

月周回衛星かぐや(SELENE)
プロダクトフォーマット記述書
～プラズマ観測装置(PACE)編～

Version 1.0

平成 21 年 11 月 1 日

目次

1. 概要	1
1.1 目的	1
1.2 本フォーマット記述書の構成	1
1.3 データセット.....	2
1.3.1 プロダクト	2
1.3.2 カタログ情報ファイル	3
1.3.3 サムネイル画像ファイル	3
1.4 対象プロダクト	4
2. 電子反射計月面磁気異常マップ (プロダクト ID:PACR_ERMA_MAP).....	5
2.1 ファイル命名規約	5
2.2 ラベルフォーマット.....	5
2.3 データオブジェクトフォーマット.....	7
2.4 カタログ情報ファイルフォーマット	7
3. 月面 2 次イオンマップ(プロダクト ID:PACE_SI_MAP).....	8
3.1 ファイル命名規約	8
3.2 ラベルフォーマット.....	8
3.3 データオブジェクトフォーマット.....	10
3.4 カタログ情報ファイルフォーマット	10
4. 電子・イオンエネルギースペクトル時系列データ(処理後)(プロダクト ID:PACE_PBF_1) ..	11
4.1 ファイル命名規約	11
4.2 ラベルフォーマット.....	11
4.3 データオブジェクトフォーマット.....	12
4.4 カタログ情報ファイルフォーマット	13
5. 電子・イオンエネルギースペクトル時系列データ(CDF) (プロダクト ID:PACE_CDF).....	14
5.1 ファイル命名規約	14
5.2 ラベルフォーマット.....	14
5.3 データオブジェクトフォーマット.....	14
5.4 カタログ情報ファイルフォーマット	14
6. 電子・イオンエネルギースペクトル時系列サマリプロット(プロダクト ID:PACE_ET_summary)	15
6.1 ファイル命名規約	15

6.2	ラベルフォーマット.....	15
6.3	データオブジェクトフォーマット.....	16
6.4	カタログ情報ファイルフォーマット	17

添付資料-1 Saito et al. 2008("Low-energy charged particle measurement by
MAP-PACE onboard SELENE")

添付資料-2 paceql_outputdata_070904.h

添付資料-3 dump_pbf.c

1. 概要

1.1 目的

本文書は、宇宙航空研究開発機構（以下、「JAXA」という。）が、月周回衛星かぐや（以下、「SELENE」という。）搭載のプラズマ観測装置^{※1}（以下、「PACE」という）のPRODUCTファイル^{※2} およびカタログファイルのフォーマットを記述するものである。

※1：PACEのミッションについては、下記の「かぐやプロジェクトホームページ」および「かぐや画像ギャラリー」を参照。

- ✓ かぐやプロジェクトホームページ
http://www.kaguya.jaxa.jp/ja/equipment/pace_j.htm
- ✓ かぐや画像ギャラリー
http://wms.selene.darts.isas.jaxa.jp/selene_viewer/jpn/observation_mission/pace/

※2：SELENEのデータフォーマットは、NASAのPDS(Planetary Data System)を元に定められている。ただし、完全準拠はしていない。

1.2 本フォーマット記述書の構成

本フォーマット記述書の構成を表 1-1 に示す。

表 1-1 本フォーマット記述書の構成

No.	参照先	項目	記述内容
1	1.3 節	表 1-2 PACE プロダクト一覧	本記述書で記述しているプロダクト一覧として、プロダクトの名称、オブジェクト形式、プロダクトの構成について記載している。
		表 1-3 各プロダクト説明	No.1 のプロダクト一覧で示した各プロダクトについて、データに含まれる内容、観測方法等に関する解説を記述している。
2	X 章	“プロダクト”	No.1 のプロダクト一覧で示したプロダクトについて、ファイル命名規約、ラベルフォーマット、データオブジェクトフォーマット、カタログ情報ファイルフォーマットを記述している。
3	X.1 節	ファイル命名規約	No.2 で示したプロダクトについて、ファイル命名規約を記述している。
4	X.2 節	ラベルフォーマット	No.2 で示したプロダクトについて、オブジェクトのラベル部のフォーマットを記述している。
5	X.3 節	データオブジェクトフォーマット	No.2 で示したプロダクトについて、データオブジェクトのデータフォーマットを記述している。 (データファイルの拡張子は、プロダクト毎にユニークであるため、2.1 章のファイル命名規約を参照のこと)
6	X.4 節	カタログ情報ファイルフォーマット	No.2 で示したプロダクトについて、プロダクトのカタログ情報ファイル(拡張子.ctg)のフォーマットを記述。
7	X+1 章		
		以降、同様	

1.3 データセット

ある一つのデータセットは、プロダクト、カタログ情報ファイルおよびサムネイル画像ファイル (jpeg 形式) がセットで tar アーカイブされており、これを L2 データセットと呼ぶ。拡張子は「SL2」としている。ただし、サムネイル画像ファイルはプロダクト作成者の判断により省略される場合がある。

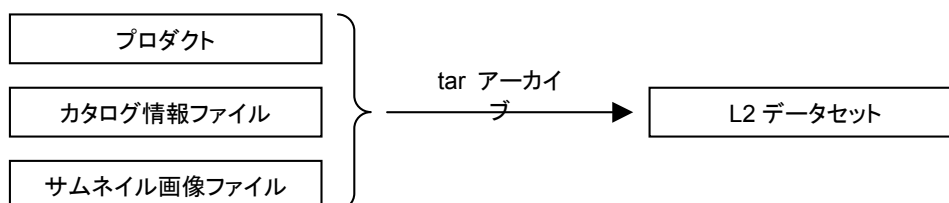


図 1-1 L2 データセットの構成

1.3.1 プロダクト

プロダクトはラベル情報とデータオブジェクトが同一ファイルとして構成されている「アタッチド形式」と、ラベルとデータオブジェクトが別ファイルとして構成されている「デタッチド形式」がある。

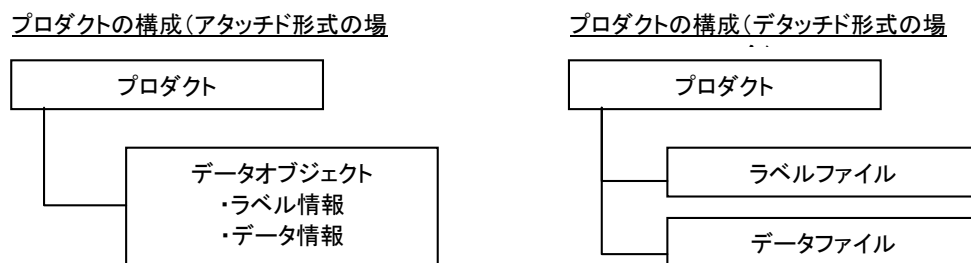


図 1-2 アタッチド形式とデタッチド形式

- (1) ラベルファイル(データオブジェクト(ラベル情報))
ラベルファイル(ラベル情報)は、データファイル(データ情報)を識別するための情報をテキスト形式で格納する。
- (2) データファイル(データオブジェクト(データ情報))
プロダクトのデータファイル(データオブジェクト(データ情報))は、データの形態に応じ、主に以下に示す種類の形式に分類される。
 - a) IMAGE : 画像データ
2次元配列の画像データである。brightness level や display color を割り当てることによって、サンプルの視覚表示を作成されている。ひとつの IMAGE は、同じ数のサンプルを含んだ一連の Line から構成されている。
※PDS standard reference V3.8 Appendix A.20 IMAGE を参照

- b) TABLE : 表形式データ
表形式にデータを格納したファイルである。バイナリー、または ASCII で記述されている。
※PDS standard reference V3.8 Appendix A.29 TABLE を参照
- c) SERIES : 時系列データ
TABLE と同様に表形式にデータを格納したファイルである。TABLE に要素間の変化のパラメータ情報を追加した TABLE と同じ物理的な書式仕様を使用する。
※PDS standard reference V3.8 Appendix A.24 SERIES を参照
- d) TEXT : 文字列データ
プロダクトの解説を記述してあるシンプルなテキストファイルである。
※PDS standard reference V3.8 Appendix A.30 TEXT を参照

1.3.2 カタログ情報ファイル

カタログ情報ファイルは、プロダクトの概要を説明するために添付される情報ファイルであり、L2DB サブシステムからプロダクトの検索を行う際に使用される。

1.3.3 サムネイル画像ファイル

サムネイル画像ファイルは、データオブジェクトの縮小画像であり、JPEG 形式等の画像である。なお、プロダクト作成者の判断により省略される場合がある。

1.4 対象プロダクト

本文書が対象とする PACE プロダクトの一覧を表 1-2 に示す。また、各プロダクトの説明を表 1-3 に示す。

表 1-2 PACE プロダクト一覧

処理レベル※1	プロダクト和名	Product ID	Object 形式	プロダクト構成※2
高次 (Higher Level)	電子反射計月面磁気異常マップ	PACE_ERMA_MAP	IMAGE	A
高次 (Higher Level)	月面2 次イオンマップ	PACE_SI_MAP	IMAGE	A
標準 (Standard)	電子・イオンエネルギースペクトル時系列データ(処理後)	PACE_PBF_1	SERIES	D
標準 (Standard)	電子・イオンエネルギースペクトル時系列データ(CDF)	PACE_CDF	~※3	~※3
標準 (Standard)	電子・イオンエネルギースペクトル時系列サマリプロット	PACE_ET_summary	IMAGE	A

■ : マッププロダクト

※1 : 機器で得たデータはそのままでは人の目で見て分かる物では無いため、地上のシステムで各種加工や補正処理を施す必要がある。加工・補正処理の工程の差により標準処理と高次処理に分けることができる。高次処理は、標準処理のデータを研究目的等に応じて様々な加工・補正処理が施されたデータ。

※2 : A → アタッチド形式 D→デタッチド形式

※3 : CDF 形式。Ver3.1 に完全準拠。

表 1-3 各プロダクト説明

プロダクト和名	Product ID
電子反射計月面磁気異常マップ	PACE ESA 及び LMAG を電子反射計として使って測定した月表面磁気異常の強度分布
月面2 次イオンマップ	PACE で計測される月面から反射されたイオンの2次元マップ
電子・イオンエネルギースペクトル時系列データ(処理後)	PACE で計測される電子・イオンのエネルギースペクトルの高時間分解能データ
電子・イオンエネルギースペクトル時系列データ(CDF)	PACE で計測される電子・イオンのエネルギースペクトルの高時間分解能データ
電子・イオンエネルギースペクトル時系列サマリプロット	PACE で計測される電子・イオンのエネルギースペクトルのサマリプロット

■ : マッププロダクト

(1) CDF 形式

CDF※1(Common Data Format)で作成され、これにカタログ情報が付与される。よって、L2 プロダクトとしては、CDF 形式のデータプロダクトファイルとカタログ情報ファイルが含まれ、レベルは定義されない。CDF Ver3.1 に完全準拠している。

※1 : <http://cdf.gsfc.nasa.gov/>

2. 電子反射計月面磁気異常マップ (プロダクト ID:PACR_ERMA_MAP)

2.1 ファイル命名規約

PACE プロダクトファイルを構成するラベル、データオブジェクトファイルおよびカタログ情報ファイルの命名規約を以下に示す。なお、ファイル名は大文字、小文字の区別はしない。

PACE_AAAA_MAP_V###.dat

- AAAA : プロダクト名
 - ✓ ERMA : 電子反射計月面磁気異常マップ
- ### : バージョンナンバー(3 桁)
- dat : ファイル識別
 - ✓ img : データファイル
 - ✓ ctg : カタログ情報ファイル
 - ✓ sl2 : L2 データセット (tar アーカイブ)

2.2 ラベルフォーマット

電子反射計月面磁気異常マッププロダクトの IMAGE オブジェクトラベルフォーマットを表 2-1 に示す。電子反射計月面磁気異常マッププロダクトの IMAGE オブジェクトのラベルには、基本項目、画像データオブジェクトフォーマット記述部および地図投影オブジェクト記述部が含まれる。

