 and the range is -90.0 to 180 degrees. Calculated values less than -90.0 degrees will be set to -32687, calculated values exceeding 180 degrees will be set to 32768. For other errors case, it will be set to -32768. The data calculated corresponding to the observation points of 6GHz to 36GHz. This value is calculated for the representative point of the lower frequency channels (e.g. latitudes and longitudes in “latlon_low_mean.”)

(3) earth_incid

“earth_incid” is the Earth incidence angle at an observation point. The definition is shown in Fig 4.7.3-2 and the range is -90.0 to 180 degrees. Calculated value less than -90.0 will be set to -32687, Calculated values exceeding 180 degrees will be set to 32768. For other errors will be set to -32768. The data calculated corresponding to the observation points of 6GHz to 36GHz. This value is calculated for the representative point of the lower frequency channels (e.g. latitudes and longitudes in “latlon_low_mean.”)

(4) earth_azimuth

“earth_azimuth” is the Earth azimuth angle, which is defined as the angle between the north vector and the observation direction vector of AMSR/AMSR-E at an observation point. The definition is shown Fig 4.7.3-2. This data is calculated corresponding to the observation points of 6GHz to 36GHz. This value is calculated for the representative point of the lower frequency channels (e.g. latitudes and longitudes in “latlon_low_mean.”)

(4) ol_flag

“ol_flag” is a ratio of land area in the main beam footprint (3dB down beam width) and is expressed on percentage. The data range is from 0 to 100, in abnormal case, data is set to 255. There are 196 stored points in a scan, and these data corresponds to the footprints of 6GHz, 10GHz, 18GHz, 23GHz, 36GHz, 50GHz, and 89GHz-A.

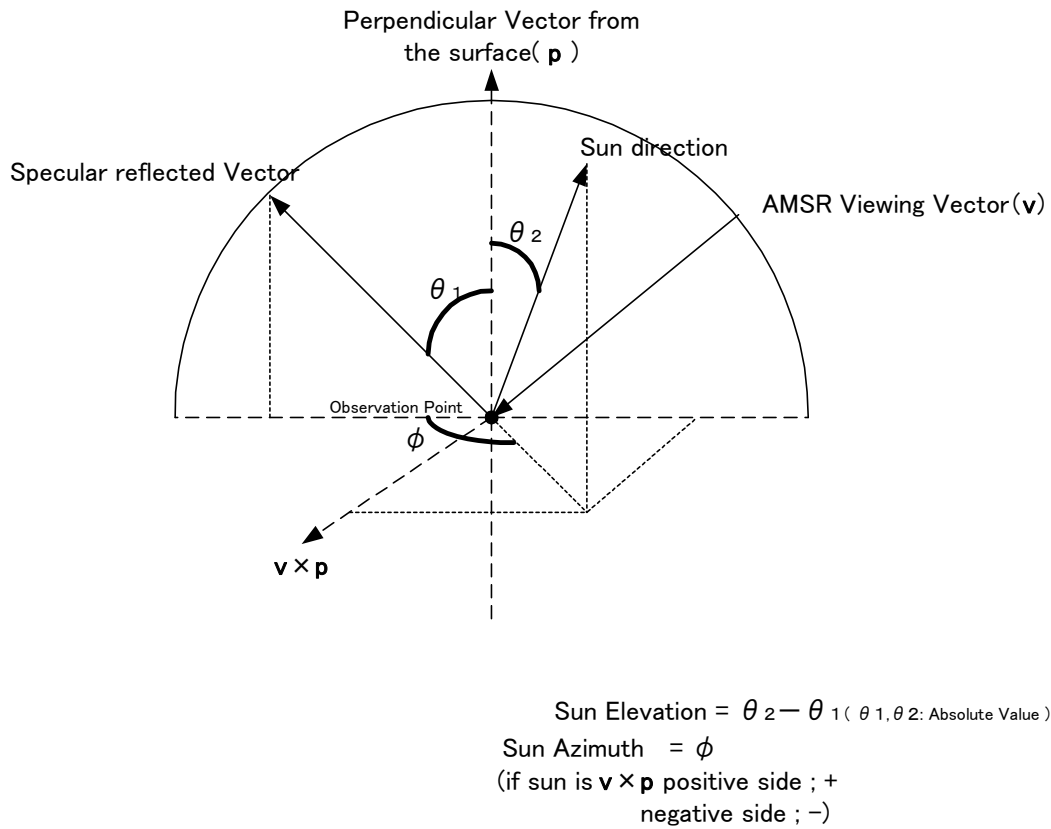


Fig 4.7.3-1 Definition of Sun Elevation/Azimuth

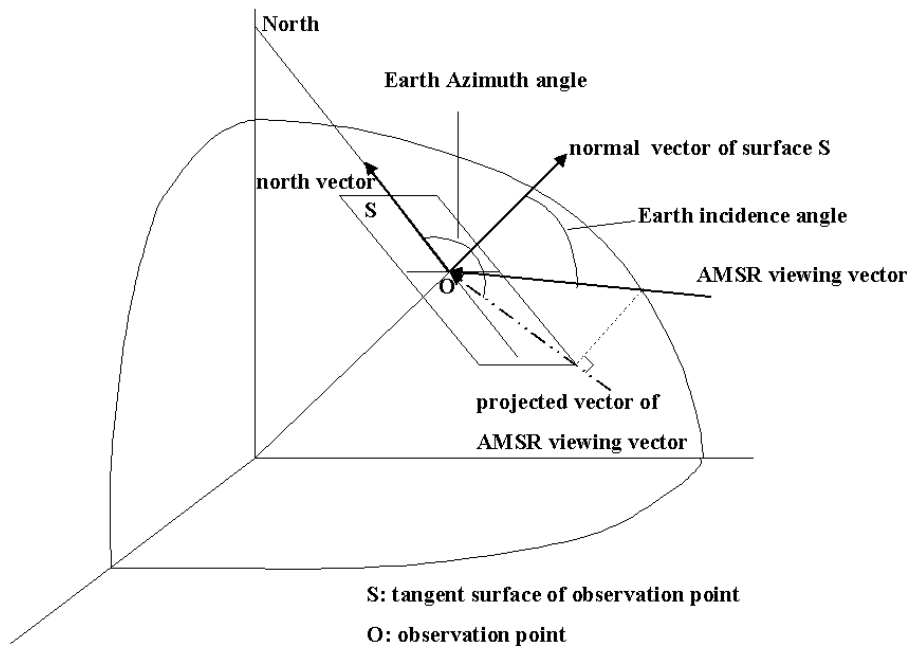


Fig 4.7.3-2 Definition of Earth Azimuth/Incidence

4.3.4 STATUS_L1B (for L1B)

Table 4.3.4-1 STATUS_L1B

Name of structure	member	type	size	Description
STATUS_L1B	pos_orbit	8byte real	1	Orbit No.
	obs_supple	unsigned 2byte int	27	Observation supplement data Dimension: n supplement kind Contents of this array are defined as in Table 4.3.4-2.
	gpsr	2byte int	1	Check value of the GPSR count (0:OK,1:NG) The checking conclusion of GPSR count In the case that the difference of GPSR in before scan and after scan is not satisfied $1.5 \pm 1.0\text{sec}$ or $-6.5 \pm 1.0\text{sec}$, this flag will be 1.
	hts	2byte int	1	Check value of the HTS count (0:OK,1:NG) The checking conclusion of HTS temperature. In the case that the difference of HTS temperature in before scan and after scan is not satisfied within 0.5° , this flag will be set 1.
	moon_azimuth	4byte real	1	Moon azimuth [deg] The moon direction from Cold Sky Mirror (See Fig 4.3.4-1)
	sun_azimuth	4byte real	1	Sun azimuth [deg] The sun direction from Cold Sky Mirror (See Fig 4.3.4-1)
	tacopulse	4byte real	1	Taco pulse count [count] The average data of Taco pulse count in a product
	quality	4byte real	16 x 4	Statistic values of calibration data Dimension: n channel x n statistic value For AMSR-E, 50GHz vertical and 52GHz vertical element are set to 0.0. Variable numbers are defined as follows. (for n channel) 1: 6GHz vertical elements 2: 6GHz horizontal elements 3: 10GHz vertical elements 4: 10GHz horizontal elements 5: 18GHz vertical elements 6: 18GHz horizontal elements 7: 23GHz vertical elements 8: 23GHz horizontal elements 9: 36GHz vertical elements 10: 36GHz horizontal elements 11: 50GHz vertical elements 12: 52GHz vertical elements 13: 89GHz A-horn vertical elements 14: 89GHz A-horn horizontal elements 15: 89GHz B-horn vertical elements 16: 89GHz B-horn horizontal elements Variable numbers are defined as follows. (for n statistic value) 1: Cold Sky Mirror Count mean value [count] 2: Hot-load Count mean value [count] 3: Cold Sky Mirror Count root mean square [count] 4: Hot-load Count root mean square [count]

(1) pos_orbit

This data expresses the scanning position in an orbit and is stored in every scan.

