

ハマナ1基地局サブシステム仕様書

1. 概要	1
2. LLHファイルフォーマット	1
3. EEPROM操作ソフトウェア	2
3.1. 概要	2
3.2. DFD0	2
3.3. プロセス仕様	3
3.3.1 Readプロセス	3
3.3.2 LLH data展開プロセス	4
3.3.3 ファイル保存プロセス	4
3.4. モジュール構造	5
4. 軌跡表示ソフトウェア	6
4.1. 概要	6
4.1.1 画面イメージ	6
4.1.2 計算方法	7
4.2. DFD0	7

1. 概要

ハマナ1基地局は、地上設備の一つとして、主にユーザーインターフェースと、一部発射前の探査体への指示を行います。基地局上では、下記の2つの機能が動作します。

機能名	使用場面	詳細
EEPROM操作	回収後の軌跡表示の準備	打ち上げ後回収された探査体と通信を行い、LLH情報を取り出してファイルに保存する。
軌跡の表示	計測結果を確認する時	ファイルに保存されたLLH情報を取り出し、軌跡として表示を行う。

上記の機能は、各機能に個別のGUIアプリケーションを用意して実現します。

2. LLHファイルフォーマット

軌跡表示ソフトウェアは、LLHファイルを表示のための情報として使用します。したがって、LLHファイルフォーマットは、軌跡表示ソフトウェアにとって使いやすいフォーマットであることが重要です。また、開発途中段階では、LLHファイルは、テキストで直接入力可能であり、かつ目でチェックできるフォーマットにしておいた方が、デバッグなどの用途で使いやすいと考えられます。

この両者を満たすために、LLHファイルフォーマットは、緯度、経度を浮動小数点形式で記録するという最もシンプルな形式を取ることにしました。この形であれば、特別なツールを用意しなくとも、テキストエディタで簡単に編集できますし、人間の目にも見やすく、軌跡計算においても地球の半径との関係からすぐにメートル単位での座標が計算可能となります。

1個のLLH情報は、ファイルでは1行に格納することにした。各行のフォーマットをBNFにて以下のように定義します。

```
line ::= date time latitude longitude
```

```
date ::= year¥.month¥.day
```

```
time ::= our:minute:second
```

```
latitude ::= [N | S] [0-9][0-9]¥.[0-9][0-9]¥.[0-9][0-9][0-9][0-9]
```

```
longitude ::= [E | W] [0-9][0-9]¥.[0-9][0-9]¥.[0-9][0-9][0-9][0-9]
```

```
year ::= [0-9][0-9][0-9][0-9]
```

```
month ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12
```

```
day ::= [0-3]?[0-9]
```

```
hour ::= [0-2]?[0-9]
```

```
minute ::= [0-6]?[0-9]
```

```
second ::= [0-6]?[0-9]
```

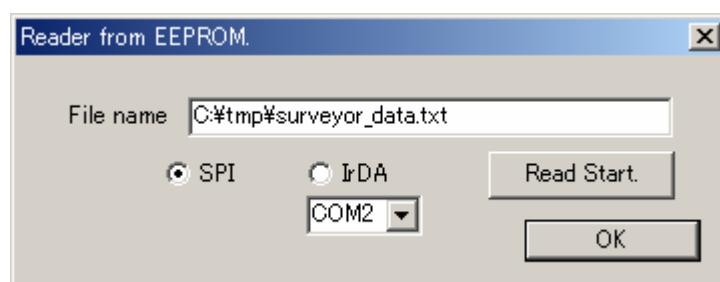
(例)

```
2004.06.08 02:15:45 N34.40.0000 E135.20.0000
```

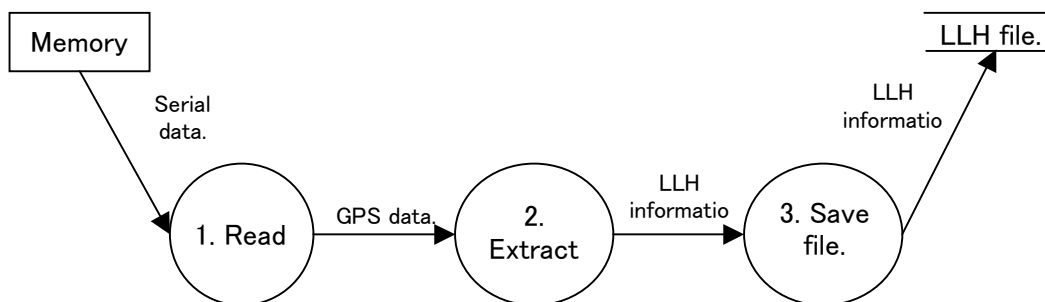
3. EEPROM操作ソフトウェア

3.1. 概要

このソフトは、探査体を回収後、EEPROMからデータを読み出すのに使用します。
SPI/IrDAのいずれを使用するか指定、IrDAを使用する場合はそのポート番号、保存するファイル名を指定して「Read Start」を実行すると、EEPROMからデータを読み込み、LLH情報のみを取り出してファイルに保存します。