表 2-1 において、【固定】と示された以外の項目については、プロダクトの種別等に応じた数値、文字列が設定される。

表 2-1 電子反射計月面磁気異常マップ IMAGE オブジェクトラベル

No	項目名	要素	型	値
基本項目				
1	PDS バージョン宣言	PDS_VERSION_ID = %s	char(PDS3 【固定】
2	ファイルレコード形式	RECORD_TYPE = %s	char	UNDEFINED 【固定】
3	ファイル名	FILE_NAME = %s	char	2.1 節 ファイル命名規約 参照
4	画像オブジェクト先頭位置	^IMAGE = %d <BYTES>	int	XXXX <BYTES>
5	ファイルレコードバイト数	RECORD_BYTES = %d	int	XXXXXX
6	ミッション名	MISSION_NAME = %s	char	SELENE 【固定】
7	探査機名称	SPACECRAFT_NAME = %s	char	SELENE-M【固定】
8	センサ名(フルネーム)	INSTRUMENT_NAME = %s	char	PACE 【固定】
9	プロダクト名	PRODUCT_SET_ID = %s	char	表 1-2 Product ID 参照
10	観測対象名	TARGET_NAME = %s	char	MOON 【固定】
11	コメント	COMMENT_TEXT = "%s"	char	
画像データオブジェクトフォーマット記述部 (* IMAGE *)				
		OBJECT = IMAGE		
12	バンド格納種別	BAND_STORAGE_TYPE = %s	char	SAMPLE_INTERLEAVED 【固定】 ※PDS standard reference V3.5 Appendix A.19 IMAGE を参照
13	バンド数	BANDS = %d	smallint	X
14	圧縮種類と暗号化名	ENCODING_TYPE = %s	char	N/A 【固定】
15	画像横方向画素数	LINE_SAMPLES = %d	int	XXX
16	画像縦方向画素数	LINES = %d	int	XXX
17	画素ビット長	SAMPLE_BITS = %d	int	X
18	画素タイプ	SAMPLE_TYPE = %s	char	MSB_INTEGE R【固定】 ※MSB_INTEGER の詳細につ

				いては、PDS standard reference V3.5 Appendix C.1を参照
19	ストレッチング済みフラグ	STRETCHED_FLAG = %s	char	FALSE【固定】
20	想定外値の代替値	INVALID_CONSTANT = %s	char	X
21	オフセット	OFFSET = %f	float	X.X
22	スケーリングファクター	SCALING_FACTOR = %f	float	X.X
		END_OBJECT = IMAGE		
地図投影オブジェクト記述部 (* IMAGE_MAP_PROJECTION *)				
		OBJECT IMAGE_MAP_PROJECTION =		
23	解像度	MAP_RESOLUTION = %f<PIXEL/DEGREE>	float	X < PIXEL/DEGREE>
24	楕円体の半長径	A_AXIS_RADIUS = %f<KM>	float	1737.400 <km> 【固定】
25	楕円体の中間軸	B_AXIS_RADIUS = %f<KM>	float	1737.400 <km> 【固定】
26	楕円体の半短径	C_AXIS_RADIUS = %f<KM>	float	1737.400 <km> 【固定】
27	座標系名称	COORDINATE_SYSTEM_NAME = "%s"	char	"PLANETOCENTRIC "【固定】
28	座標系タイプ	COORDINATE_SYSTEM_TYPE = "%s"	char	"BODY-FIXED ROTATING" 【固定】
29	最北緯度	MAXIMUM_LATITUDE = %f	float	89.0 【固定】
30	最南緯度	MINIMUM_LATITUDE = %f	float	-89.0 【固定】
31	最西経度	WESTERNMOST_LONGITUDE = %f	float	0.0 【固定】
32	最東経度	EASTERNMOST_LONGITUDE = %f	float	359.0 【固定】
33	地図投影法	MAP_PROJECTION_TYPE = "%s"	char	SIMPLE_CYLINDRICAL 【固定】
34	経度の正の方向	POSITIVE_LONGITUDE_DIRECTION = "%s"	char	EAST 【固定】
		END_OBJECT IMAGE_MAP_PROJECTION =		
終了記述				
		END		

【ラベルサンプル: 電子反射計月面磁気異常マッププロダクト】

```

PDS_VERSION_ID          = PDS3
RECORD_TYPE              = UNDEFINED
FILE_NAME                = PACE_SI_MAP_001.img
^IMAGE                   = 1082 <BYTES>
RECORD_BYTES             = 579960
SPACECRAFT_NAME         = SELENE-M
INSTRUMENT_NAME         = PACE
PRODUCT_SET_ID          = PACE_SI_MAP
TARGET_NAME              = MOON
COMMENT_TEXT             = "Magnetic field anomaly map. 9 bands."

```

```

/* IMAGE */
OBJECT                   = IMAGE
  BAND_STORAGE_TYPE     = SAMPLE_INTERLEAVED
  BANDS                  = 9
  ENCODING_TYPE          = N/A
  LINE_SAMPLES           = 360
  LINES                  = 179
  SAMPLE_BITS           = 8
  SAMPLE_TYPE            = MSB_INTEGER
  STRETCHED_FLAG        = FALSE
  INVALID_CONSTANT      = 0
  OFFSET                 = 0.0
  SCALING_FACTOR        = 0.5
END_OBJECT               = IMAGE

/* IMAGE_MAP_PROJECTION */
OBJECT                   = IMAGE_MAP_PROJECTION
  MAP_RESOLUTION         = 1 < PIXEL / DEGREE>
  A_AXIS_RADIUS          = 1737.400 <km>
  B_AXIS_RADIUS          = 1737.400 <km>
  C_AXIS_RADIUS          = 1737.400 <km>
  COORDINATE_SYSTEM_NAME = "PLANETOCENTRIC"

```

```

COORDINATE_SYSTEM_TYPE = "BODY-FIXED ROTATING"
MAXIMUM_LATITUDE       = 89.0
MINIMUM_LATITUDE       = -89.0
WESTERNMOST_LONGITUDE  = 0.0
EASTERNMOST_LONGITUDE  = 359.0
MAP_PROJECTION_TYPE    = "SIMPLE_CYLINDRICAL"
POSITIVE_LONGITUDE_DIRECTION = EAST
END_OBJECT              = IMAGE_MAP_PROJECTION
END

```

2.3 データオブジェクトフォーマット

T.B.D

2.4 カタログ情報ファイルフォーマット

PACE プロダクトの IMAGE オブジェクトカタログ情報ファイルフォーマットを表 2-2 に示す。

表 2-2 IMAGE オブジェクトカタログ情報ファイルフォーマット

項目名	要素	設定値のフォーマット	設定値の範囲	設定値
データファイル名(*1)	DataFileName	AAAA...AAAA (最大31 桁)	任意の英数字	プロダクトによる (2.1 節 ファイル命名規約 参照)
データファイルサイズ	DataFileSize	NNNNNNNNNNNN (最大12 桁)	単位:バイト	プロダクトによる
データファイルフォーマット	DataFileFormat	AAAA...AAAA (最大16 桁)	任意の文字列	PDS【固定】
機器名	InstrumentName	AAAA...AAAA (最大16 桁)	任意の文字列	PACE【固定】
処理レベル	ProcessingLevel	AAAA...AAAA (最大16 桁)	任意の文字列	プロダクトによる(表 1-2 処理レベル 参照)
プロダクト種別	ProductID	AAAA...AAAA (最大30 桁)	任意の文字列	プロダクトによる(表 1-2 Product ID 参照)
プロダクトバージョン	ProductVersion	AAAA...AAAA (最大16 桁)	任意の文字列	プロダクトによる
アクセスレベル	AccessLevel	N	0-4 の数値	N/A

(*1) データファイル名にはプロダクトのファイル名を格納する。デタッチド形式の場合は、データファイルの名前を格納する。

【カタログ情報ファイルサンプル: 電子反射計月面磁気異常マッププロダクト】

```

DataFileName = PACE_SI_MAP_001.img
DataFileSize = 581041
DataFileFormat = PDS
InstrumentName = PACE
ProcessingLevel = Higher Level
ProductID = PACE_SI_MAP
ProductVersion = 1.0
AccessLevel = 4

```

3. 月面 2 次イオンマップ(プロダクト ID:PACE_SI_MAP)

3.1 ファイル命名規約

PACE プロダクトファイルを構成するラベル、データオブジェクトファイルおよびカタログ情報ファイルの命名規約を以下に示す。なお、ファイル名は大文字、小文字の区別はしない。

PACE_AAAA_MAP_V###.dat

- AAAA : プロダクト名
 - ✓ SI : 月面 2 次イオンマップ
- ### : バージョンナンバー(3 桁)
- dat : ファイル識別
 - ✓ img : データファイル
 - ✓ ctg : カタログ情報ファイル
 - ✓ sl2 : L2 データセット (tar アーカイブ)

3.2 ラベルフォーマット

月面 2 次イオンマッププロダクトの IMAGE オブジェクトラベルフォーマットを表 3-1 に示す。月面 2 次イオンマッププロダクトの IMAGE オブジェクトのラベルには、基本項目、画像データオブジェクトフォーマット記述部および地図投影オブジェクト記述部が含まれる。

表 3-1 において、【固定】と示された以外の項目については、プロダクトの種別等に応じた数値、文字列が設定される。

表 3-1 月面 2 次イオンマップ IMAGE オブジェクトラベル

No	項目名	要素	型	値
基本項目				
1	PDS バージョン宣言	PDS_VERSION_ID = %s	char	PDS3【固定】
2	ファイルレコード形式	RECORD_TYPE = %s	char	UNDEFINED【固定】
3	ファイル名	FILE_NAME = %s	char	3.1 節 ファイル命名規約 参照
4	画像オブジェクト先頭位置	^IMAGE = %d <BYTES>	int	XXXX <BYTES>
5	ファイルレコードバイト数	RECORD_BYTES = %d	int	XXXXXX
6	ミッション名	MISSION_NAME = %s	char	SELENE【固定】
7	探査機名称	SPACECRAFT_NAME = %s	char	SELENE-M【固定】
8	センサ名(フルネーム)	INSTRUMENT_NAME = %s	char	PACE【固定】
9	プロダクト名	PRODUCT_SET_ID = %s	char	表 1-2 Product ID 参照
10	観測対象名	TARGET_NAME = %s	char	MOON【固定】
11	コメント	COMMENT_TEXT = "%s"	char	
画像データオブジェクトフォーマット記述部 (* IMAGE *)				
		OBJECT = IMAGE		
12	バンド格納種別	BAND_STORAGE_TYPE = %s	char	SAMPLE_INTERLEAVED【固定】 ※PDS standard reference V3.5 Appendix A.19 IMAGE を参照
13	バンド数	BANDS = %d	smallint	X
14	圧縮種類と暗号化名	ENCODING_TYPE = %s	char	N/A【固定】
15	画像横方向画素数	LINE_SAMPLES = %d	int	XXX
16	画像縦方向画素数	LINES = %d	int	XXX
17	画素ビット長	SAMPLE_BITS = %d	int	X
18	画素タイプ	SAMPLE_TYPE = %s	char	MSB_INTEGE R【固定】

				※MSB_INTEGER の詳細については、PDS standard reference V3.5 Appendix C.1 を参照
19	ストレッチング済みフラグ	STRETCHED_FLAG = %s	char	FALSE【固定】
20	想定外値の代替値	INVALID_CONSTANT = %s	char	X
21	オフセット	OFFSET = %f	float	X.X
22	スケーリングファクター	SCALING_FACTOR = %f	float	X.X
		END_OBJECT = IMAGE		
地図投影オブジェクト記述部 (* IMAGE_MAP_PROJECTION *)				
		OBJECT IMAGE_MAP_PROJECTION =		
23	解像度	MAP_RESOLUTION = %f<PIXEL/DEGREE>	float	X < PIXEL/DEGREE>
24	楕円体の半長径	A_AXIS_RADIUS = %f<KM>	float	1737.400 <km> 【固定】
25	楕円体の中間軸	B_AXIS_RADIUS = %f<KM>	float	1737.400 <km> 【固定】
26	楕円体の半短径	C_AXIS_RADIUS = %f<KM>	float	1737.400 <km> 【固定】
27	座標系名称	COORDINATE_SYSTEM_NAME = "%s"	char	"PLANETOCENTRIC" 【固定】
28	座標系タイプ	COORDINATE_SYSTEM_TYPE = "%s"	char	"BODY-FIXED ROTATING" 【固定】
29	最北緯度	MAXIMUM_LATITUDE = %f	float	89.0 【固定】
30	最南緯度	MINIMUM_LATITUDE = %f	float	-89.0 【固定】
31	最西経度	WESTERNMOST_LONGITUDE = %f	float	0.0 【固定】
32	最東経度	EASTERNMOST_LONGITUDE = %f	float	359.0 【固定】
33	地図投影法	MAP_PROJECTION_TYPE = "%s"	char	SIMPLE_CYLINDRICAL 【固定】
34	経度の正の方向	POSITIVE_LONGITUDE_DIRECTION = "%s"	char	EAST 【固定】
		END_OBJECT IMAGE_MAP_PROJECTION =		
終了記述				
		END		

【ラベルサンプル: 月面 2 次イオンマッププロダクト】

```

PDS_VERSION_ID          = PDS3
RECORD_TYPE             = UNDEFINED
FILE_NAME               = PACE_SI_MAP_001.img
^IMAGE                  = 1082 <BYTES>
RECORD_BYTES            = 579960
SPACECRAFT_NAME        = SELENE-M
INSTRUMENT_NAME        = PACE
PRODUCT_SET_ID         = PACE_SI_MAP
TARGET_NAME             = MOON
COMMENT_TEXT            = "Magnetic field anomaly map.  9 bands."