Example: The value of “pos_orbit” 100.5 denotes the middle point between orbit number 100. and 101.

(2) obs_supple

“obs_supple” is included in AMSR and AMSR-E telemetry data. This data is stored in every scan. The details of this data are shown in the Table 4.3.4-2.

(3) quality

“quality” is statistic values of calibration data about Cold Sky Mirror Count and Hot-load Count for AMSR and AMSR-E data in every scan. This statistic data contains mean value and root mean square value in unit [count].

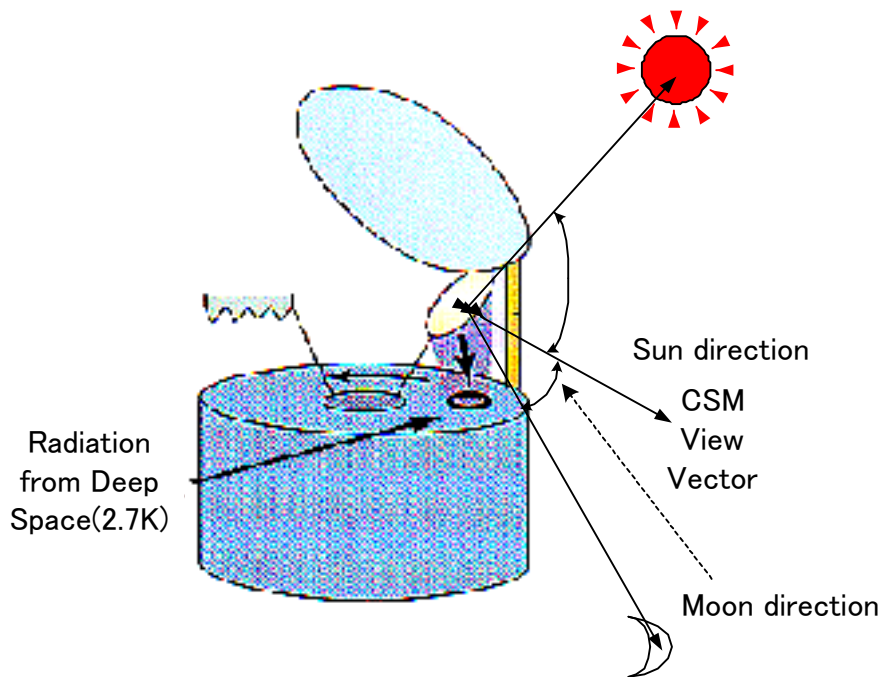


Fig 4.3.4-1 Definition of Sun/Moon direction

Table 4.3.4-2 "obs_supple" data table

Observation supplements array NO.	Description
1	GPSR (Global Positioning System Receiver) count
2	Taco pulse count #1
3	Taco pulse count #2

4	Taco pulse count #3
5	Taco pulse count #4
6	Taco pulse count #5
7	SPC (Signal Processor Control Unit) ON/OFF #1
8	SPC (Signal Processor Control Unit) ON/OFF #2
9	SPC (Signal Processor Control Unit) operation flag
10	SPC (Signal Processor Control Unit) error flag #1
11	SPC (Signal Processor Control Unit) error flag #2
12	SPC (Signal Processor Control Unit) error flag #3
13	SPC (Signal Processor Control Unit) error flag #4
14	Redundancy Switching Control #1
15	Redundancy Switching Control #2
16	SPS(Signal Processor Sensor Unit) ON/OFF #1
17	SPS(Signal Processor Sensor Unit) ON/OFF #2
18	SPS(Signal Processor Sensor Unit) ON/OFF #3
19	SPS(Signal Processor Sensor Unit) ON/OFF #4
20	SPS(Signal Processor Sensor Unit) operation mode
21	RX AGC (Auto Gain Control)/MGC (Manual Gain Control) mode #1
22	RX AGC (Auto Gain Control)/MGC (Manual Gain Control) mode #2
23	SPS(Signal Processor Sensor Unit) operation flag
24	SPS(Signal Processor Sensor Unit) error flag #1
25	SPS(Signal Processor Sensor Unit) error flag #2
26	SPS(Signal Processor Sensor Unit) error flag #3
27	SPS(Signal Processor Sensor Unit) error flag #4

4.3.5 CAL (for L1B)

Table 4.3.5-1 CAL

Name of structure	member	type	size	Description
CAL	ahotload_low	2byte int	12 x 8	Hot-load count for lower frequency channels (AMSR) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 6GHz vertical elements data [count] 2: 6GHz horizontal elements data [count] 3: 10GHz vertical elements data [count] 4: 10GHz horizontal elements data [count] 5: 18GHz vertical elements data [count] 6: 18GHz horizontal elements data [count] 7: 23GHz vertical elements data [count] 8: 23GHz horizontal elements data [count] 9: 36GHz vertical elements data [count] 10: 36GHz horizontal elements data [count] 11: 50GHz vertical elements data [count] 12: 52GHz vertical elements data [count]
	ahotload_high_A	2byte int	2 x 16	Hot-load count for 89GHZ channels A-scan (AMSR) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz A-horn vertical elements data [count] 2: 89GHz A-horn horizontal elements data [count]
	ahotload_high_B	2byte int	2 x 16	Hot-load count for 89GHZ channels B-scan (AMSR) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz B-horn vertical elements data [count] 2: 89GHz B-horn horizontal elements data [count]
	acoldsky_low	2byte int	12 x 8	Cold sky mirror count for lower frequency channels

Table 4.3.5-1 CAL

Name of structure	member	type	size	Description
				(AMSR) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 6GHz vertical elements data [count] 2: 6GHz horizontal elements data [count] 3: 10GHz vertical elements data [count] 4: 10GHz horizontal elements data [count] 5: 18GHz vertical elements data [count] 6: 18GHz horizontal elements data [count] 7: 23GHz vertical elements data [count] 8: 23GHz horizontal elements data [count] 9: 36GHz vertical elements data [count] 10: 36GHz horizontal elements data [count] 11: 50GHz vertical elements data [count] 12: 52GHz vertical elements data [count]
	acoldsky_high_A	2byte int	2 x 16	Cold sky mirror count for 89GHZ channels A-scan (AMSR) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz A-horn vertical elements data [count] 2: 89GHz A-horn horizontal elements data [count]
	acoldsky_high_B	2byte int	2 x 16	Cold sky mirror count for 89GHZ channels B-scan (AMSR) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz B-horn vertical elements data [count] 2: 89GHz B-horn horizontal elements data [count]
	ehotload_low	2byte int	12 x 16	Hot-load count for lower frequency channels (AMSR-E) Dimension: n channel x n pixel Variable numbers are defined as follows. AMSR-E data does not have 50GHz and 52GHz frequency bands, therefore these two band data are set to <i>zero</i> in every scan and pixel. 1: 6GHz vertical elements data [count] 2: 6GHz horizontal elements data [count] 3: 10GHz vertical elements data [count] 4: 10GHz horizontal elements data [count] 5: 18GHz vertical elements data [count] 6: 18GHz horizontal elements data [count] 7: 23GHz vertical elements data [count] 8: 23GHz horizontal elements data [count] 9: 36GHz vertical elements data [count] 10: 36GHz horizontal elements data [count] 11: 50GHz vertical elements data [count] 12: 52GHz vertical elements data [count]
	ehotload_high_A	2byte int	2 x 32	Hot-load count for 89GHZ channels A-scan (AMSR-E) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz A-horn vertical elements data [count] 2: 89GHz A-horn horizontal elements data [count]
	ehotload_high_B	2byte int	2 x 32	Hot-load count for 89GHZ channels B-scan (AMSR-E) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz B-horn vertical elements data [count] 2: 89GHz B-horn horizontal elements data [count]