3.2. DFD0



プロセス名	内容
1. Read.	Memoryに書き込まれているGPSデータをそのまま読み出す。 このプロセスが完了しても、GPSデータの前後にはSync HeaderやCRCデータが入ったままである。ただし同期チェックやCRCチェックはここで行う。
2. Extract LLH data.	LLHデータ展開プロセス。GPSデータから、時刻情報とLLHデータのみを取り出す。
3. Save file.	2. で取り出したデータをファイルに保存する。

3.3. プロセス仕様

3.3.1 Readプロセス

Readプロセスは、EEPROMに入っているデータの終端まで読み込み、メモリ上に格納します。
128byteを1フレームとして、アドレス0000hから順に読み込んで行き、データ同期が確認できる限り読み込みを続けます。

EEPROMが書き込まれているフレームのデータフォーマットは下記の通りです。

Syn. Pattern 1 byte	Syn. Pattern 1 byte	GPS Data 124 byte	Checksum 1 byte	~ CheckSum 1 byte
------------------------	------------------------	----------------------	--------------------	----------------------

0xE0	0xE0	124 Data byte	Checksum	~Checksum
------	------	---------------	----------	-----------

このフレーム構成が続く限り、読み込みを続けます。

3.3.2 LLH data展開プロセス

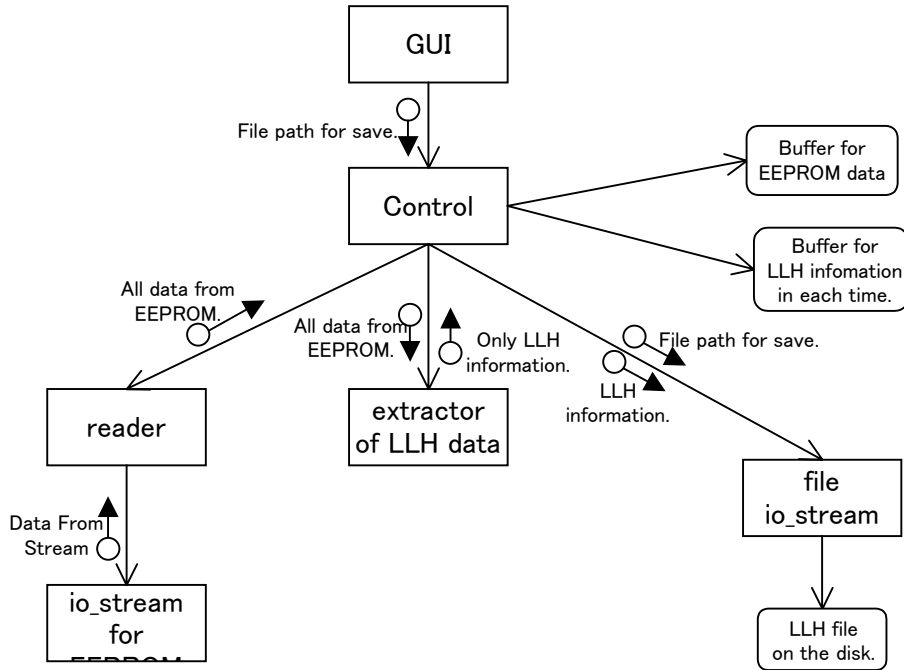
自明なので省略します。

3.3.3 ファイル保存プロセス

このプロセスは、LLHデータ展開プロセスで作成されたデータを、前章で紹介した形式でファイルに保存します。

3.4. モジュール構造

モジュール構造図は以下の通りです。



4.1.2 計算方法

垂直方法は自明なため、水平方向のみ説明します。

話を簡略化するため、経度について説明しますが、緯度についても原理は全く同じです。

まず、画面表示に使用する地図を確保したら、その中のある特定の地点を原点として定めます。

次に、その原点の緯度、経度はそれぞれ何度かを明確にします。

ここでは、その地点の経度を X_0 とします。

また、原点の画面上での座標をドット単位で明らかにします。これは X_d とします。

また、地図の縮尺から、実際の距離 L とドット数 D を明らかにしておきます。

探査体が飛行中、ある時刻で観測された経度を X とします。

すると、そのときの探査体の位置を画面上にプロットするには以下のようにします。

- (1) $X - X_0$ を計算し、経度、緯度の差分を算出します。
- (2) 地球の半径を R として、(1)で算出した角度の差分から、 X 方向の距離の差分を算出します。
具体的には、 $2 * 3.14 * R * (X - X_0) / 360$ となります。(緯度経度の表し方が 360° 形式のためです。)
- (3) 最後に、距離を画面上のドット数に変換します。
具体的には、(2) で計算した値 $* D / L$ となります。
- (4) 原点のドット座標 X_d に、(3)を足した値を表示座標とします。

4.2. DFD0