```

```

/* IMAGE */
OBJECT                  = IMAGE
  BAND_STORAGE_TYPE    = SAMPLE_INTERLEAVED
  BANDS                 = 9
  ENCODING_TYPE        = N/A
  LINE_SAMPLES         = 360
  LINES                = 179
  SAMPLE_BITS          = 8
  SAMPLE_TYPE          = MSB_INTEGER
  STRETCHED_FLAG       = FALSE
  INVALID_CONSTANT     = 0
  OFFSET               = 0.0
  SCALING_FACTOR       = 0.5
END_OBJECT              = IMAGE

```

```

/* IMAGE_MAP_PROJECTION */
OBJECT                  = IMAGE_MAP_PROJECTION
  MAP_RESOLUTION       = 1 < PIXEL / DEGREE>
  A_AXIS_RADIUS        = 1737.400 <km>
  B_AXIS_RADIUS        = 1737.400 <km>
  C_AXIS_RADIUS        = 1737.400 <km>

```

```

COORDINATE_SYSTEM_NAME      = "PLANETOCENTRIC"
COORDINATE_SYSTEM_TYPE      = "BODY-FIXED ROTATING"
MAXIMUM_LATITUDE             = 89.0
MINIMUM_LATITUDE             = -89.0
WESTERNMOST_LONGITUDE        = 0.0
EASTERNMOST_LONGITUDE        = 359.0
MAP_PROJECTION_TYPE          = "SIMPLE_CYLINDRICAL"
POSITIVE_LONGITUDE_DIRECTION = EAST
END_OBJECT                   = IMAGE_MAP_PROJECTION
END

```

3.3 データオブジェクトフォーマット

T.B.D

3.4 カタログ情報ファイルフォーマット

PACE プロダクトの IMAGE オブジェクトカタログ情報ファイルフォーマットを表 3-2 に示す。

表 3-2 IMAGE オブジェクトカタログ情報ファイルフォーマット

項目名	要素	設定値のフォーマット	設定値の範囲	設定値
データファイル名(*1)	DataFileName	AAAA...AAAA (最大31 桁)	任意の英数字	プロダクトによる (3.1 節 ファイル命名規約 参照)
データファイルサイズ	DataFileSize	NNNNNNNNNNNN (最大12 桁)	単位: バイト	プロダクトによる
データファイルフォーマット	DataFileFormat	AAAA...AAAA (最大16 桁)	任意の文字列	PDS【固定】
機器名	InstrumentName	AAAA...AAAA (最大16 桁)	任意の文字列	PACE【固定】
処理レベル	ProcessingLevel	AAAA...AAAA (最大16 桁)	任意の文字列	プロダクトによる(表 1-2 処理レベル 参照)
プロダクト種別	ProductID	AAAA...AAAA (最大30 桁)	任意の文字列	プロダクトによる(表 1-2 Product ID 参照)
プロダクトバージョン	ProductVersion	AAAA...AAAA (最大16 桁)	任意の文字列	プロダクトによる
アクセスレベル	AccessLevel	N	0-4 の数値	N/A

(*1) データファイル名にはプロダクトのファイル名を格納する。デタッチド形式の場合は、データファイルの名前を格納する。

【カタログ情報ファイルサンプル: 月面 2 次イオンマッププロダクト】

```

DataFileName = PACE_SI_MAP_001.img
DataFileSize = 581041
DataFileFormat = PDS
InstrumentName = PACE
ProcessingLevel = Higher Level
ProductID = PACE_SI_MAP
ProductVersion = 1.0
AccessLevel = 4

```

4. 電子・イオンエネルギースペクトル時系列データ(処理後)(プロダクト ID: PACE_PBF_1)

4.1 ファイル命名規約

PACE プロダクトファイルを構成するラベル、データオブジェクトファイルおよびカタログ情報ファイルの命名規約を以下に示す。なお、ファイル名は大文字、小文字の区別はしない。

xPACE_PBFn_yyyymmdd_BBB_V###.dat

- x : バイナリ形式
 - ✓ I : インテル形式
 - ✓ S : 非インテル(標準)形式
- n : プロダクト名
 - ✓ 1 : 処理後
- yyyymmdd : 観測年月日
- BBB : 使用センサ(ESA1, ESA2, IMA, IEA)
- ### : バージョンナンバー(3桁)
- dat : ファイル識別
 - ✓ dat.gz : データファイル
 - ✓ lbl : ラベルファイル
 - ✓ ctg : カタログ情報ファイル
 - ✓ sl2 : L2 データセット (tar アーカイブ)

4.2 ラベルフォーマット

PACEプロダクトのSERIES オブジェクトラベルフォーマットを表 4-1に示す。SERIES オブジェクトのラベルには、基本項目、オブジェクトフォーマット記述部が含まれる。

表 4-1において、【固定】と示された以外の項目については、プロダクトの種別等に応じた数値、文字列が設定される。

表 4-1 SERIES オブジェクトラベル

No.	項目名	要素	型	値
基本項目				
1	PDS バージョン宣言	PDS_VERSION_ID = %s	char	PDS3【固定】
2	ファイルレコード形式	RECORD_TYPE = %s	char	UNDEFINED【固定】
3	ファイル名	FILE_NAME = %s	char	4.1 節 ファイル命名規約参照
4	ミッション名	MISSION_NAME = %s	char	SELENE【固定】
5	探査機名称	SPACECRAFT_NAME = %s	char	SELENE-M【固定】
6	センサ名(フルネーム)	INSTRUMENT_NAME = %s	char	PACE【固定】
7	プロダクト名	PRODUCT_SET_ID = %s	char	表 1-2 Product ID 参照
8	プロダクトバージョン	PRODUCT_VERSION_ID = %s	char	Ver.X.X
9	観測対象名	TARGET_NAME = %s	char	MOON【固定】
10	コメント	COMMENT_TEXT = "%s"	char	
11	オブジェクト先頭位置	^TIME_SERIES = %d <BYTES>	int	0 <BYTES>【固定】
オブジェクトフォーマット記述部				
		OBJECT = TIME_SERIES		
12	データ形式	INTERCHANGE_FORMAT = %s	char	ASCII【固定】
13	データ行数	ROWS = %d	int	XXXXX
14	データ列数	COLUMNS = %d	int	XX

15	データ行バイト数	ROW_BYTES = %d	int	XX
16	パラメータ名	SAMPLING_PARAMETER_NAME = %s	char	TIME【固定】
17	パラメータ単位	SAMPLING_PARAMETER_UNIT = %s	char	SECOND【固定】
18	パラメータ間隔	SAMPLING_PARAMETER_INTERVAL = %f	float	X.X
19	データ開始日時	START_TIME = %s	char	yyyy-mm-ddThh:mm:ss
20	データ終了日時	STOP_TIME = %s	char	yyyy-mm-ddThh:mm:ss
		END_OBJECT = TIME_SERIES		
終了記述				
		END		

【ラベルサンプル：電子・イオンエネルギースペクトル時系列データプロダクト】

```

PDS_VERSION_ID          = PDS3
RECORD_TYPE             = UNDEFINED
FILE_NAME                = PACE_ET_20080610.dat
MISSION_NAME            = SELENE
SPACECRAFT_NAME        = SELENE-M
INSTRUMENT_NAME        = PACE
PRODUCT_SET_ID         = PACE_ET
PRODUCT_VERSION_ID     = Ver.1.0
TARGET_NAME             = MOON
COMMENT_TEXT            = "Energy-time diagram"
^TIME_SERIES           = 0 <BYTES>

/* TIME SERIES */
OBJECT                  = TIME_SERIES
INTERCHANGE_FORMAT     = ASCII
ROWS                   = 86400
COLUMNS               = 33
ROW_BYTES              = 64
SAMPLING_PARAMETER_NAME = TIME
SAMPLING_PARAMETER_UNIT = SECOND
SAMPLING_PARAMETER_INTERVAL = 1.0
START_TIME             = 2008-06-10T00:00:00
STOP_TIME              = 2008-06-10T23:59:56
END_OBJECT             = TIME_SERIES
END

```

4.3 データオブジェクトフォーマット

電子・イオンエネルギースペクトル時系列データの各センサ(ESA1, ESA2, IMA, IEA)それぞれの観測モードに対応するデータ内容は、下記の論文に記載している。

Saito et al. 2008("Low-energy charged particle measurement by MAP-PACE onboard SELENE") (添付資料-1)

また具体的なフォーマットはヘッダーファイル"paceql_outputdata_070904.h" (添付資料-2) 中に全てある。データの読み取りは上記ヘッダーファイルを使ったサンプルプログラム"dump_pbf.c" (添付資料-3) にて行うことができる。

4.4 カタログ情報ファイルフォーマット

PACE プロダクトの SERIES オブジェクトのカタログ情報ファイルフォーマットを表 4-2 に示す。

表 4-2 SERIES オブジェクトカタログ情報ファイルフォーマット

項目名	要素	設定値のフォーマット	設定値の範囲	設定値
データファイル名(*1)	DataFileName	AAAA...AAAA (最大31 桁)	任意の英数字	プロダクトによる (4.1 節 ファイル命名規約 参照)
データファイルサイズ	DataFileSize	NNNNNNNNNNNN (最大12 桁)	単位:バイト	プロダクトによる
データファイルフォーマット	DataFileFormat	AAAA...AAAA (最大16 桁)	任意の文字列	PDS【固定】
機器名	InstrumentName	AAAA...AAAA (最大16 桁)	任意の文字列	PACE【固定】
処理レベル	ProcessingLevel	AAAA...AAAA (最大16 桁)	任意の文字列	プロダクトによる (表 1-2 処理レベル 参照)
プロダクト種別	ProductID	AAAA...AAAA (最大30 桁)	任意の文字列	プロダクトによる (表 1-2 Product ID 参照)
プロダクトバージョン	ProductVersion	AAAA...AAAA (最大16 桁)	任意の文字列	プロダクトによる
アクセスレベル	AccessLevel	N	0-4 の数値	N/A
データ開始日時	StartDateTime	yyyy- mmddT hh: mm: ss.sssssZ	日時	プロダクトによる
データ終了日時	EndDateTime	yyyy- mmddT hh: mm: ss.sssssZ	日時	プロダクトによる

(*1) データファイル名にはプロダクトのファイル名を格納する。デタッチド形式の場合は、データファイルの名前を格納する。

【カタログ情報ファイルサンプル: 電子・イオンエネルギー分光時系列データプロダクト】

DataFileName =PACE_ET_20080610.dat
 DataFileSize =10875290
 DataFileFormat =24*60*60
 InstrumentName =PACE
 ProcessingLevel =Standard
 ProductID =PACE_ET
 ProductVersion =1.0
 AccessLevel =3
 StartDateTime = 2008-06-10T00:00:00Z
 EndDateTime = 2008-06-10T23:59:59Z

5. 電子・イオンエネルギースペクトル時系列データ(CDF) (プロダクト ID: PACE_CDF)

5.1 ファイル命名規約

PACE プロダクトファイルを構成するラベル、データオブジェクトファイルおよびカタログ情報ファイルの命名規約を以下に示す。なお、ファイル名は大文字、小文字の区別はしない。

PACE_CDFn_yyyymmdd_BBB_V###.dat

- n : プロダクト名
 - ✓ 0 : 処理前
 - ✓ 1 : 処理後
- yyyymmdd : 観測年月日
- BBB : 使用センサ
 - ✓ ESA :
 - ✓ IMA :
 - ✓ IEA :
- ### : バージョンナンバー(3 桁)
- dat : ファイル識別
 - ✓ cdf : データファイル
 - ✓ ctg : カタログ情報ファイル
 - ✓ sl2 : L2 データセット (tar アーカイブ)

5.2 ラベルフォーマット

データプロダクトファイルは CDF (Common Data Format)形式であるため、ラベルは定義されない。

5.3 データオブジェクトフォーマット

データプロダクトファイルは CDF※1(Common Data Format)形式で作成される。CDF Ver3.1 に完全準拠している。

※1 : <http://cdf.gsfc.nasa.gov/>

5.4 カタログ情報ファイルフォーマット

T.B.D

6. 電子・イオンエネルギー分光時系列サマリプロット(プロダクト ID: PACE_ET_summary)

6.1 ファイル命名規約

PACE プロダクトファイルを構成するラベル、データオブジェクトファイルおよびカタログ情報ファイルの命名規約を以下に示す。なお、ファイル名は大文字、小文字の区別はしない。