Table 4.3.5-1 CAL

Name of structure	member	type	size	Description
	ecoldsky_low	2byte int	12 x 16	Cold sky mirror count for lower frequency channels (AMSR-E) Dimension: n channel x n pixel Variable numbers are defined as follows. AMSR-E does not have 50GHz and 52GHz frequency bands, therefore these two band data are set to <i>zero</i> in every scan and pixel. 1: 6GHz vertical elements data [count] 2: 6GHz horizontal elements data [count] 3: 10GHz vertical elements data [count] 4: 10GHz horizontal elements data [count] 5: 18GHz vertical elements data [count] 6: 18GHz horizontal elements data [count] 7: 23GHz vertical elements data [count] 8: 23GHz horizontal elements data [count] 9: 36GHz vertical elements data [count] 10: 36GHz horizontal elements data [count] 11: 50GHz vertical elements data [count] 12: 52GHz vertical elements data [count]
	ecoldsky_high_A	2byte int	2 x 32	Cold sky mirror count for 89GHZ channels A-scan (AMSR-E) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz A-horn vertical elements data [count] 2: 89GHz A-horn horizontal elements data [count]
	ecoldsky_high_B	2byte int	2 x 32	Cold sky mirror count for 89GHZ channels B-scan (AMSR-E) Dimension: n channel x n pixel Variable numbers are defined as follows. 1: 89GHz B-horn vertical elements data [count] 2: 89GHz B-horn horizontal elements data [count]
	ant_temp_coef	4byte real	32	Antenna Temperature Coefficient for every channel in a scan Dimension: n channel offset/gain Variables are listed in Table 4.3.5-4. The unit of gain is [K/count] and the unit of offset is [K].
	offset_gain	unsigned 2byte int	32	Rx Offset/Gain Count for every channel in a scan Dimension: n channel offset/gain Variables are listed in Table 4.3.5-1. Unit is [count].
	SPC_temp_cnt	2byte int	20	Temperature counts of Signal processor control unit array style n kind Variables are listed in Table 4.3.5-5. Unit is [count].
	SPS_temp_cnt	2byte int	32	Temperature counts of Signal processor sensor unit array style n kind Variable is defined as Table 4.3.5-6. Unit is [count].
	SPC_temp_calc	8byte real	20	Temperature of Signal processor control unit calculated from "SPC_temp_cnt." Variables are listed in Table 4.3.5-5. Unit is [°C].
	SPS_temp_calc	8byte real	32	Temperature of Signal processor sensor unit calculated from "SPS_temp_cnt." Variables are listed in Table 4.3.5-6. Unit is [°C].

(1) a[e]hotload_low, a[e]hotload_high_A[B]

There are 8 (AMSR) or 16 (AMSR-E) points in one scan for lower frequency channels and 16 (AMSR) or 32 (AMSR-E) points in one scan for 89GHz channels. Hot-load count data are observed digital counts of the High Temperature nose Source (HTS, e.g., hot load) in a scan.

If you use AMSR L1B, ADIT uses automatically the members of “acoldsky_low,” “ahotload_high_A,” and “ahotload_high_B.” In using AMSR-E L1B, ADIT uses automatically the members of “ehotload_low,” “ehotload_high_A” and “ehotload_high_B”. The meaning of each lower frequency channel element’s value is shown in Table 4.3.5-2.

Table 4.3.5-2 Hot-load counts data value

value of data	meaning of data value
positive	normal data
negative	questionable data
-32768	parity error data
0	missing packet data

(2) a[e]coldsky_low, a[e]coldsky_high_A[B]

There are 8 (AMSR) or 16 (AMSR-E) points in one scan for lower frequency channels and 16 (AMSR) or 32 (AMSR-E) points in one scan for 89GHz channels. Cold Sky Mirror Count data are observed digital counts of Deep space (Cosmic Microwave Background) using the Clod Sky Mirror in a scan.

If you use AMSR L1B, ADIT uses automatically the members of “acoldsky_low,” “acoldsky_high_A“ and “acoldsky_high_B.” In using AMSR-E L1B, ADIT uses automatically the members of “ecoldsky_low,” “ecoldsky_high_A” and “ecoldsky_high_B.” The meaning of each lower frequency channel element’s value is shown in Table 4.3.5-3.

Table 4.3.5-3 Cold sky mirror counts data value

value of data	meaning of data value
positive	questionable data
negative	normal data
32767	parity error data
0	missing packet data

(3) ant_temp_coef

“ant_temp_coef” is the coefficient for converting from observation counts to antenna temperature. The coefficients are the slope and offset for every frequency and polarization channel, and are stored in every scan. This data array is defined in Table 4.3.5-4.

(4) offset_gain

“offset_gain” is the receiver offset/gain data measured every scan. This data array is

defined in Table 4.3.5-4.

(5) SPC_temp_cnt, SPC_temp_calc

“SPC_temp_cnt” is the temperature count data of the signal-processor control unit. The “SPC_temp_calc” is calculated physical temperatures from “SPC_temp_cnt.” Contents of this data are listed in Table 4.3.5-5.

(6) SPS_temp_cnt, SPS_temp_calc

“SPS_temp_cnt” is the temperature count data of the signal-processor sensor unit. The “SPS_temp_calc” is calculated value from “SPS_temp_cnt.” Contents of this data are listed in Table 4.3.5-6.

Table 4.3.5-4” ant_temp_coef”/” offset_gain” data table

Variable No. of n channel offsetgain	Description
1	6GHz vertical elements of offset
2	6GHz vertical elements of gain [slope]
3	6GHz horizontal elements of offset
4	6GHz horizontal elements of gain [slope]
5	10GHz vertical elements of offset
6	10GHz vertical elements of gain [slope]
7	10GHz horizontal elements of offset
8	10GHz horizontal elements of gain [slope]
9	18GHz vertical elements of offset
10	18GHz vertical elements of gain [slope]
11	18GHz horizontal elements of offset
12	18GHz horizontal elements of gain [slope]
13	23GHz vertical elements of offset
14	23GHz vertical elements of gain [slope]
15	23GHz horizontal elements of offset
16	23GHz horizontal elements of gain [slope]
17	36GHz vertical elements of offset
18	36GHz vertical elements of gain [slope]
19	36GHz horizontal elements of offset
20	36GHz horizontal elements of gain [slope]
21	50GHz vertical elements of offset
22	50GHz vertical elements of gain [slope]
23	52GHz vertical elements of offset
24	52GHz vertical elements of gain [slope]
25	89GHz A-horn vertical elements of offset
26	89GHz A-horn vertical elements of gain [slope]
27	89GHz A-horn horizontal elements of offset
28	89GHz A-horn horizontal elements of gain [slope]
29	89GHz B-horn vertical elements of offset
30	89GHz B-horn vertical elements of gain [slope]
31	89GHz B-horn horizontal elements of offset
32	89GHz B-horn horizontal elements of gain [slope]

Table 4.3.5-5 "SPC_temp_cnt"/"SPC_temp_calc" data table

Variable No. of n kind	Description
1	Thermistor #1 SPC A temperature
2	Thermistor #2 SPC B temperature
3	Thermistor #3 TCC temperature
4	Thermistor #4 PDUC temperature
5	Thermistor #5 ADASTATOR temperature
6	Thermistor #7 MWA Wheel temperature
7	Thermistor #8 MWA Bearing temperature
8	Thermistor #9 ADE temperature
9	Thermistor #11 Control STR temperature
10	Thermistor #12 Control STR temperature
11	Thermistor #13 Control STR temperature
12	Thermistor #14 Control STR temperature
13	Platinum sensor #1 HTS temperature 1
14	Platinum sensor #2 HTS temperature 2
15	Platinum sensor #3 HTS temperature 3
16	Platinum sensor #4 HTS temperature 4
17	Platinum sensor #5 HTS temperature 5
18	Platinum sensor #6 HTS temperature 6
19	Platinum sensor #7 HTS temperature 7
20	Platinum sensor #8 HTS temperature 8