PACE_ETn_yyyymmdd_V###.dat

- n : プロダクト名
 - ✓ 0 : 処理前
 - ✓ 1 : 処理後
- yyyymmdd : 観測年月日
- ### : バージョンナンバー(3桁)
- dat : ファイル識別
 - ✓ png : データファイル
 - ✓ ctg : カタログ情報ファイル
 - ✓ sl2 : L2 データセット (tar アーカイブ)

6.2 ラベルフォーマット

電子・イオンエネルギー分光時系列サマリプロットの IMAGE オブジェクトラベルフォーマットを表 6-1 に示す。電子・イオンエネルギー分光時系列サマリプロットの IMAGE オブジェクトのラベルには、基本項目、画像データオブジェクトフォーマット記述部が含まれる。

表 6-1 において、【固定】と示された以外の項目については、プロダクトの種別等に応じた数値、文字列が設定される。

表 6-1 電子・イオンエネルギー分光時系列サマリプロット IMAGE オブジェクトラベル

No	項目名	要素	型	値
基本項目				
1	PDS バージョン宣言	PDS_VERSION_ID = %s	char	PDS3【固定】
2	ファイルレコード形式	RECORD_TYPE = %s	char	UNDEFINED【固定】
3	ファイル名	FILE_NAME = %s	char	6.1 節 ファイル命名規約 参照
4	画像オブジェクト先頭位置	^IMAGE = %d <BYTES>	int	XXXX <BYTES>
5	ファイルレコードバイト数	RECORD_BYTES = %d	int	XXXXXX
6	ミッション名	MISSION_NAME = %s	char	SELENE【固定】
7	探査機名称	SPACECRAFT_NAME = %s	char	SELENE-M【固定】
8	センサ名(フルネーム)	INSTRUMENT_NAME = %s	char	PACE【固定】
9	プロダクト名	PRODUCT_SET_ID = %s	char	表 1-2 Product ID 参照
10	プロダクトバージョン	PRODUCT_VERSION_ID = %s	char	Ver.X.X
11	観測対象名	TARGET_NAME = %s	char	MOON【固定】
12	コメント	COMMENT_TEXT = "%s"	char	
画像データオブジェクトフォーマット記述部 (* IMAGE *)				
		OBJECT = IMAGE		
13	バンド格納種別	BAND_STORAGE_TYPE = %s	char	SAMPLE_INTERLEAVED【固定】 ※PDS standard reference V3.5 Appendix A.19 IMAGE を参照
14	バンド数	BANDS = %d	smallint	X

15	圧縮種類と暗号化名	ENCODING_TYPE = %s	char	N/A【固定】
16	画像横方向画素数	LINE_SAMPLES = %d	int	XXX
17	画像縦方向画素数	LINES = %d	int	XXX
18	画素ビット長	SAMPLE_BITS = %d	int	X
19	画素タイプ	SAMPLE_TYPE = %s	char	MSB_INTEGER【固定】 ※MSB_INTEGERの詳細については、PDS standard reference V3.5 Appendix C.1を参照
20	ストレッチング済みフラグ	STRETCHED_FLAG = %s	char	FALSE【固定】
21	想定外値の代替値	INVALID_CONSTANT = %s	char	X
22	オフセット	OFFSET = %f	float	X.X
23	スケーリングファクター	SCALING_FACTOR = %f	float	X.X
24	開始時刻	START_TIME = %s	char	YYYY-MM-DDThh:mm:ss
25	終了時刻	STOP_TIME = %s	char	YYYY-MM-DDThh:mm:ss
		END_OBJECT = IMAGE		
終了記述				
		END		

【ラベルサンプル：電子・イオンエネルギースペクトル時系列サマリプロットプロダクト】

```

PDS_VERSION_ID      = PDS3
RECORD_TYPE         = UNDEFINED
FILE_NAME           = PACE_ET1_20080601_V001.png
^IMAGE              = 0 <BYTES>
RECORD_BYTES        = 64475
MISSION_NAME        = SELENE
SPACECRAFT_NAME     = SELENE-M
INSTRUMENT_NAME     = PACE
PRODUCT_SET_ID      = PACE_ET_summary
PRODUCT_VERSION_ID  = Ver.1.0
TARGET_NAME         = MOON
COMMENT_TEXT        = "Energy-time diagram, Magnetic field and S/C orbit"

```

```

/* IMAGE */
OBJECT = IMAGE
BAND_STORAGE_TYPE = SAMPLE_INTERLEAVED
BANDS      = 1
ENCODING_TYPE = N/A
LINE_SAMPLES = 865
LINES      = 690
SAMPLE_BITS = 24
SAMPLE_TYPE = MSB_INTEGER
STRETCHED_FLAG = FALSE
INVALID_CONSTANT = 0
OFFSET = 0.0
SCALING_FACTOR = 0.0
START_TIME = 2008-06-01T00:00:00
STOP_TIME = 2008-06-01T23:59:59
END_OBJECT = IMAGE
END

```

6.3 データオブジェクトフォーマット

「電子・イオンエネルギースペクトル時系列サマリプロット」は画像データであり、PNG形式で作成される。

6.4 カタログ情報ファイルフォーマット

PACE プロダクトの IMAGE オブジェクトカタログ情報ファイルフォーマットを表 6-2 に示す。

表 6-2IMAGE オブジェクトカタログ情報ファイルフォーマット

項目名	要素	設定値のフォーマット	設定値の範囲	設定値
データファイル名(*1)	DataFileName	AAAA...AAAA (最大31 桁)	任意の英数字	プロダクトによる (6.1 節 ファイル命名規約 参照)
データファイルサイズ	DataFileSize	NNNNNNNNNNNN (最大12 桁)	単位:バイト	プロダクトによる
データファイルフォーマット	DataFileFormat	AAAA...AAAA (最大16 桁)	任意の文字列	PDS【固定】
機器名	InstrumentName	AAAA...AAAA (最大16 桁)	任意の文字列	PACE【固定】
処理レベル	ProcessingLevel	AAAA...AAAA (最大16 桁)	任意の文字列	プロダクトによる(表 1-2 処理レベル 参照)
プロダクト種別	ProductID	AAAA...AAAA (最大30 桁)	任意の文字列	プロダクトによる(表 1-2 Product ID 参照)
プロダクトバージョン	ProductVersion	AAAA...AAAA (最大16 桁)	任意の文字列	プロダクトによる
アクセスレベル	AccessLevel	N	0-4 の数値	N/A

(*1) データファイル名にはプロダクトのファイル名を格納する。デタッチド形式の場合は、データファイルの名前を格納する。

【カタログ情報ファイルサンプル: 電子・イオンエネルギースペクトル時系列サマリプロット プロダクト】

DataFileName = PACE_SI_MAP_001.img
 DataFileSize = 581041
 DataFileFormat = PDS
 InstrumentName = PACE
 ProcessingLevel = Higher Level
 ProductID = PACE_SI_MAP
 ProductVersion = 1.0
 AccessLevel = 4

添付資料-1

Saito et al. 2008("Low-energy charged particle measurement by
MAP-PACE onboard SELENE")

添付資料-2
paceql_outputdata_070904.h

```

/*
SELENE PACE level2B data header file
    version    0.4        15 February 2007
                mode 01,02,11,12,13,14,24,15,25,16,26,17,18,19,29,1A,2A,1B, (+80:internal count)
                mode 72->(17),73->(11),74->(17), 71, 78, 70
    version    1.0        05 March 2007
    version    1.1        01 May 2007
    version 1.2 14 June 2007
    version 1.3 18 June 2007
    version 1.4 19 June 2007
    version 1.5 20 June 2007
    version 1.6 30 June 2007
    version 2.0 08 August 2007
    version 2.1 10 August 2007
    version 3.0 23 August 2007
    version 4.0 04 September 2007

```

```
*/
```

```

#define          UCHAR          unsigned char
#define          USHORT         unsigned short
#define          ULONG          unsigned long

```

```

/*****
/*****
/* HEADER
/*****
/*****
*/

```

```

typedef struct Header_PACE {
    ULONG        sensor; /* 0 ESA-S1  1 ESA-S2  2 IMA  3 IEA 4 ALL */
    ULONG        mode; /* data mode = data mode command */
    ULONG        mode2; /* sub-data mode */
    ULONG        type; /* data type */
    ULONG        size; /* data size */
    ULONG        time_resolution; /* time resolution (msec) */
    ULONG        sampl_time; /* sampling time (16000/*** msec)*/
    ULONG        ver; /* data version */
    ULONG        tbl_ver; /* onboard table version */
    ULONG        obs_ver; /* onboard software version */
    ULONG        timeH; /* 1pps TI High Word */
    ULONG        timeM; /* 1pps TI Medium Word */
    ULONG        timeL; /* 1pps TI Low Word */
    ULONG        bc; /* base clock */
    ULONG        ic; /* increment counter */
    ULONG        sc; /* base counter */
    ULONG        sc_step0; /* sc @ energy sweep 0 */
    ULONG        t_date; /* total date */
    ULONG        time_ms; /* msec of day @ energy sweep 0 */
    ULONG        yyymmdd;
    ULONG        hhmmss;
    ULONG        tof_tbl; /* IMA */
    ULONG        pd_pha;
    ULONG        svg_tbl; /* IMA IEA */
    ULONG        sva_tbl;
    ULONG        svb_tbl;
    ULONG        obs_tbl;
    ULONG        obs_ctr;
    ULONG        nv_high;
    ULONG        nv_low;
    ULONG        data_quality; /* data quality */
    ULONG        pol_step; /* polar angle step number */
    ULONG        az_step; /* azimuthal angle step number */
    ULONG        ene_step; /* energy step number */
    ULONG        mass_step; /* mass step number */
    ULONG        pitch_step; /* pitch angle step number */
    ULONG        tof_step; /* tof step number */
    ULONG        solwnd_step; /* solar wind number */
    ULONG        exb_step; /* ExB number */
    ULONG        event_step; /* event counter number */

```

```

        ULONG   trash_step; /* trash counter number */
        ULONG   tof_disc_start; /* TOF DISCRI SCAN h'73 MODE IMA IEA ONLY */
        ULONG   tof_disc_stop; /* TOF DISCRI SCAN h'73 MODE IMA IEA ONLY */
        ULONG   hv_scan_level; /* 1Byte HV SCAN h'72 MODE ONLY */
        ULONG   spare[20]; /* total header 256bytes = 64 long */
} H_P_t;

```

```

/*===== ESA DATA TYPE =====*/

```

```

/* ----- TYPE 00 ----- */
struct D_ESA_TYPE00 {
        ULONG   event[16];
        USHORT  cnt[32][16][64];
        USHORT  trash[32][16][2];
};

```

```

/* ----- TYPE 01 ----- */
struct D_ESA_TYPE01 {
        ULONG   event[16];
        USHORT  cnt[32][4][16];
        USHORT  trash[32][4][2];
};

```

```

/* ----- TYPE 02 ----- */
struct D_ESA_TYPE02 {
        ULONG   event[16];
        USHORT  cnt[32][32];
};

```

```

/* ----- TYPE 03 ----- */
struct D_ESA_TYPE03 {
        ULONG   event[16];
        USHORT  cnt[32][8][64];
        USHORT  trash[32][8][2];
};

```

```

/*===== IMA DATA TYPE =====*/

```

```

/* ----- TYPE 40 ----- */
struct D_IMA_TYPE40 {
        ULONG   event[4][16];
        USHORT  cnt[4][32][1024];
};

```

```

/* ----- TYPE 41 ----- */
struct D_IMA_TYPE41 {
        ULONG   event[4][16];
        USHORT  cnt[32][16][64];
        USHORT  trash[32][16][2];
};

```

```

/* ----- TYPE 42 ----- */
struct D_IMA_TYPE42 {
        ULONG   event[4][16];
        USHORT  cnt[32][4][16];
        USHORT  trash[32][4][2];
};

```

```

/* ----- TYPE 43 ----- */
struct D_IMA_TYPE43 {
        ULONG   event[4][16];
        USHORT  cnt[8][32][4][16];
        USHORT  trash[8][32][4][2];
};

```

```

/* ----- TYPE 44 ----- */
struct D_IMA_TYPE44 {
        ULONG   event[4][16];
        USHORT  s_cnt[16][32][64];
        USHORT  cnt[16][32][16][64];
};

```

```

/* ----- TYPE 45 ----- */
struct D_IMA_TYPE45 {
        ULONG   event[4][16];
        USHORT  cnt[16][32][4][16];
        USHORT  trash[16][32][4][2];
};

```

```

/*===== IEA DATA TYPE =====*/

```

```

/* ----- TYPE 80 ----- */
struct D_IEA_TYPE80 {
        ULONG   event[16];
};