Table 4.3.5-6 "SPS_temp_cnt"/"SPS_temp_calc" data table

n kind variable No.	Description
1	Thermistor #1 SPS temperature
2	Thermistor #2 PUDC temperature
3	Thermistor #3 TCS temperature
4	Thermistor #4 DC/DC RX 1 temperature
5	Thermistor #5 DC/DC RX 2 temperature
6	Thermistor #6 6G LNA temperature
7	Thermistor #7 10G LNA temperature
8	Thermistor #8 50G LNA temperature
9	Thermistor #9 89G H LNA1 temperature
10	Thermistor #10 89G H LNA2 temperature
11	Thermistor #11 89G V LNA1 temperature
12	Thermistor #12 89G V LNA2 temperature
13	Thermistor #13 Sensor STR3 temperature
14	Thermistor #14 Control STR4 temperature
15	Thermistor #15 ADA ROT A temperature
16	Thermistor #16 ADA ROT B temperature
17	Platinum sensor #1 6G RX temperature
18	Platinum sensor #2 10G RX temperature
19	Platinum sensor #3 18G RX temperature
20	Platinum sensor #4 23G RX temperature
21	Platinum sensor #5 36G RX temperature
22	Platinum sensor #6 50G RX temperature
23	Platinum sensor #7 89G RX1 temperature
24	Platinum sensor #8 89G RX2 temperature
25	Platinum sensor #9 MREF 1 temperature
26	Platinum sensor #10 MREF 2 temperature
27	Platinum sensor #11 MREF 3 temperature
28	Platinum sensor #12 MREF 4 temperature
29	Platinum sensor #13 FEED 1 temperature

Table 4.3.5-6 "SPS_temp_cnt"/"SPS_temp_calc" data table

n kind variable No.	Description
30	Platinum sensor #14 FEED 2 temperature
31	Platinum sensor #15 sensor STR1 temperature
32	Platinum sensor #16 sensor STR2 temperature

4.3.6 NAVI (for L1B)

"NAVI" is the structure of the navigation data of the platform. The structure is defined in Table 4.3.6-1.

Table 4.3.6-1 NAVI

Name of structure	member	type	size	Description
NAVI	posX	4byte real	1	platform position in X coordinate [m]
	posY	4byte real	1	platform position in Y coordinate [m]
	posZ	4byte real	1	platform position in Z coordinate [m]
	velX	4byte real	1	platform velocity in X coordinate [m/s]
	velY	4byte real	1	platform velocity in Y coordinate [m/s]
	velZ	4byte real	1	platform velocity in Z coordinate [m/s]
	roll	4byte real	1	platform attitude of roll angle [deg]
	pitch	4byte real	1	platform attitude of pitch angle [deg]
yaw	4byte real	1	platform attitude of yaw angle [deg]	

(1) NAVI

The structure "NAVI" contains position, velocity and attitude data of the platform. Position and velocity data are expressed in an inertia co-ordinate system and stored corresponding to the structure "SCAN_TIME." The unit of position ("posX," "posY," "posZ") is [m], and velocity ("velX," "velY," "velZ") is [m/s]. Three kinds of navigation data are used to acquire position data and velocity data, GPS, ELMD and ELMP. Metadata (attribute name is "EphemerisType") specifies which data is stored.

Attitude data ("roll," "pitch," "yaw") have units of [deg]. The value "roll" is the direction of flight, "yaw" is the direction of the nadir, and "pitch" is the direction of "yaw" x "Roll."

Table 4.3.6-2 Navigation data value table

value of data	meaning of data value
except -9999	normal data
-9999	missing packet data

4.3.7 AMSRL2_SWATH (for L2)

Table 4.3.7-1 AMSRL2_SWATH

Name of structure	member	type	size	Description
AMSRL2_SWATH	scan_time	SCAN_TIME	20	Structure of SCAN_TIME

geophys	4byte real	3 x 196	Geophysical data in a scan. Dimension: n rank x n pixel Variable numbers are defined as follows. 1: geophysical data 2: depends on PI 3: depends on PI
latlon_low	4byte real	2 x 196	Geolocation of the observation <i>mean</i> point for lower channels (simple mean value) Dimension: n geolocation variable x n pixel. Variable numbers are defined as follows. 1: latitude [deg] 2: longitude [deg]

(1) scan_time

“scan_time” is the structure “SCAN_TIME.”

(2) geophys

“geophys” is the geophysical data in a scan. There are several kinds of geophysical parameters. (See Table 4.3.7-2.)

Table 4.3.7-2 Geophysical quantity parameters and L2 product code

geophysical parameters	product code	unit	maximum value	minimum value
Water Vapor	WV0	kg/m ²	0	70
Cloud Liquid Water	CLW	kg/m ²	0	1.0
Amount of Precipitation	AP0	mm/h	0	100
Sea Surface Wind	SSW	m/s	0	30
Sea Surface Temperature	SST	°C	-2	35
Ice Concentration	IC0	%	0	100
Soil Moisture	SM0	g/cm ³	0	To be defined
Snow Water Equivalence	SWE	mm	0	10000

(3) latlon_low

“latlon_low” includes the latitude and longitude of the representative observation point for lower frequency channels in a scan. There are 196 points in a scan. The “latlon_low” has units of [deg]. The latitude ranges from -90 to 90, positive value is north latitude, and negative value is south latitude. The longitude ranges from -180 to 180. (See Table 4.3.2-2 and Table 4.3.2-3.)

4.3.8 STATUS_L2 (for L2)

Table 4.3.8-1 STATUS_L2

Name of struct	member	type	size	Description
STATUS_L2	pos_orbit	4byte real	1	Orbit No.
	quality	1byte int	3 x 196 x 8	Quality flag corresponding to each point of geophysical quantity data. Dimension: n rank x n pixel n bit-position Variable numbers of n rank are defined as follows. 1: geophysical data quality flag 2: depends on PI ^(*) 3: depends on PI ^(*) Variable numbers of n bit-position are shown in Table 4.3.8-2.

(*1) PI: Proposal Instructor

(1) pos_orbit

This data express the scanning position in an orbit and is stored every scan.

Example: “pos_orbit” 100.5 denotes the middle point between orbit number 100. and 101.

(2) quality

“quality” is the quality flag for L2 data in every scan. (See Table 4.3.8-2.)

Table 4.3.8-2 Quality Flag in detail

Data	Bit position							
	7	6	5	4	3	2	1	0
WV	Land/coast	Abnormal brightness temperature	Sea ice	Abnormal supplementary-sea_surface temperature-wind at sea-temperature of 850hPa	Abnormal calculation of sea_surface emissivity	Cloud	Rainfall	Low precision
CLW	IRETX(2) means no retrieval was done	ISUR2 means land contamination	ICE means sea ice	IOOB(2) means TB OOB	Unused	Unused	Unused	Unused
AP	Tb OK/Bad Tb	no rain/light rain	no rain/heavier rain	retrieval done/no retrieval	Unused	Unused	Unused	Unused
SSW	Land area	Sea ice	Sun glitter	Rain	no data of w6 in correcting wind direction	incident angle error	abnormal wind speed	not used
SST	Land area	Sea ice	Sun glitter	Rain	Wind	Incident angle	Abnormal SST	Not enough number for average TB
IC	No calculation took place	Invalid brightness temperature	Land location	Latitude is out of ice range	Pixel is out of sea area	High SST	Unused	Unused
SWE	0:No snow (normal retrieval) 1:Water 2:Snow impossible 3:Permanent ice 4:Surface temperature too warm 5:Heavy forest 6:Mountainous region 7:Rain 8:Wet snow 9:Dry snow (currently unused) 10:Wet soil 11:Dry soil (currently unused) 12:Tb out of range 13:Snow possible 14:Satellite attitude out of range * 15:Missing Tb values *							
SM	Unused							

4.3.9 L3 Science data

There is no structure defined in ADIT for L3. But ADIT provides L3 science data as a 4byte real data, whose size is corresponding to geophysical parameters and map projection type. (See Table 4.3.9-1.)

Table 4.3.9-1 L3 science data size

geophysical parameters	product code	map projection type	size		unit
			line	x pixel	
Brightness Temperature	TB	Equirectangular	721	x 1440	K
		Polar stereo in the northern hemisphere	448	x 304	
		Polar stereo in the southern hemisphere	332	x 316	
Water Vapor	WV0	Equirectangular	721	x 1440	kg/m ²
Cloud Liquid Water	CLW	Equirectangular	721	x 1440	kg/m ²
Amount of Precipitation	AP0	Equirectangular	721	x 1440	mm/h
Sea Surface Wind	SSW	Equirectangular	721	x 1440	m/s
Sea Surface Temperature	SST	Equirectangular	721	x 1440	°C
Ice Concentration	IC0	Polar stereo in the northern hemisphere	448	x 304	%
		Polar stereo in the southern hemisphere	332	x 316	
Soil Moisture	SM0	Equirectangular	721	x 1440	g/cm ³
Snow Water Equivalence	SWE	Equirectangular	721	x 1440	cm
		Polar stereo in the southern hemisphere	573	x 431	mm

4.4 Metadata

4.4.1 L1B Metadata

Table 4.4.1-1 L1B Metadata

metadata index	metadata name	Description	metadata values (example)
0	ShortName	product name	AMSREL1B
1	VersionID	product version ID	RELEASE2
2	SizeMBECSDDataGranule	product size (MB)	31.4
3	LocalGranuleID	Local Granule ID	P1AME030203164MA_P01B000000
4	ProcessingLevelID	Processing Level ID	L1B
5	ReprocessingActual	Reprocessing Actual (UTC)	Blank
6	ProductionDateTime	Production Date Time (UTC)	2004-11-29T04:30:39.000Z
7	RangeBeginningTime	Range Beginning Time (UTC)	23:00:19.17Z
8	RangeBeginningDate	Range Beginning Date (UTC)	2003/2/3
9	RangeEndingTime	Range Ending Time (UTC)	23:50:18.94Z
10	RangeEndingDate	Range Ending Date (UTC)	2003/2/3
11	GringPointLatitude	Gring Point of Latitude	-74.35,-84.96,-35.13,24.82,74.89,86.64,26.87,-32.87
12	GringPointLongitude	Gring Point of Longitude	-33.90,23.41,-148.70,-161.55,127.15,177.88,-147.18,-132.87
13	PGENAME	Name of L1B Process Software	L1B_Process_Software
14	PGEVersion	Version of L1B Process Software	222*22****22220222
15	InputPointer	Inputted file name	R1540402AGS0303423394200.RBD,R1540402SGS0303501282600.RBD
16	ProcessingCenter	Data Processing Center	JAXA EOC
17	ContactOrganizationName	Contact Organization Name	JAXA,1401,Ohashi,Hatoyama-machi,Hiki-gun,Saitama,350-0393,JAPAN,+81-49-298-1307,orderdesk@eoc.jaxa.go.jp
18	StartOrbitNumber	Start Orbit Number	4015
19	StopOrbitNumber	Stop Orbit Number	4016
20	EquatorCrossingLongitude	Equator Crossing Longitude	-148.8
21	EquatorCrossingDate	Equator Crossing Date	2003/2/3
22	EquatorCrossingTime	Equator Crossing Time	23:27:28.11Z
23	OrbitDirection	Orbit Direction	ASCENDING
24	EphemerisGranulePointer	Ephemeris Granule Pointer	R1540957AGS0303423394200.RBD

Table 4.4.1-1 L1B Metadata

metadata index	metadata name	Description	metadata values (example)
25	EphemerisType	Ephemeris Type	ELMP
26	PlatformShortName	Platform Short Name	EOS-PM1
27	SensorShortName	Sensor Short Name	AMSR-E
28	NumberOfScans	Number of Scan	2001
29	NumberOfMissingScans	Number of Missing Scan	0
30	ECSDataModel	ECS Data Model(name of metadata model)	B.0
31	DiscontinuityVirtualChannelCounter	Virtual channel Unit Counter Discontinuity	DEAD Encounter
32	QALocationPacketDiscontinuity	Packet Sequence Counter Discontinuity	continuation
33	NumberOfPackets	Number of Packets of L0 data	32016
34	NumberOfInputFiles	Number of Input L0 Files	2
35	NumberOfMissingPackets	Number Missing Packets	0
36	NumberOfGoodPackets	Number of Good Packets	32016
37	ReceivingCondition	Receiving Condition	Blank
38	EphemerisQA	Ephemeris limit check	OK
39	AutomaticQAFlag	Automatic QA Flag check	PASS
40	AutomaticQAFlagExplanation	Automatic QA Flag Explanation	1.MissingDataQA:Less than 20 is available->OK,2.AntennaRotationQA:Less than 20 is available->OK,3.HotCalibrationSourceQA:Less than 20 is available->OK,4.AttitudeDataQA:Less than 20 is available->OK,5.EphemerisDataQA:Less than 20 is available->OK,6.QualityofGeometricInformationQA:Less than 0 is available->OK,7.BrightnessTemperatureQA:Less than 20 is available->OK,All items are OK,'PASS' is employed
41	ScienceQualityFlag	Science Data calculation Quality Flag	
42	ScienceQualityFlagExplanation	Science Data calculation Quality Description	
43	QAPercentMissingData	QA Percent of Missing Data	0
44	QAPercentOutofBoundsData	QA Percent Out of Bounds Data	0

Table 4.4.1-1 L1B Metadata

metadata index	metadata name	Description	metadata values (example)
45	QAPercentParityErrorData	Number of QA Percent Parity Error Data	0
46	ProcessingQADescription	Processing QA Description	PROC_COMP
47	ProcessingQAAttribute	Processing QA Attribute Name	
48	SatelliteOrbit	Satellite Orbit	Sun-synchronous_sub-recurrent
49	Altitude	Satellite Altitude	707.9km
50	OrbitSemiMajorAxis	Satellite Orbit Semi Major Axis	7085.858km
51	OrbitEccentricity	Satellite Orbit Eccentricity	0.00095
52	OrbitArgumentPerigee	Satellite Orbit Argument Perigee	106.480deg
53	OrbitInclination	Satellite Orbit Inclination	98.15deg
54	OrbitPeriod	Satellite Orbit Period	98minutes
55	RevisitTime	Revisit Time	16days
56	AMSRChannel	AMSR/AMSR-E Channel	6.925GHz,10.65GHz,18.7GHz,23.8GHz,36.5GHz,89.0GHz-A,89.0GHz-B
57	AMSRBandWidth	AMSR/AMSR-E Band Width	6G-350MHz,10G-100MHz,18G-200MHz,23G-400MHz,36G-1000MHz,50.3G-0,52G-0,89GA-3000MHz,89GB-3000MHz
58	AMSRBeamWidth	AMSR/AMSR-E Beam Width	6G-1.8deg,10G-1.2deg,18G-0.64deg,23G-0.75deg,36G-0.35deg,50.3G-0,52G-0,89GA-0.15deg,89GB-0.15deg
59	OffNadir	Angle of offnadir	47.0deg : 89GB, 47.5deg : others
60	SpatialResolution(AzXEI)	Spatial Resolution	6G-43.2kmX75.4km,10G-29.4kmX51.4km,18G-15.7kmX27.4km,23G-18.1kmX31.5km,36G-8.2kmX14.4km,50.3G- ,52G- ,89GA-3.7kmX6.5km,89GB-3.5kmX5.9km
61	ScanningPeriod	Scanning Period	1.5sec
62	SwathWidth	Swath Width	1450km
63	DynamicRange	Dynamic Range	2.7K-340K
64	DataFormatType	Data Format Type	NCSA-HDF
65	HDFFormatVersion	HDF Format Version	Ver4.1r2
66	EllipsoidName	Ellipsoid Model Name	WGS84
67	SemiMajorAxisofEarth	Semi Major Axis of Earth	6378.1km
68	FlatteningRatioofEarth	Flattening Ratio of Earth	0.00335
69	SensorAlignment	Sensor Alignment	Rx=0.00000,Ry=0.00000,Rz=0.00000
70	ThermistorCountRangeWx	Thermistor Count Adaptable Range Wx	#####