```



```

        USHORT cnt[32][4][16];
        USHORT trash[32][4][2];
};
/* ----- TYPE 81 ----- */
struct D_IEA_TYPE81 {
        ULONG event[16];
        USHORT cnt[32][16][64];
        USHORT trash[32][16][2];
};
/* ----- TYPE 82 ----- */
struct D_IEA_TYPE82 {
        ULONG event[16];
        USHORT s_cnt[32][128];
        USHORT cnt[32][16][64];
};

/*===== SPECIAL DATA TYPE =====*/
/* ----- TYPE SV_MONITOR ----- */
struct D_TYPEF0_ESA_SV_MONITOR {
        USHORT sv_unit;
        USHORT sp[1024];
        USHORT ang1[1024];
        USHORT ang2[1024];
};
/* ----- TYPE SV_MONITOR ----- */
struct D_TYPEF1_IMEA_SV_MONITOR {
        USHORT sv_unit;
        USHORT sp[1024];
        USHORT ang1[1024];
        USHORT ang2[1024];
        USHORT g[1024];
};

/* ----- TYPE LMAG_MONITOR ----- */
struct D_TYPEF2_LMAG_MONITOR {
        ULONG time;
        ULONG B[16];
        ULONG x[16];
        ULONG y[16];
        ULONG z[16];
};

/* ----- TYPE INTERNAL_INFORMATION ----- */
struct D_TYPEF3_INTERNAL_INFORMATION {
        ULONG ir1;
        USHORT esa_s1_ir1_addr;
        USHORT esa_s2_ir1_addr;
        USHORT ima_ir1_addr;
        USHORT iea_ir1_addr;
        ULONG esa_s1_intr;
        ULONG esa_s2_intr;
        ULONG ima_intr;
        ULONG iea_intr;
        USHORT tof_disc_start_org;
        USHORT tof_disc_flg;
        ULONG hv_scan_start;
        ULONG hv_scan_gap;
        ULONG esa_s1_sva1ovsvs_p;
        ULONG esa_s2_sva1ovsvs_p;
        ULONG ima_sva1ovsvs_p;
        ULONG iea_sva1ovsvs_p;
        ULONG esa_s1_sva2ovsvs_p;
        ULONG esa_s2_sva2ovsvs_p;
        ULONG ima_sva2ovsvs_p;
        ULONG iea_sva2ovsvs_p;
        ULONG esa_s1_sva1ovsvs_h;
        ULONG esa_s2_sva1ovsvs_h;
        ULONG ima_sva1ovsvs_h;
        ULONG iea_sva1ovsvs_h;
        ULONG esa_s1_sva2ovsvs_h;
        ULONG esa_s2_sva2ovsvs_h;
        ULONG ima_sva2ovsvs_h;
        ULONG iea_sva2ovsvs_h;
        USHORT tof_disc_start;
};

```

```

USHORT tof_disc_start_mon;
USHORT tof_disc_stop;
USHORT tof_disc_stop_mon;
USHORT tof_disc_start_scan1;
USHORT tof_disc_start_scan2;
USHORT tof_disc_start_scan3;
USHORT tof_disc_start_scan4;
USHORT tof_disc_stop_scan1;
USHORT tof_disc_stop_scan2;
USHORT tof_disc_stop_scan3;
USHORT tof_disc_stop_scan4;
ULONG nv_l;
ULONG nv_l_cmd;
ULONG nv_h;
ULONG nv_h_cmd;
ULONG errbufclr_ctr;
USHORT esa_s1_obs_mode;
USHORT esa_s2_obs_mode;
USHORT ima_obs_mode;
USHORT iea_obs_mode;
USHORT esa_s1_ena_dis;
USHORT esa_s2_ena_dis;
USHORT ima_ena_dis;
USHORT iea_ena_dis;
USHORT esa_s1_clk_status;
USHORT esa_s2_clk_status;
USHORT ima_clk_status;
USHORT iea_clk_status;
USHORT tof_disc_start2;
USHORT tof_disc_stop2;
};

/*=====*/

/*****/
/* ELECTRON MODE EC-N */
/* Electron Check Mode 0x00 */
/* 16X64(mode1) 16s RAM67 NB TYPE00 */
/*****/
typedef struct D_ESA_TYPE00 D_ESAS1_TYPE00_EC_N_t; /* ESAS1 */
typedef struct D_ESA_TYPE00 D_ESAS2_TYPE00_EC_N_t; /* ESAS2 */

/*****/
/* ELECTRON MODE EM_N */
/* Electron Magnetosphere Noraml Mode 0x01 */
/* 4X16(mode3) 2s RAM67 NB TYPE01 */
/*****/
typedef struct D_ESA_TYPE01 D_ESAS1_TYPE01_EM_N_t; /* ESAS1 */
typedef struct D_ESA_TYPE01 D_ESAS2_TYPE01_EM_N_t; /* ESAS2 */

/*****/
/* ELECTRON MODE EM_H */
/* Electron Magnetosphere High Mode 0x08 */
/* 4X16(mode3) 1s RAM67 NB TYPE01 */
/*****/
typedef struct D_ESA_TYPE01 D_ESAS1_TYPE01_EM_H_t; /* ESAS1 */
typedef struct D_ESA_TYPE01 D_ESAS2_TYPE01_EM_H_t; /* ESAS2 */

/*****/
/* ELECTRON MODE EM_R */
/* Electron Magnetosphere No Compression Mode 0x02 */
/* 4X16(mode3) 4s RAM67 NB TYPE01 */
/*****/
typedef struct D_ESA_TYPE01 D_ESAS1_TYPE01_EM_R_t; /* ESAS1 */
typedef struct D_ESA_TYPE01 D_ESAS2_TYPE01_EM_R_t; /* ESAS2 */

/*****/

```

```

/* ELECTRON MODE ER_N */
/* Electron Electron Reflectometer Normal Mode 0x03 */
/* 16X64(mode1) 16s(2s) RAM0123 NB TYPE00 */
/*****/
typedef struct D_ESA_TYPE00 D_ESAS1_TYPE00_ER_N_t; /* ESAS1 */
typedef struct D_ESA_TYPE00 D_ESAS2_TYPE00_ER_N_t; /* ESAS2 */

/*****/
/* ELECTRON MODE ER_H */
/* Electron Electron Reflectometer High Mode 0x08 */
/* 16X64(mode1) 8s(1s) RAM0123 B TYPE02 */
/*****/
typedef struct D_ESA_TYPE02 D_ESAS1_TYPE02_ER_H_t; /* ESAS1 */
typedef struct D_ESA_TYPE02 D_ESAS2_TYPE02_ER_H_t; /* ESAS2 */

/*****/
/* ELECTRON MODE ER_W */
/* Electron Electron Reflectometer Wake Mode 0x04 */
/* 16X64(mode1) 8s(1s) RAM0123 NB POL8 TYPE03 */
/*****/
typedef struct D_ESA_TYPE03 D_ESAS1_TYPE03_ER_W_t; /* ESAS1 */
typedef struct D_ESA_TYPE03 D_ESAS2_TYPE03_ER_W_t; /* ESAS2 */

/*****/
/* ELECTRON MODE ER_B */
/* Electron Electron Reflectometer Backup Mode 0x05 */
/* 16X64(mode1) 16s(4s) RAM45 NB POL16 TYPE00 */
/*****/
typedef struct D_ESA_TYPE00 D_ESAS1_TYPE00_ER_B_t; /* ESAS1 */
typedef struct D_ESA_TYPE00 D_ESAS2_TYPE00_ER_B_t; /* ESAS2 */

/*****/
/* ELECTRON MODE ER_R */
/* Electron Electron Reflectometer No Compression Mode 0x07 */
/* 16X64(mode1) 8s(1s) RAM0123 B TYPE02 */
/*****/
typedef struct D_ESA_TYPE02 D_ESAS1_TYPE02_ER_R_t; /* ESAS1 */
typedef struct D_ESA_TYPE02 D_ESAS2_TYPE02_ER_R_t; /* ESAS2 */

/*****/
/* ELECTRON MODE PM-1 */
/* Electron PM NORMAL 1 Mode 0x21 */
/* 4X16(mode2) 16s RAM01234567 NB TYPE01 */
/*****/
typedef struct D_ESA_TYPE01 D_ESAS1_TYPE01_PM_1_t; /* ESAS1 */
typedef struct D_ESA_TYPE01 D_ESAS2_TYPE01_PM_1_t; /* ESAS2 */

/*****/
/* ELECTRON MODE PM-2 */
/* Electron PM NORMAL 2 Mode 0x22 */
/* 16X64(mode1) 32s RAM01234567 NB TYPE00 */
/*****/
typedef struct D_ESA_TYPE00 D_ESAS1_TYPE00_PM_2_t; /* ESAS1 */
typedef struct D_ESA_TYPE00 D_ESAS2_TYPE00_PM_2_t; /* ESAS2 */

/*-----*/

/*****/
/* ION MODE IC_T */
/* Ion Time Check Mode 0x10 */
/* IMA 4X1X1024(direct1) 16s RAM0123 MASS1024 TYPE40 */
/* IEA 4X16(mode3) 16s RAM0123 TYPE80 */
/*****/
typedef struct D_IMA_TYPE40 D_IMA_TYPE40_IC_T_t; /* IMA */
typedef struct D_IEA_TYPE80 D_IEA_TYPE80_IC_T_t; /* IEA */

```

```

/*****/
/* ION MODE IC_P */
/* Ion Position Check Mode 0x11 */
/* IMA 16X64X16(mode1) 16s RAM0123 MASS1 TYPE41 */
/* IEA 16X64(mode1) 16s RAM0123 TYPE81 */
typedef struct D_IMA_TYPE41 D_IMA_TYPE41_IC_P_t; /* IMA */
typedef struct D_IEA_TYPE81 D_IEA_TYPE81_IC_P_t; /* IEA */

/*****/
/* ION MODE IM_N */
/* Ion Magnetosphere Normal Mode 0x12 */
/* IMA 4X16X16(mode3) 1s RAM0123 MASS1 TYPE42 */
/* IEA 4X16(mode3) 1s RAM0123 TYPE80 */
typedef struct D_IMA_TYPE42 D_IMA_TYPE42_IM_N_t; /* IMA */
typedef struct D_IEA_TYPE80 D_IEA_TYPE80_IM_N_t; /* IEA */

/*****/
/* ION MODE IM_L */
/* Ion Magnetosphere Low Mode 0x19 */
/* IMA 4X16X16(mode3) 2s RAM0123 MASS1 TYPE42 */
/* IEA 4X16(mode3) 2s RAM0123 TYPE80 */
typedef struct D_IMA_TYPE42 D_IMA_TYPE42_IM_L_t; /* IMA */
typedef struct D_IEA_TYPE80 D_IEA_TYPE80_IM_L_t; /* IEA */

/*****/
/* ION MODE IL_S */
/* Ion Lunar Solar Wind Mode 0x13 */
/* IMA 4X16X16(mode3) 8s RAM0 MASS8 TYPE43 */
/* IEA 16X64(mode1) 2s RAM0 128PT TYPE82 */
typedef struct D_IMA_TYPE43 D_IMA_TYPE43_IL_S_t; /* IMA */
typedef struct D_IEA_TYPE82 D_IEA_TYPE82_IL_S_t; /* IEA */

/*****/
/* ION MODE IL_V */
/* Ion Lunar WAKE Mode 0x14 */
/* IMA 4X16X16(mode3) 8s RAM1 MASS8 TYPE43 */
/* IEA 16X64(mode1) 8s RAM1 TYPE81 */
typedef struct D_IMA_TYPE43 D_IMA_TYPE43_IL_V_t; /* IMA */
typedef struct D_IEA_TYPE81 D_IEA_TYPE81_IL_V_t; /* IEA */

/*****/
/* ION MODE IL_M */
/* Ion Lunar Ion Mass Survey Mode 0x15 */
/* IMA 4X16X64(drct3) 16s RAM0123 Time Profile TYPE45 */
/* IEA 4X16(mode3) 16s RAM0123 TYPE80 */
typedef struct D_IMA_TYPE45 D_IMA_TYPE45_IL_M_t; /* IMA */
typedef struct D_IEA_TYPE80 D_IEA_TYPE80_IL_M_t; /* IEA */

/*****/
/* ION MODE IL_R */
/* Ion Lunar Ion Wake No Compression Mode 0x16 */
/* IMA 4X16X16(mode3) 16s RAM1 MASS8 TYPE43 */
/* IEA 4X16(mode3) 16s RAM1 TYPE80 */
typedef struct D_IMA_TYPE43 D_IMA_TYPE43_IL_R_t; /* IMA */
typedef struct D_IEA_TYPE80 D_IEA_TYPE80_IL_R_t; /* IEA */

/*****/
/* ION MODE IS_A */
/* Ion Lunar Ion Sputtering Mode 0x17 */
/* IMA 16X64X16(mode1) 16s(4s) RAM2 8ENERGY MASS16 TYPE44 */

```

```

/*      IEA 16X64(mode1) 2s RAM0 128PT      TYPE82      */
/*****/
typedef struct          D_IMA_TYPE44 D_IMA_TYPE44_IS_A_t; /* IMA */
typedef struct          D_IEA_TYPE82 D_IEA_TYPE82_IS_A_t; /* IEA */

/*****/
/* ION MODE PM_1      */
/* Ion PM NORMAL1 Mode 0x21      */
/*      IMA 16X4X16(mode2) 16s RAM0123 MASS16 TYPE45      */
/*      IEA 4X16(mode2) 16s RAM0123      TYPE80      */
/*****/
typedef struct          D_IMA_TYPE45 D_IMA_TYPE45_PM_1_t; /* IMA */
typedef struct          D_IEA_TYPE80 D_IEA_TYPE80_PM_1_t; /* IEA */

/*****/
/* ION MODE PM_2      */
/* Ion PM NORMAL2 Mode 0x22      */
/*      IMA 16X16X64(mode1) 32s RAM0123 MASS1 TYPE41      */
/*      IEA 16X64(mode1) 32s RAM0123      TYPE81      */
/*****/
typedef struct          D_IMA_TYPE41 D_IMA_TYPE41_PM_2_t; /* IMA */
typedef struct          D_IEA_TYPE81 D_IEA_TYPE81_PM_2_t; /* IEA */

/*****/
/* SPECIAL MODE SV_MONITOR      */
/* SV MONITOR MODE Command Mode 0x0071      */
/*      ESA-S1 sp 1024 , ang1 1024 , ang2 1024      */
/*      ESA-S2 sp 1024 , ang1 1024 , ang2 1024      */
/*      IMA sp 1024 , ang1 1024 , ang2 1024 , g 1024      */
/*      IEA sp 1024 , ang1 1024 , ang2 1024 , g 1024      */
/*****/
typedef struct          D_TYPEF0_ESA_SV_MONITOR          D_TYPEF0_ESAS1_SV_MONITOR_t; /*
ESAS1 */
typedef struct          D_TYPEF0_ESA_SV_MONITOR          D_TYPEF0_ESAS2_SV_MONITOR_t; /*
ESAS2 */
typedef struct          D_TYPEF1_IMEA_SV_MONITOR D_TYPEF1_IMA_SV_MONITOR_t; /* IMA */
typedef struct          D_TYPEF1_IMEA_SV_MONITOR D_TYPEF1_IEA_SV_MONITOR_t; /* IEA */

/*****/
/* SPECIAL MODE LMAG_MONITOR      */
/* LMAG MONITOR MODE Command Mode 0x0078      */
/*****/
typedef struct          D_TYPEF2_LMAG_MONITOR          D_TYPEF2_LMAG_MONITOR_t;

/*****/
/* SPECIAL MODE INTERNAL_INFORMATION      */
/* INTERNAL INFORMATION MODE Command Mode 0x0070      */
/*****/
typedef struct          D_TYPEF3_INTERNAL_INFORMATION          D_TYPEF3_INTERNAL_INFORMATION_t;

/*****/
/* data type definition ESA      */
/*****/
typedef struct D_ESA_TYPE00 D_ESAS1_TYPE00_t; /* ESAS1 */
typedef struct D_ESA_TYPE00 D_ESAS2_TYPE00_t; /* ESAS2 */
typedef struct D_ESA_TYPE01 D_ESAS1_TYPE01_t; /* ESAS1 */
typedef struct D_ESA_TYPE01 D_ESAS2_TYPE01_t; /* ESAS2 */
typedef struct D_ESA_TYPE02 D_ESAS1_TYPE02_t; /* ESAS1 */
typedef struct D_ESA_TYPE02 D_ESAS2_TYPE02_t; /* ESAS2 */
typedef struct D_ESA_TYPE03 D_ESAS1_TYPE03_t; /* ESAS1 */
typedef struct D_ESA_TYPE03 D_ESAS2_TYPE03_t; /* ESAS2 */
/*****/
/* data type definition IMA      */
/*****/

```

```

typedef struct D_IMA_TYPE40 D_IMA_TYPE40_t;      /* IMA */
typedef struct D_IMA_TYPE41 D_IMA_TYPE41_t;      /* IMA */
typedef struct D_IMA_TYPE42 D_IMA_TYPE42_t;      /* IMA */
typedef struct D_IMA_TYPE43 D_IMA_TYPE43_t;      /* IMA */
typedef struct D_IMA_TYPE44 D_IMA_TYPE44_t;      /* IMA */
typedef struct D_IMA_TYPE45 D_IMA_TYPE45_t;      /* IMA */
/******/
/* data type definition IEA                               */
/******/
typedef struct D_IEA_TYPE80 D_IEA_TYPE80_t;      /* IEA */
typedef struct D_IEA_TYPE81 D_IEA_TYPE81_t;      /* IEA */
typedef struct D_IEA_TYPE82 D_IEA_TYPE82_t;      /* IEA */

```

```

/******/
/******/
/* MODE 01 PM NORMAL1                                     */
/******/
/******/
typedef struct S_ESAS1_MODE_01 {
    H_P_t      header;
    D_ESAS1_TYPE01_PM_1_t      data;
} S_ESAS1_MODE_01_t;

typedef struct S_ESAS2_MODE_01 {
    H_P_t      header;
    D_ESAS2_TYPE01_PM_1_t      data;
} S_ESAS2_MODE_01_t;

typedef struct S_IMA_MODE_01 {
    H_P_t      header;
    D_IMA_TYPE45_PM_1_t      data;
} S_IMA_MODE_01_t;

typedef struct S_IEA_MODE_01 {
    H_P_t      header;
    D_IEA_TYPE80_PM_1_t      data;
} S_IEA_MODE_01_t;

```

```

/******/
/******/
/* MODE 02 PM NORMAL2                                     */
/******/
/******/
typedef struct S_ESAS1_MODE_02 {
    H_P_t      header;
    D_ESAS1_TYPE00_PM_2_t      data;
} S_ESAS1_MODE_02_t;

typedef struct S_ESAS2_MODE_02 {
    H_P_t      header;
    D_ESAS2_TYPE00_PM_2_t      data;
} S_ESAS2_MODE_02_t;

typedef struct S_IMA_MODE_02 {
    H_P_t      header;
    D_IMA_TYPE41_PM_2_t      data;
} S_IMA_MODE_02_t;

typedef struct S_IEA_MODE_02 {
    H_P_t      header;
    D_IEA_TYPE81_PM_2_t      data;
} S_IEA_MODE_02_t;

```

```

/******/
/******/

```

```

/* MODE 11 TOFCAL */
/*****
/*****
typedef struct S_ESAS1_MODE_11 {
    H_P_t    header;
    D_ESAS1_TYPE00_EC_N_t    data;
} S_ESAS1_MODE_11_t;

typedef struct S_ESAS2_MODE_11 {
    H_P_t    header;
    D_ESAS2_TYPE00_EC_N_t    data;
} S_ESAS2_MODE_11_t;

typedef struct S_IMA_MODE_11 {
    H_P_t    header;
    D_IMA_TYPE40_IC_T_t    data;
} S_IMA_MODE_11_t;

typedef struct S_IEA_MODE_11 {
    H_P_t    header;
    D_IEA_TYPE80_IC_T_t    data;
} S_IEA_MODE_11_t;

/*****
/*****
/* MODE 12 POSCAL */
/*****
/*****
typedef struct S_ESAS1_MODE_12 {
    H_P_t    header;
    D_ESAS1_TYPE00_EC_N_t    data;
} S_ESAS1_MODE_12_t;

typedef struct S_ESAS2_MODE_12 {
    H_P_t    header;
    D_ESAS2_TYPE00_EC_N_t    data;
} S_ESAS2_MODE_12_t;

typedef struct S_IMA_MODE_12 {
    H_P_t    header;
    D_IMA_TYPE41_IC_P_t    data;
} S_IMA_MODE_12_t;

typedef struct S_IEA_MODE_12 {
    H_P_t    header;
    D_IEA_TYPE81_IC_P_t    data;
} S_IEA_MODE_12_t;

/*****
/*****
/* MODE 13 E2s1s */
/*****
/*****
typedef struct S_ESAS1_MODE_13 {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_13_t;

typedef struct S_ESAS2_MODE_13 {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_13_t;

typedef struct S_IMA_MODE_13 {
    H_P_t    header;
    D_IMA_TYPE42_IM_N_t    data;
} S_IMA_MODE_13_t;

typedef struct S_IEA_MODE_13 {
    H_P_t    header;
    D_IEA_TYPE80_IM_N_t    data;
} S_IEA_MODE_13_t;

```

```

/*****/
/*****/
/* MODE 14 3D MASS SOLAR WIND */
/*****/
/*****/
typedef struct S_ESAS1_MODE_14 {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_14_t;

typedef struct S_ESAS2_MODE_14 {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_14_t;

typedef struct S_IMA_MODE_14 {
    H_P_t    header;
    D_IMA_TYPE43_IL_S_t    data;
} S_IMA_MODE_14_t;

typedef struct S_IEA_MODE_14 {
    H_P_t    header;
    D_IEA_TYPE82_IL_S_tdata;
} S_IEA_MODE_14_t;

/*****/
/*****/
/* MODE 24 3D MASS WAKE */
/*****/
/*****/
typedef struct S_ESAS1_MODE_24 {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_24_t;

typedef struct S_ESAS2_MODE_24 {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_24_t;

typedef struct S_IMA_MODE_24 {
    H_P_t    header;
    D_IMA_TYPE43_IL_V_t    data;
} S_IMA_MODE_24_t;

typedef struct S_IEA_MODE_24 {
    H_P_t    header;
    D_IEA_TYPE81_IL_V_tdata;
} S_IEA_MODE_24_t;

/*****/
/*****/
/* MODE 15 ER LUNAR ION SOLAR WIND */
/*****/
/*****/
typedef struct S_ESAS1_MODE_15E {
    H_P_t    header;
    D_ESAS1_TYPE02_ER_H_t    data;
} S_ESAS1_MODE_15E_t;
typedef struct S_ESAS1_MODE_15T {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_15T_t;

typedef struct S_ESAS2_MODE_15E {
    H_P_t    header;
    D_ESAS2_TYPE02_ER_H_t    data;
} S_ESAS2_MODE_15E_t;
typedef struct S_ESAS2_MODE_15T {
    H_P_t    header;

```



```

        D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_15T_t;

typedef struct S_IMA_MODE_15 {
        H_P_t    header;
        D_IMA_TYPE43_IL_S_t    data;
} S_IMA_MODE_15_t;

typedef struct S_IEA_MODE_15 {
        H_P_t    header;
        D_IEA_TYPE82_IL_S_tdata;
} S_IEA_MODE_15_t;

/*****/
/*****/
/* MODE 25 ER LUNAR ION WAKE */
/*****/
/*****/
typedef struct S_ESAS1_MODE_25E {
        H_P_t    header;
        D_ESAS1_TYPE03_ER_W_t    data;
} S_ESAS1_MODE_25E_t;
typedef struct S_ESAS1_MODE_25T {
        H_P_t    header;
        D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_25T_t;

typedef struct S_ESAS2_MODE_25E {
        H_P_t    header;
        D_ESAS2_TYPE03_ER_W_t    data;
} S_ESAS2_MODE_25E_t;
typedef struct S_ESAS2_MODE_25T {
        H_P_t    header;
        D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_25T_t;

typedef struct S_IMA_MODE_25 {
        H_P_t    header;
        D_IMA_TYPE43_IL_V_t    data;
} S_IMA_MODE_25_t;

typedef struct S_IEA_MODE_25 {
        H_P_t    header;
        D_IEA_TYPE81_IL_V_tdata;
} S_IEA_MODE_25_t;

/*****/
/*****/
/* MODE 16 ER SPUTTERING ION SOLAR WIND */
/*****/
/*****/
typedef struct S_ESAS1_MODE_16E {
        H_P_t    header;
        D_ESAS1_TYPE02_ER_H_t    data;
} S_ESAS1_MODE_16E_t;
typedef struct S_ESAS1_MODE_16T {
        H_P_t    header;
        D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_16T_t;

typedef struct S_ESAS2_MODE_16E {
        H_P_t    header;
        D_ESAS2_TYPE02_ER_H_t    data;
} S_ESAS2_MODE_16E_t;
typedef struct S_ESAS2_MODE_16T {
        H_P_t    header;
        D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_16T_t;

typedef struct S_IMA_MODE_16 {
        H_P_t    header;
        D_IMA_TYPE44_IS_A_t    data;