Table 4.4.1-1 L1B Metadata

metadata index	metadata name	Description	metadata values (example)
71	ThermistorConversionTableWa	Thermistor Conversion Table Wa	0.000000,0.000015,0.000161,0.000618,0.002331,0.011459,0.010101,0.000000
72	ThermistorConversionTableWb	Thermistor Conversion Table Wb	0.000000,0.056460,-0.109878,-0.819170,-3.801865,-20.783040,-18.212120,0.000000
73	ThermistorConversionTableWc	Thermistor Conversion Table Wc	-35.000000,-38.250000,9.220000,284.170000,1582.770000,9480.000000,8263.350000,90.000000
74	ThermistorConversionTableWd	Thermistor Conversion Table Wd	0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000
75	Platinum#1CountRangeWx	Platinum#1 Count Adaptable Range Wx	#####
76	Platinum#1ConversionTableWa	Platinum#1 Conversion Table Wa	0.000000,0.000000,0.000000,0.000000,0.000000
77	Platinum#1ConversionTableWb	Platinum#1 Conversion Table Wb	0.000000,0.039000,0.042000,0.039000,0.042000
78	Platinum#1ConversionTableWc	Platinum#1 Conversion Table Wc	-35.000000,-80.625000,-84.000000,-80.000000,-84.667000
79	Platinum#1ConversionTableWd	Platinum#1 Conversion Table Wd	0.000000,0.000000,0.000000,0.000000,0.000000
80	Platinum#2CountRangeWx	Platinum#2 Count Adaptable Range Wx	#####
81	Platinum#2ConversionTableWa	Platinum#2 Conversion Table Wa	0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000
82	Platinum#2ConversionTableWb	Platinum#2 Conversion Table Wb	0.000000,0.078300,0.078000,0.083000,0.078000,0.083000,0.085300,0.000000
83	Platinum#2ConversionTableWc	Platinum#2 Conversion Table Wc	-140.000000,-161.440000,-160.000000,-169.333000,-158.750000,-170.667000,-177.640000,140.000000
84	Platinum#2ConversionTableWd	Platinum#2 Conversion Table Wd	0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000
85	Platinum#3CountRangeWx	Platinum#3 Count Adaptable Range Wx	#####
86	Platinum#3ConversionTableWa	Platinum#3 Conversion Table Wa	0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000
87	Platinum#3ConversionTableWb	Platinum#3 Conversion Table Wb	0.000000,0.009100,0.009100,0.009100,0.009900,0.009900,0.008500,0.000000
88	Platinum#3ConversionTableWc	Platinum#3 Conversion Table Wc	10.000000,6.845000,6.803800,6.803800,4.719500,4.719500,9.835000,44.000000
89	Platinum#3ConversionTableWd	Platinum#3 Conversion Table Wd	0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000,0.000000
90	CoefficientAvv	Brightness Temperature Coefficient Avv	6G-1.037,10G-1.032,18G-1.025,23G-1.032,36G-1.029,50G-0.000,52G-0.000,89GA-1.025,89GB-1.029

Table 4.4.1-1 L1B Metadata

metadata index	metadata name	Description	metadata values (example)
91	CoefficientAhv	Brightness Temperature Coefficient Ahv	6G--0.003,10G--0.003,18G--0.003,23G--0.004,36G--0.004,50G-0.000,52G-0.000,89GA--0.003,89GB--0.004
92	CoefficientAov	Brightness Temperature Coefficient Aov	6G--0.034,10G--0.029,18G--0.022,23G--0.028,36G--0.024,50G-0.000,52G-0.000,89GA--0.022,89GB--0.024
93	CoefficientAhh	Brightness Temperature Coefficient Ahh	6G-1.037,10G-1.031,18G-1.025,23G-1.034,36G-1.029,50G-0.000,52G-0.000,89GA-1.028,89GB-1.031
94	CoefficientAvh	Brightness Temperature Coefficient Avh	6G--0.003,10G--0.002,18G--0.003,23G--0.006,36G--0.004,50G-0.000,52G-0.000,89GA--0.006,89GB--0.006
95	CoefficientAoh	Brightness Temperature Coefficient Aoh	6G--0.034,10G--0.029,18G--0.022,23G--0.028,36G--0.024,50G-0.000,52G-0.000,89GA--0.022,89GB--0.024
96	CSMTemperature	Temperature of Cosmic Microwave Background(CSB) on cold sky mirror	6GV-2.800, 6GH-2.800, 10GV-2.800, 10GH-2.800, 18GV-2.800, 18GH-2.800, 23GV-2.800, 23GH-2.800, 36GV-2.800, 36GH-2.800, 50GV-0.000, 52GV-0.000, 89GAV-2.800, 89GAH-2.800, 89GBV-2.800, 89GBH-2.800
97	CoRegistrationParameterA1	Co Registration Parameter A1	6G-0.15500, 10G--0.14300, 18G--0.18200, 23G--0.19200, 36G--0.27800, 50G-0.00000
98	CoRegistrationParameterA2	Co Registration Parameter A2	6G--0.67800, 10G--0.42900, 18G--0.03100, 23G-0.18500, 36G--0.06900, 50G-0.00000
99	CalibrationCurveCoefficient#1	The radiometric correction coefficient for the 0th order	6GV--0.2099101, 6GH--0.2054645, 10GV--0.0580782, 10GH--0.0103279, 18GV--0.0853578, 18GH--0.0435186, 23GV--0.1288643, 23GH--0.1288643, 36GV--0.0475611, 36GH--0.0536047, 50GV-0.0000000, 52GV-0.0000000, 89GAV--0.0278573, 89GAH--0.0447590, 89GBV--0.0273764, 89GBH--0.0316265
100	CalibrationCurveCoefficient#2	The radiometric correction coefficient for the 1st order	6GV-1.0756783, 6GH-1.0740756, 10GV-1.0209393, 10GH-1.0037236, 18GV-1.0307711, 18GH-1.0156885, 23GV-1.0464586, 23GH-1.0464586, 36GV-1.0171470, 36GH-1.0193259, 50GV-0.0000000, 52GV-0.0000000, 89GAV-1.0100426, 89GAH-1.0161356, 89GBV-1.0098693, 89GBH-1.0114014
101	CalibrationCurveCoefficient#3	The radiometric correction coefficient for the 2nd order	6GV--0.0002537, 6GH--0.0002483, 10GV--0.0000704, 10GH--0.0000125, 18GV--0.0001022, 18GH--0.0000522, 23GV--0.0001556, 23GH--0.0001556, 36GV--0.0000575, 36GH--0.0000648, 50GV-0.0000000, 52GV-0.0000000, 89GAV--0.0000334, 89GAH--0.0000537, 89GBV--0.0000329, 89GBH--0.0000379

Table 4.4.1-1 L1B Metadata

metadata index	metadata name	Description	metadata values (example)
102	CalibrationCurveCoefficient#4	The radiometric correction coefficient for the 3rd order	6GV-0.000000, 6GH-0.000000, 10GV-0.000000, 10GH-0.000000, 18GV-0.000000, 18GH-0.000000, 23GV-0.000000, 23GH-0.000000, 36GV-0.000000, 36GH-0.000000, 50GV-0.000000, 52GV-0.000000, 89GAV-0.000000, 89GAH-0.000000, 89GBV-0.000000, 89GBH-0.000000
103	CalibrationCurveCoefficient#5	The radiometric correction coefficient for the 4th order	6GV-0.000000, 6GH-0.000000, 10GV-0.000000, 10GH-0.000000, 18GV-0.000000, 18GH-0.000000, 23GV-0.000000, 23GH-0.000000, 36GV-0.000000, 36GH-0.000000, 50GV-0.000000, 52GV-0.000000, 89GAV-0.000000, 89GAH-0.000000, 89GBV-0.000000, 89GBH-0.000000
104	CalibrationMethod	Calibration method name	RxTemperatureReferenced,SpillOver,CSMInterpolation,Absolute89GPositioning,NonlinearityCorrection
105	HTSCorrectionParameterVersion	The version number of the parameter file used for HTS correction	ver0002
106	SpillOverParameterVersion	The version number of the parameter file used for the spill over correction	ver0001
107	CSMInterpolationParameterVersion	The version number of the parameter file used for CSM interpolation correction	ver0001
108	Absolute89GPositioningParameterVersion	The version number of the parameter file used for absolute positions of 89 GHz	ver0001