```

```

} S_IMA_MODE_16_t;

typedef struct S_IEA_MODE_16 {
    H_P_t    header;
    D_IEA_TYPE82_IS_A_t    data;
} S_IEA_MODE_16_t;

/*****
/*****
/* MODE 26 ER SPUTTERING ION WAKE */
/*****
/*****
typedef struct S_ESAS1_MODE_26E {
    H_P_t    header;
    D_ESAS1_TYPE03_ER_W_t    data;
} S_ESAS1_MODE_26E_t;
typedef struct S_ESAS1_MODE_26T {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_26T_t;

typedef struct S_ESAS2_MODE_26E {
    H_P_t    header;
    D_ESAS2_TYPE03_ER_W_t    data;
} S_ESAS2_MODE_26E_t;
typedef struct S_ESAS2_MODE_26T {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_26T_t;

typedef struct S_IMA_MODE_26 {
    H_P_t    header;
    D_IMA_TYPE44_IS_A_t    data;
} S_IMA_MODE_26_t;

typedef struct S_IEA_MODE_26 {
    H_P_t    header;
    D_IEA_TYPE82_IS_A_t    data;
} S_IEA_MODE_26_t;

/*****
/*****
/* MODE 17 LUNAR ION WAKE NO COMPRESSION */
/*****
/*****
typedef struct S_ESAS1_MODE_17 {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_R_t    data;
} S_ESAS1_MODE_17_t;

typedef struct S_ESAS2_MODE_17 {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_R_t    data;
} S_ESAS2_MODE_17_t;

typedef struct S_IMA_MODE_17 {
    H_P_t    header;
    D_IMA_TYPE43_IL_R_t    data;
} S_IMA_MODE_17_t;

typedef struct S_IEA_MODE_17 {
    H_P_t    header;
    D_IEA_TYPE80_IL_R_t    data;
} S_IEA_MODE_17_t;

/*****
/*****
/* MODE 18 ER LUNAR ION NO COMPRESSION */
/*****
/*****
typedef struct S_ESAS1_MODE_18E {

```

```

        H_P_t    header;
        D_ESAS1_TYPE02_ER_R_t    data;
    } S_ESAS1_MODE_18E_t;
typedef struct S_ESAS1_MODE_18T {
        H_P_t    header;
        D_ESAS1_TYPE01_EM_R_t    data;
    } S_ESAS1_MODE_18T_t;

typedef struct S_ESAS2_MODE_18E {
        H_P_t    header;
        D_ESAS2_TYPE02_ER_R_t    data;
    } S_ESAS2_MODE_18E_t;
typedef struct S_ESAS2_MODE_18T {
        H_P_t    header;
        D_ESAS2_TYPE01_EM_R_t    data;
    } S_ESAS2_MODE_18T_t;

typedef struct S_IMA_MODE_18 {
        H_P_t    header;
        D_IMA_TYPE43_IL_R_t    data;
    } S_IMA_MODE_18_t;

typedef struct S_IEA_MODE_18 {
        H_P_t    header;
        D_IEA_TYPE80_IL_R_t    data;
    } S_IEA_MODE_18_t;

```

```

/*****/
/*****/
/* MODE 19 ER LUNAR ION SOLAR WIND BACKUP
/*****/
/*****/

```

*/

```

typedef struct S_ESAS1_MODE_19E {
        H_P_t    header;
        D_ESAS1_TYPE02_ER_H_t    data;
    } S_ESAS1_MODE_19E_t;
typedef struct S_ESAS1_MODE_19T {
        H_P_t    header;
        D_ESAS1_TYPE01_EM_N_t    data;
    } S_ESAS1_MODE_19T_t;

typedef struct S_ESAS2_MODE_19E {
        H_P_t    header;
        D_ESAS2_TYPE02_ER_H_t    data;
    } S_ESAS2_MODE_19E_t;
typedef struct S_ESAS2_MODE_19T {
        H_P_t    header;
        D_ESAS2_TYPE01_EM_N_t    data;
    } S_ESAS2_MODE_19T_t;

typedef struct S_IMA_MODE_19 {
        H_P_t    header;
        D_IMA_TYPE43_IL_S_t    data;
    } S_IMA_MODE_19_t;

typedef struct S_IEA_MODE_19 {
        H_P_t    header;
        D_IEA_TYPE82_IL_S_tdata;
    } S_IEA_MODE_19_t;

```

```

/*****/
/*****/
/* MODE 29 ER LUNAR ION WAKE BACKUP
/*****/
/*****/

```

*/

```

typedef struct S_ESAS1_MODE_29E {
        H_P_t    header;
        D_ESAS1_TYPE00_ER_N_t    data;
    } S_ESAS1_MODE_29E_t;
typedef struct S_ESAS1_MODE_29T {
        H_P_t    header;
        D_ESAS1_TYPE01_EM_N_t    data;
    } S_ESAS1_MODE_29T_t;

```

```

} S_ESAS1_MODE_29T_t;

typedef struct S_ESAS2_MODE_29E {
    H_P_t    header;
    D_ESAS2_TYPE00_ER_N_t    data;
} S_ESAS2_MODE_29E_t;
typedef struct S_ESAS2_MODE_29T {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_29T_t;

typedef struct S_IMA_MODE_29 {
    H_P_t    header;
    D_IMA_TYPE43_IL_V_t    data;
} S_IMA_MODE_29_t;

typedef struct S_IEA_MODE_29 {
    H_P_t    header;
    D_IEA_TYPE81_IL_V_tdata;
} S_IEA_MODE_29_t;

/*****
/*****
/* MODE 1A ER SPUTTERING ION SOLAR WIND BACKUP          */
/*****
/*****
typedef struct S_ESAS1_MODE_1AE {
    H_P_t    header;
    D_ESAS1_TYPE02_ER_H_t    data;
} S_ESAS1_MODE_1AE_t;
typedef struct S_ESAS1_MODE_1AT {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_1AT_t;

typedef struct S_ESAS2_MODE_1AE {
    H_P_t    header;
    D_ESAS2_TYPE02_ER_H_t    data;
} S_ESAS2_MODE_1AE_t;
typedef struct S_ESAS2_MODE_1AT {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_1AT_t;

typedef struct S_IMA_MODE_1A {
    H_P_t    header;
    D_IMA_TYPE44_IS_A_t    data;
} S_IMA_MODE_1A_t;

typedef struct S_IEA_MODE_1A {
    H_P_t    header;
    D_IEA_TYPE82_IS_A_t    data;
} S_IEA_MODE_1A_t;

/*****
/*****
/* MODE 2A ER SPUTTERING ION WAKE BACKUP          */
/*****
/*****
typedef struct S_ESAS1_MODE_2AE {
    H_P_t    header;
    D_ESAS1_TYPE00_ER_N_t    data;
} S_ESAS1_MODE_2AE_t;
typedef struct S_ESAS1_MODE_2AT {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_N_t    data;
} S_ESAS1_MODE_2AT_t;

typedef struct S_ESAS2_MODE_2AE {
    H_P_t    header;
    D_ESAS2_TYPE00_ER_N_t    data;
} S_ESAS2_MODE_2AE_t;

```

```

typedef struct S_ESAS2_MODE_2AT {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_N_t    data;
} S_ESAS2_MODE_2AT_t;

typedef struct S_IMA_MODE_2A {
    H_P_t    header;
    D_IMA_TYPE44_IS_A_t    data;
} S_IMA_MODE_2A_t;

typedef struct S_IEA_MODE_2A {
    H_P_t    header;
    D_IEA_TYPE82_IS_A_t    data;
} S_IEA_MODE_2A_t;

/*****
/*****
/* MODE 1B E1sI2s */
/*****
/*****
typedef struct S_ESAS1_MODE_1B {
    H_P_t    header;
    D_ESAS1_TYPE01_EM_H_t    data;
} S_ESAS1_MODE_1B_t;

typedef struct S_ESAS2_MODE_1B {
    H_P_t    header;
    D_ESAS2_TYPE01_EM_H_t    data;
} S_ESAS2_MODE_1B_t;

typedef struct S_IMA_MODE_1B {
    H_P_t    header;
    D_IMA_TYPE42_IM_L_t    data;
} S_IMA_MODE_1B_t;

typedef struct S_IEA_MODE_1B {
    H_P_t    header;
    D_IEA_TYPE80_IM_L_t    data;
} S_IEA_MODE_1B_t;

/*****
/*****
/* MODE 71 SV MONITOR */
/*****
/*****
typedef struct S_SVMONITOR_MODE_71 {
    H_P_t    header;
    D_TYPEF0_ESAS1_SV_MONITOR_t esas1_mon;
    D_TYPEF0_ESAS2_SV_MONITOR_t esas2_mon;
    D_TYPEF1_IMA_SV_MONITOR_t ima_mon;
    D_TYPEF1_IEA_SV_MONITOR_t iea_mon;
} S_SVMONITOR_MODE_71_t;

/*****
/*****
/* MODE 78 LMAG MONITOR */
/*****
/*****
typedef struct S_LMAGMONITOR_MODE_78 {
    H_P_t    header;
    D_TYPEF2_LMAG_MONITOR_t lmag_mon;
} S_LMAGMONITOR_MODE_78_t;

/*****
/*****
/* MODE 70 INTERNAL INFORMATION */
/*****
/*****
typedef struct S_INTINF_MODE_70 {
    H_P_t    header;

```

```
    D_TYPEF3_INTERNAL_INFORMATION_t int_inf;  
} S_INTINF_MODE_70_t;
```

添付資料-3
dump_pbf.c

```

/*
    SELENE MAP PACE PBF Format Data
    Sample read program
    Created by Yoshifumi Saito on 2007/06/30 Ver. 0.0
*/

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include <math.h>

#include "paceql_outputdata_070904.h"

/* DATA HEADER */
    H_P_t s_pace_header;
    size_t size_H_P_t=(sizeof(s_pace_header));
/* ESA DATA TYPE */
    D_ESAS1_TYPE00_t s_esas1_type00;
    size_t size_s_esas1_type00_t=(sizeof(s_esas1_type00));
    D_ESAS2_TYPE00_t s_esas2_type00;
    size_t size_s_esas2_type00_t=(sizeof(s_esas2_type00));
    D_ESAS1_TYPE01_t s_esas1_type01;
    size_t size_s_esas1_type01_t=(sizeof(s_esas1_type01));
    D_ESAS2_TYPE01_t s_esas2_type01;
    size_t size_s_esas2_type01_t=(sizeof(s_esas2_type01));
    D_ESAS1_TYPE02_t s_esas1_type02;
    size_t size_s_esas1_type02_t=(sizeof(s_esas1_type02));
    D_ESAS2_TYPE02_t s_esas2_type02;
    size_t size_s_esas2_type02_t=(sizeof(s_esas2_type02));
    D_ESAS1_TYPE03_t s_esas1_type03;
    size_t size_s_esas1_type03_t=(sizeof(s_esas1_type03));
    D_ESAS2_TYPE03_t s_esas2_type03;
    size_t size_s_esas2_type03_t=(sizeof(s_esas2_type03));
/* IMA DATA TYPE */
    D_IMA_TYPE40_t s_ima_type40;
    size_t size_s_ima_type40_t=(sizeof(s_ima_type40));
    D_IMA_TYPE41_t s_ima_type41;
    size_t size_s_ima_type41_t=(sizeof(s_ima_type41));
    D_IMA_TYPE42_t s_ima_type42;
    size_t size_s_ima_type42_t=(sizeof(s_ima_type42));
    D_IMA_TYPE43_t s_ima_type43;
    size_t size_s_ima_type43_t=(sizeof(s_ima_type43));
    D_IMA_TYPE44_t s_ima_type44;
    size_t size_s_ima_type44_t=(sizeof(s_ima_type44));
    D_IMA_TYPE45_t s_ima_type45;
    size_t size_s_ima_type45_t=(sizeof(s_ima_type45));
/* IEA DATA TYPE */
    D_IEA_TYPE80_t s_ia_type80;
    size_t size_s_ia_type80_t=(sizeof(s_ia_type80));
    D_IEA_TYPE81_t s_ia_type81;
    size_t size_s_ia_type81_t=(sizeof(s_ia_type81));
    D_IEA_TYPE82_t s_ia_type82;
    size_t size_s_ia_type82_t=(sizeof(s_ia_type82));

    FILE *fp_in;
    unsigned char c_in_head[1024];

int main(int argc, char **argv)
{
    char    fil_nme[500];

/* ***** */
    FILE *fp_d;
    char dfname[100];
    int i,j,k,l;
    long today, time0, time1;
    long tmp[12][32][16][64];
/* ***** */

    if(argc != 5){

```



```

        fprintf(stderr, "\n Usage: read_pbf input_file yyyyymmdd, hhmmss1, hhmmss2, \n");
        fputs("\n", stderr);
        exit(1);
    }

    today = atol(argv[2]);
    time0 = atol(argv[3]);
    time1 = atol(argv[4]);

    for (l = 0; l < 12; ++l)
        for (k = 0; k < 32; ++k)
            for (j = 0; j < 16; ++j)
                for (i = 0; i < 64; ++i)
                    tmp[l][k][j][i] = 0;

    strcpy(fil_nme,argv[1]);
    //sprintf(dfname, "%s_%s_%s.dat", argv[2], argv[3], argv[4]);

/* INPUT DATA FILE OPEN */
    if (NULL == (fp_in=fopen(fil_nme,"rb")))
    {
        exit(-1);
    }

/* READ FILE HEADER DESCRIPTION */
    fread(c_in_head, 1024 , 1,fp_in);

/* FILE BINARY TYPE CHECK */
#ifdef PC
    if (c_in_head[1023] != 0xEE){
        printf("FILE_TYPE ERROR \n");
        exit(1);}
#else
    if (c_in_head[1023] != 0xDD){
        printf("FILE_TYPE ERROR \n");
        exit(1);}
#endif

/* PRINT FILE HEADER INFORMATION */
    printf("%s", c_in_head);

    while(1){
/* READ DATA HEADER */
        if (0 == fread(&s_pace_header, size_H_P_t, 1,fp_in))
            break;

/* DATA MODE */
        switch (s_pace_header.type){
            case 0x00:
                printf("type:%x\n", s_pace_header.type);

                if(s_pace_header.sensor == 0){
                    if (0 == fread(&s_esas1_type00, size_s_esas1_type00_t, 1,fp_in)) {
                        printf ("FILE ERROR: No Data \n");
                        break;
                    }
                    printf ("ESA-S1 DATA MODE %X DATA TYPE 00 time=%X%X%X",
event[0]=%X total_day=%X msec_day=%X  \n",
                s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
                s_pace_header.timeL,s_esas1_type00.event[0],s_pace_header.t_date,s_pace_header.time_ms);
                }
            else if(s_pace_header.sensor == 1) {
                if (0 == fread(&s_esas2_type00, size_s_esas2_type00_t, 1,fp_in)) {
                    printf ("FILE ERROR: No Data \n");
                    break;
                }
                printf ("ESA-S2 DATA MODE %X DATA TYPE 00 time=%X%X%X",
event[0]=%X total_day=%X msec_day=%X  \n",

```

```

s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
s_pace_header.timeL,s_esas2_type00.event[0],s_pace_header.t_date,s_pace_header.time_ms);
}
break;

case 0x01:
if(s_pace_header.sensor == 0) {
if (0 == fread(&s_esas1_type01, size_s_esas1_type01_t, 1,fp_in)) {
printf ("FILE ERROR: No Data %n");
break;
}
}
printf ("ESA-S1 DATA MODE %X DATA TYPE 01 time=%X%X%X
event[0]=%X total_day=%X msec_day=%X %n",

s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
s_pace_header.timeL,s_esas1_type01.event[0],s_pace_header.t_date,s_pace_header.time_ms);
}
else if(s_pace_header.sensor == 1){
if (0 == fread(&s_esas2_type01, size_s_esas2_type01_t, 1,fp_in)){
printf ("FILE ERROR: No Data %n");
break;
}
}
printf ("ESA-S2 DATA MODE %X DATA TYPE 01 time=%X%X%X
event[0]=%X total_day=%X msec_day=%X %n",

s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
s_pace_header.timeL,s_esas2_type01.event[0],s_pace_header.t_date,s_pace_header.time_ms);
}
break;

case 0x02:
if(s_pace_header.sensor == 0){
if (0 == fread(&s_esas1_type02, size_s_esas1_type02_t, 1,fp_in)){
printf ("FILE ERROR: No Data %n");
break;
}
}
printf ("ESA-S1 DATA MODE %X DATA TYPE 02 time=%X%X%X
event[0]=%X total_day=%X msec_day=%X %n",

s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
s_pace_header.timeL,s_esas1_type02.event[0],s_pace_header.t_date,s_pace_header.time_ms);
}
else if(s_pace_header.sensor == 1){
if (0 == fread(&s_esas2_type02, size_s_esas2_type02_t, 1,fp_in)){
printf ("FILE ERROR: No Data %n");
break;
}
}
printf ("ESA-S2 DATA MODE %X DATA TYPE 02 time=%X%X%X
event[0]=%X total_day=%X msec_day=%X %n",

s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
s_pace_header.timeL,s_esas2_type02.event[0],s_pace_header.t_date,s_pace_header.time_ms);
}
break;

case 0x03:
if(s_pace_header.sensor == 0){
if (0 == fread(&s_esas1_type03, size_s_esas1_type03_t, 1,fp_in)) {
printf ("FILE ERROR: No Data %n");
break;
}
}
printf ("ESA-S1 DATA MODE %X DATA TYPE 03 time=%X%X%X
event[0]=%X total_day=%X msec_day=%X %n",

s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
s_pace_header.timeL,s_esas1_type03.event[0],s_pace_header.t_date,s_pace_header.time_ms);
}
else if(s_pace_header.sensor == 1){

```

```

        if (0 == fread(&s_esas2_type03, size_s_esas2_type03_t, 1,fp_in)){
            printf ("FILE ERROR: No Data ¥n");
            break;
        }
        printf ("ESA-S2 DATA MODE %X DATA TYPE 03 time=%X%X%X
event[0]=%X total_day=%X msec_day=%X ¥n",
        s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
        s_pace_header.timeL,s_esas2_type03.event[0],s_pace_header.t_date,s_pace_header.time_ms);
    }
    break;
    case 0x40:
        if (0 == fread(&s_ima_type40, size_s_ima_type40_t, 1,fp_in)){
            printf ("FILE ERROR: No Data ¥n");
            break;
        }
        printf ("IMA DATA MODE %X DATA TYPE 40 time=%X%X%X event[0][0]=%X
total_day=%X msec_day=%X ¥n",
        s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
        s_pace_header.timeL,s_ima_type40.event[0][0],s_pace_header.t_date,s_pace_header.time_ms);
    break;
    case 0x41:
        if (0 == fread(&s_ima_type41, size_s_ima_type41_t, 1,fp_in)){
            printf ("FILE ERROR: No Data ¥n");
            break;
        }
        printf ("IMA DATA MODE %X DATA TYPE 41 time=%X%X%X event[0][0]=%X
total_day=%X msec_day=%X ¥n",
        s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
        s_pace_header.timeL,s_ima_type41.event[0][0],s_pace_header.t_date,s_pace_header.time_ms);
    break;
    case 0x42:
        if (0 == fread(&s_ima_type42, size_s_ima_type42_t, 1,fp_in)){
            printf ("FILE ERROR: No Data ¥n");
            break;
        }
        printf ("IMA DATA MODE %X DATA TYPE 42 time=%X%X%X event[0][0]=%X
total_day=%X msec_day=%X ¥n",
        s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
        s_pace_header.timeL,s_ima_type42.event[0][0],s_pace_header.t_date,s_pace_header.time_ms);
    break;
    case 0x43:
        if (0 == fread(&s_ima_type43, size_s_ima_type43_t, 1,fp_in)){
            printf ("FILE ERROR: No Data ¥n");
            break;
        }
        printf ("IMA DATA MODE %X DATA TYPE 43 time=%X%X%X event[0][0]=%X
total_day=%X msec_day=%X ¥n",
        s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
        s_pace_header.timeL,s_ima_type43.event[0][0],s_pace_header.t_date,s_pace_header.time_ms);
    break;
    case 0x44:
        if (0 == fread(&s_ima_type44, size_s_ima_type44_t, 1,fp_in)){
            printf ("FILE ERROR: No Data ¥n");
            break;
        }
        printf ("IMA DATA MODE %X DATA TYPE 44 time=%X%X%X event[0][0]=%X
total_day=%X msec_day=%X ¥n",
        s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
        s_pace_header.timeL,s_ima_type44.event[0][0],s_pace_header.t_date,s_pace_header.time_ms);
    break;
    case 0x45:
        if (0 == fread(&s_ima_type45, size_s_ima_type45_t, 1,fp_in)){
            printf ("FILE ERROR: No Data ¥n");

```

```

        break;
    }
    printf ("IMA DATA MODE %X DATA TYPE 45 time=%X%X%X event[0][0]=%X
total_day=%X msec_day=%X  \n",
        s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
        s_pace_header.timeL,s_ima_type45.event[0][0],s_pace_header.t_date,s_pace_header.time_ms);
    break;

    case 0x80:
        if (0 == fread(&s_ia_type80, size_s_ia_type80_t, 1,fp_in)){
            printf ("FILE ERROR: No Data \n");
            break;
        }
        printf ("IEA DATA MODE %X DATA TYPE 80 time=%X%X%X event[0]=%X
total_day=%X msec_day=%X \n",
            s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
            s_pace_header.timeL,s_ia_type80.event[0],s_pace_header.t_date,s_pace_header.time_ms);
        break;

    case 0x81:
        if (0 == fread(&s_ia_type81, size_s_ia_type81_t, 1,fp_in)) {
            printf ("FILE ERROR: No Data \n");
            break;
        }

        //printf ("IEA DATA MODE %X DATA TYPE 81 time=%X%X%X event[0]=%X
total_day=%X msec_day=%X \n",
            //      s_pace_header.mode,s_pace_header.timeH,s_pace_header.timeM,
            //
            s_pace_header.timeL,s_ia_type81.event[0],s_pace_header.t_date,s_pace_header.time_ms);

        /* ダンプ */

        //if (NULL == (fp_d = fopen(dfname, "a"))) {
        //    printf (" --- OUTPUT FILE cannot be open.\n");
        //    exit(1);
        //}
        //if (s_pace_header.yyyymmdd == today &&
        //    s_pace_header.hhmmss >= 180000 &&
        //    s_pace_header.hhmmss <= 181000) {
        //    fprintf(fp_d, "yyymmdd:%ld hhmmss: %ld\n",
        //        s_pace_header.yyyymmdd, s_pace_header.hhmmss);

        //for (k = 0; k < 16; ++k)
        //    fprintf(fp_d, "%d ", s_ia_type81.event[k]);
        //fprintf(fp_d, "\n");
        //fprintf(fp_d, "%ld ", s_pace_header.hhmmss);
        //for (k = 0; k < 32; ++k)
        //    for (j = 0; j < 16; ++j)
        //        for (i = 0; i < 64; ++i) {
        //            fprintf(fp_d, "%d",
s_ia_type81.cnt[k][j][i]);
        //            tmp[0][k][j][i] +=
s_ia_type81.cnt[k][j][i];
        //        }

        //fprintf(fp_d, "\n");

        //}
        //fclose(fp_d);
        for (l = 0; l < 6; ++l)
            if (s_pace_header.yyyymmdd == today &&
                s_pace_header.hhmmss >= 180000 +l*1000 &&
                s_pace_header.hhmmss <= 180000 +(l+1)*1000)
                for (k = 0; k < 32; ++k)
                    for (j = 0; j < 16; ++j)
                        for (i = 0; i < 64; ++i)

tmp[l][k][j][i] += s_ia_type81.cnt[k][j][i];

        for (l = 0; l < 6; ++l)
            if (s_pace_header.yyyymmdd == today &&

```

```

s_pace_header.hhmmss >= 190000 + l*1000    &&
s_pace_header.hhmmss <= 190000 + (l+1)*1000)
    for (k = 0; k < 32; ++k)
        for (j = 0; j < 16; ++j)
            for (i = 0; i < 64; ++i)

tmp[l+6][k][j][i] += s_ia_type81.cnt[k][j][i];
break;

case 0x82:
    if (0 == fread(&s_ia_type82, size_s_ia_type82_t, 1, fp_in)) {
        printf("FILE ERROR: No Data %n");
        break;
    }
    printf("IEA DATA MODE %X DATA TYPE 82 time=%X%X%X event[0]=%X
total_day=%X msec_day=%X %n",
        s_pace_header.mode, s_pace_header.timeH, s_pace_header.timeM,
        s_pace_header.timeL, s_ia_type82.event[0], s_pace_header.t_date, s_pace_header.time_ms);
    break;

default:
    break;
} /* switch type */

} /* while(1) */

for (l = 0; l < 12; ++l) {
    sprintf(dfname, "%02d_disp.dat", l);
    if (NULL == (fp_d = fopen(dfname, "w"))) {
        printf(" --- OUTPUT FILE cannot be open.%n");
        exit(1);
    }
    for (k = 0; k < 32; ++k) {
        for (j = 0; j < 16; ++j)
            for (i = 0; i < 64; ++i)
                fprintf(fp_d, "%d ", tmp[l][k][j][i]);

        fprintf(fp_d, "%n");
    }
    fclose(fp_d);
}

fclose(fp_in);
exit(0);
}

```