4.4.2 L2 Metadata

Table 4.4.2-1 L2 Metadata

metadata index	metadata name	Description	metadata values (example)
0	ShortName	Product name	AMSR-L2
1	GeophysicalName	Geophysical quantity name	Water Vapor
2	VersionID	ID of product version	0-255
3	SizeMBECSDDataGranule	Product size (Mbyte)	30(actual)
4	Local Granule ID	Number for production management	A2AMS020101001A_P2WV0Tak111
5	ProcessingLevelID	ID of processing level	L2
6	ProductionDateTime	Time of production (UT)	2002-1-3-T00:00:00.00Z
7	RangeBeginningTime	Time to start observing (UT)	00:00:00.00Z
8	RangeBeginningDate	Date to start observing (UT)	2002-1-3
9	RangeEndingTime	Time to end observing (UT)	01:00:00.00Z
10	RangeEndingDate	Date to end observing (UT)	2002-1-3
11	GringPointLatitude	Area of interest for latitude	90
12	GringPointLongitude	Area of interest for longitude	-180
13	PGEName	Name of software	(max 20 character)
14	PGEVersion	Version of software	(max 18 character)

Table 4.4.2-1 L2 Metadata

metadata index	metadata name	Description	metadata values (example)
15	PGEAlgorithmDeveloper	Name of algorithm developer	(max 20 character)
16	InputPointer	Input file name	A2AMS02010101MD_P01B00000000.00.00
17	ProcessingCenter	Name of data processing center	HATOYAMA
18	ContactOrganizationName	Organization name to contact about this product	NASDA Address: OOAZA-OHASHI-AZA-NUMANOU HIKI-GUN SAITAMA,JAPAN Postal code : 350-0393 Telephone Number : 0492-98-1200 E-mail Address : abc@rd.tksc.nasda.go.jp Instructions : 9:20(JST) - 17 (JST) is the working time
19	StartOrbitNumber	Start orbit number	100
20	StopOrbitNumber	Stop orbit number	100
21	EquatorCrossingLongitude	Equator crossing latitude	89
22	EquatorCrossingDate	Equator crossing date	1998.2.4
23	EquatorCrossingTime	Equator crossing time	00:30:00Z
24	OrbitDirection	Orbit direction	DESCENDING
25	EphemerisGranulePointer	File name for using orbit	EPHEMERIS-1
26	EphemerisType	Type of using orbit	GPS
27	PlatformShortName	Abbreviated name of platform	ADEOS-II
28	SensorShortName	Abbreviated name of observing sensor	AMSR
29	NumberofScan	Number of scan	2020
30	ECSDataModel	Name of meta data model	B.0
31	DiscontinuityVirtualChannelCounter	Discontinuity flag of virtual channel unit counter	Continuation/Discontinuation
32	QALocationPacketDiscontinuity	Discontinuity flag of packet sequence counter	Continuation/Discontinuation
33	NumberofPackets	Number of L0 packet	32320
34	NumberofInputFiles	Number of L0 file	1
35	NumberofMissingPackets	Number of missing packet	nnnn
36	NumberofGoodPackets	Number of good packet	nnnn
37	ReceivingCondition	Condition for record or receive	
38	EphemerisQA	Result of limit check for ephemeris	GOOD
39	AutomaticQAFlag	Result by program check	OK
40	AutomaticQAFlagExplanation	Explanation of program check	PASS
41	ScienceQualityFlag	Flag when it calculates geophysical quantity	Blank for L1A,L1B,L1BMap
42	ScienceQualityFlagExplanation	Explanation when it calculate geophysical quantity	Blank for L1A,L1B,L1BMap
43	QAPercentMissingData	Number of missing data	nnn
44	QAPercentOutOfBoundsData	Ratio of data out of bound	nnn

4.4.3 L3 Metadata

Table 4.4.3-1 L3 Metadata

metadata index	metadata name	Description	metadata values (example)
0	Short Name	Product name	AMSR-L3
1	GeophysicalName	Geophysical quantity name	Water Vapor,
2	VersionID	ID of product version	0-255
3	SizeMBECSDataGranule	Product size (Mbyte)	30(actual)
4	Local Granule ID	Number for production management	A2AMS010101A_P3WV0Tak111E0
5	ProcessingLevelID	ID of processing level	L3

Table 4.4.3-1 L3 Metadata

metadata index	metadata name	Description	metadata values (example)
6	ProductionDateTime	Time of production (UT)	2002-1-3-T00:00:00.00Z
7	RangeBeginningTime	Time to start observing (UT)	00:00:00.00Z
8	RangeBeginningDate	Date to start observing (UT)	2002-1-3
9	RangeEndingTime	Time to end observing (UT)	01:00:00.00Z
10	RangeEndingDate	Date to end observing (UT)	2002-1-3
11	InputPointer	Input file name	A2AMS02010101MD_P01B00000000 00.00
12	StartOrbitNumber	Start orbit number	100
13	StopOrbitNumber	Stop orbit number	100
14	OrbitDirection	Orbit direction	DESCENDING
15	PlatformShortName	Abbreviated name of platform	ADEOS-II
16	SensorShortName	Abbreviated name of observing sensor	AMSR
17	ECSDataModel	Name of meta data model	B.0
18	PGENAME	Name of software	(max 20 character)
19	PGEVersion	Version of software	(max 18 character)
20	ProcessingCenter	Input file name	HAToyAMA
21	ContactOrganizationName	Organization name to contact about this product	

5 サンプルプログラム一覧

本章では、L1B、L2 及び L3 プロダクトの各種データの値を画面に表示するためのサンプルプログラムを示します。ADIT インストールディレクトリの下に「sample」ディレクトリ以下に、図 5-1 に示す構成でサンプルプログラム及びデータが格納されています。これらのプログラムのコンパイル方法は、各ソースコードの先頭に記述されています(図 5-2)。また、詳細については、本マニュアルの 3.3.2 節及び 3.3.3 節を参照ください。

サンプルプログラムを上記に従いコンパイルすると、実行モジュールができます。ここで作成される実行モジュール名は、プログラムファイル名から拡張子を除いた文字列となります。ここで作成した実行モジュールの起動方法を以下に示します。

【C プログラムの場合】

% 実行モジュール名△入力データ名

但し、△はブランクを意味します。

例) L1_swath1b.c をコンパイルして作成した実行モジュールを起動する場合を以下に示します。実行モジュール名は「L1_swath1b」、入力データ名は「P1AME030609207MA_P01B0000000.00.sample」とすれば、

```
% L1_swath1b P1AME030609207MA_P01B0000000.00.sample
```

【Fortran プログラムの場合】

% 実行モジュール名

* Fortran プログラムの場合、プログラム実行時に入力データ名をパラメータとして指定できないので、プログラム中に入力データ名を記述する必要があります(図 5-3)。

例) L1_swath1b.f をコンパイルして作成した実行モジュールを起動する場合を以下に示します。

```
% L1_swath1b
```

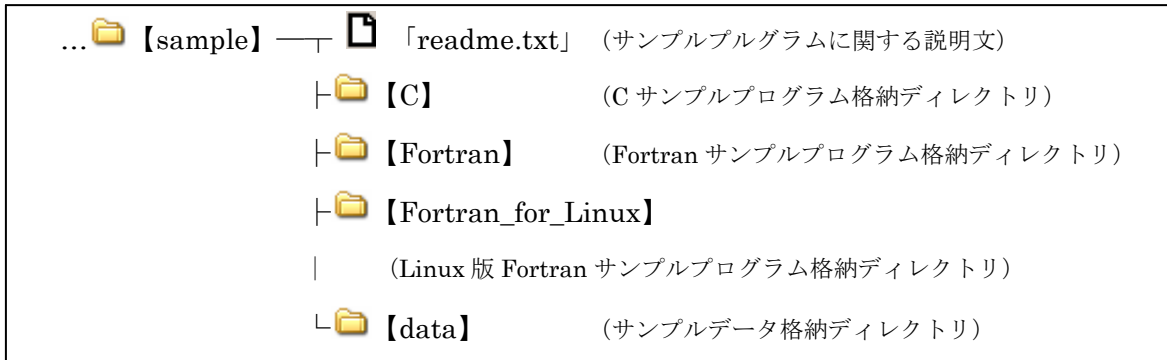


図 5-1 ディレクトリ構成

```

/*****
This is a sample program to read AMSR/L1B data, and
following are instructions for compiling a sample program with ADIT.

For SGI
cc -DSGI -xansi -O -s -o L1_cal L1_cal.c ¥
-I$HDFINC -I../include ¥
-L$HDFLIB -L../lib ¥
-IADIT -lmfhdf -ldf [-ljpeg] -lz -lm

For SunOS
cc -DSUN -Xc -xO2 -lnsl -o L1_cal L1_cal.c ¥
-I$HDFINC -I../include ¥
-L$HDFLIB -L../lib ¥
-IADIT -lmfhdf -ldf [-ljpeg] -lz -lm

For HP-UX
cc -DHP9000 -Ae -s -o L1_cal L1_cal.c ¥
-I$HDFINC -I../include ¥
-L$HDFLIB -L../lib ¥
-IADIT -lmfhdf -l df [-ljpeg] -lz -lm

FOR DEC ALPHA
cc -DDEC_ALPHA -Olimit 2048 -std1 -o L1_cal L1_cal.c ¥
-I$HDFINC -I../include ¥
-L$HDFLIB -L../lib ¥
-IADIT -lmfhdf -ldf [-ljpeg] -lz -lm

FOR LINUX
gcc -DLINUX -ansi -o L1_cal L1_cal.c ¥
-I$HDFINC -I../include ¥
-L$HDFLIB -L../lib ¥
-IADIT -lmfhdf -ldf [-ljpeg] -lz -lm

Note:
$HDFINC indicates the directory of included files of HDF library.
$HDFLIB indicates the directory of library files of HDF library.
*****/

```

各サンプルプログラムの先頭には、計算機ごとのコンパイル方法が記述されています。

図 5-2 コンパイル方法の記述

```

program main

include 'AMSR_f.h'

character*46 fname
data fname/ ../data/P1AME030609207MA_P01B0000000.00.sample '/
integer status
integer i
integer sd_id

record /CAL/      cal
                  .
                  .
                  .

```

入力データを指定する記述

図 5-3 Fortran プログラムでの入力データの指定方法

表 5-1 サンプルプログラム一覧

番号	プログラムファイル名	内容
1.	(1 走査読み込み) L1_swath1b.c (C) L1_swath1b.f (Fortran)	L1B プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。 <ul style="list-style-type: none"> • Scan_Time • Brightness_Temperature(6GHz-89GHz) • Lat_of_Observation_Point_Except_89B • Long_of_Observation_Point_Except_89B • Lat_of_Observation_Point_for_89B • Long_of_Observation_Point_for_89B
	(複数走査読み込み) L1_swath1b_line.c (C) L1_swath1b_line.f (Fortran)	
2.	(1 走査読み込み) L1_sunearth.c (C) L1_sunearth.f (Fortran)	L1B プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。 <ul style="list-style-type: none"> • Sun_Azimuth • Sun_Elevation • Earth_Incidence • Earth_Azimuth • Land/Ocean_Flag_for_6_10_18_23_36_50_89A
	(複数走査読み込み) L1_sunearth_line.c (C) L1_sunearth_line.f (Fortran)	
3.	(1 走査読み込み) L1_status1b.c (C) L1_status1b.f (Fortran)	L1B プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。 <ul style="list-style-type: none"> • Position_in_Orbit • Data_Quality
	(複数走査読み込み) L1_status1b_line.c (C) L1_status1b_line.f (Fortran)	
4.	(1 走査読み込み) L1_cal.c (C) L1_cal.f (Fortran)	L1B プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。 <ul style="list-style-type: none"> • Hot_Load_Count_6_to_52 • Hot_Load_Count_89

表 5-1 サンプルプログラム一覧

番号	プログラムファイル名	内容
	(複数走査読み込み) L1_cal_line.c (C) L1_cal_line.f (Fortran)	<ul style="list-style-type: none"> • Cold_Sky_Mirror_Count_6_to_52 • Cold_Sky_Mirror_Count_89 • Antenna_Temp_Coeff(Of+SI) • Rx_Offset/Gain_Count • SPC_Temperature_Count • SPS_Temperature_Count
5.	(1 走査読み込み) L1_navi.c (C) L1_navi.f (Fortran) (複数走査読み込み) L1_navi_line.c (C) L1_navi_line.f (Fortran)	<p>L1B プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。</p> <ul style="list-style-type: none"> • Navigation_Data • Attitude_Data
6.	(1 走査読み込み) L1_scantime.c (C) L1_scantime.f (Fortran) (複数走査読み込み) L1_scantime_line.c (C) L1_scantime_line.f (Fortran)	<p>L1B プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。</p> <ul style="list-style-type: none"> • Scan_Time
7.	L1_89GHz_low.c(C) L1_89GHz_low.f(Fortran)	<p>L1B プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。</p> <ul style="list-style-type: none"> • 89GHz lof frequency data
8.	(1 走査読み込み) L2_swath2.c (C) L2_swath2.f (Fortran) (複数走査読み込み) L2_swath2_line.c (C) L2_swath2_line.f (Fortran)	<p>L2 プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。</p> <ul style="list-style-type: none"> • Scan Time Table • Geophysical Quantity Data • Lat. of observation point except 89B • Long. of observation point except 89B
9.	(1 走査読み込み) L2_status2.c (C) L2_status2.f (Fortran) (複数走査読み込み) L2_status2_line.c (C) L2_status2_line.f (Fortran)	<p>L2 プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。</p> <ul style="list-style-type: none"> • Position_in_Orbit • Data Quality
10.	(1 走査読み込み) L2_scantime.c (C) L2_scantime.f (Fortran) (複数走査読み込み) L2_scantime_line.c (C) L2_scantime_line.f (Fortran)	<p>L2 プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。</p> <ul style="list-style-type: none"> • Scan Time Table
11.	L3.c (C) L3.f (Fortran)	<p>L3 プロダクトを入力して、以下に示すデータの値を画面に出力するプログラム。</p>

表 5-1 サンプルプログラム一覧

番号	プログラムファイル名	内容
		<ul style="list-style-type: none"> Mean for Brightness Temperature(6GHz-89GHz) Mean for Geophysical Data
12.	sample1.c (C) sample1.f (Fortran)	L1B プロダクトを入力して、メタデータや各種データセットへアクセスするプログラム。なお、本プログラムの詳細な説明は 3.2.3 節及び 3.3.3 節に示します。
13.	sample2.c (C) sample2.f (Fortran)	L2 プロダクトを入力して、メタデータや各種データセットへアクセスするプログラム。
14.	sample3.c (C) sample3.f (Fortran)	L3 プロダクトを入力して、メタデータや各種データセットへアクセスするプログラム。

表 5-2 サンプルデータ一覧

番号	データファイル名	内容
1.	P1AME030201006MA_P01B0000000.00.sample	AMSR-E L1B プロダクト
2.	P1AME030609207A_P2WV0Tak071.00.sample	AMSR-E L2 プロダクト
3.	P1AME030609A_P3WV0Tak071E0.00.sample	AMSR-E L3 プロダクト