

# 「屋外自律移動ロボットにおける GPS コンポーネント」

## ユーザーマニュアル

平成 19 年 12 月 18 日

Shibaura Institute of Technology  
Human-Robot-Interaction Lab.

## 目次

1. はじめに.....	3
2. GPS コンポーネントの機能概要.....	4
2.1 概要.....	4
2.2 ナビゲーション機能.....	4
2.3 再利用性.....	5
2.3.1 入力部.....	5
2.3.2 出力部.....	5
3. GPS コンポーネントの使用手順.....	6
3.1 GPS レシーバの設定.....	6
3.2 GPS レシーバとの接続.....	7
3.2 マップデータの作成.....	7
4. GPS コンポーネントの仕様.....	8
4.1 マップデータ仕様.....	8
4.2 プロファイル.....	8
4.3 インタフェース.....	9
5. GPS コンポーネントの使用方法.....	10
5.1 ビルド方法.....	10
5.2 実行方法.....	10
5.3 GPS コンポーネントの接続.....	11
5.4 GPS コンポーネントによるロボット制御.....	11
6. GPS コンポーネントのシステム例.....	12
6.1.1 出力表示システム.....	12
6.1.2 出力表示用コンポーネント仕様.....	13
6.1.3 GPS レシーバ通信仕様.....	14
6.2.1 ロボット実装例.....	15
6.2.2 ロボット実装例用コンポーネント仕様.....	16
6.2.3 CAN 通信インタフェース仕様.....	17
7. マップ作成手順.....	18
7.1 マップデータによるナビゲーション.....	18
7.2 「Map_Maker」機能概要.....	19
7.3 概観.....	19
7.4 使用手順.....	20
8. 動作環境.....	21
9. その他.....	21
9.1 ライセンス.....	21
9.2 ホームページ.....	21

## 1. はじめに

本書は、RT ミドルウェア上で動作する RT コンポーネント「GPS コンポーネント」に関して、その機能、使用手順、仕様等を記述するものである。

## 2. GPS コンポーネントの機能概要

本 GPS コンポーネントは, GPS を利用したナビゲーションを RT ミドルウェア上で容易に利用可能にするため RT コンポーネント化したものである. 本章では, GPS コンポーネントの機能について概説する.

### 2.1 概要

GPS コンポーネントは, GPS レシーバが受信したデータからロボットのナビゲーションのために必要な処理を行うコンポーネントである. ここでいうナビゲーションとは, スタート地点からゴールとなる目的地まで進行経路をロボットへ設定することである. そこで, GPS コンポーネントは目的地までの経路上に設定した目標点をマップデータとして利用しナビゲーション情報を出力する.

### 2.2 ナビゲーション機能

GPS コンポーネントはナビゲーション情報として現在地点から目標点への距離・旋回角度の出力を行う. 目的地までのナビゲーションは, 目標点を順次切り替えていくことで行う. ナビゲーション情報の導出には移動前の点と現在地点・目標点の 3 点の位置情報が必要である. これらの 3 点と出力される距離・旋回角度の関係は図 1 のようになる. GPS コンポーネントは, 移動前の点から 1[m]移動することにナビゲーション情報の出力を行う.

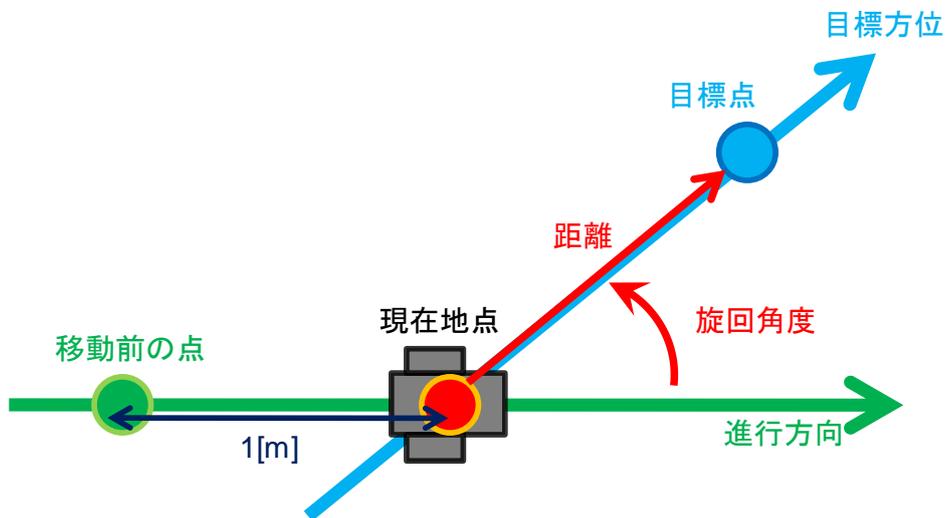


図 1 ナビゲーション概要

## 2.3 再利用性

### 2.3.1 入力部

GPS コンポーネントは直接 GPS レシーバとは接続せず，図 2 のように，GPS レシーバからデータを受信する通信コンポーネントと接続する構成となっている．これは，シリアル通信や CAN など、異なる接続方式の GPS レシーバであっても通信コンポーネントの変更で対応可能にするためである．これにより，GPS コンポーネントの再利用性を高めることができる．

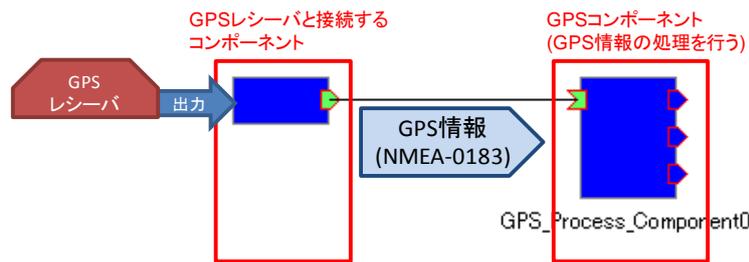


図 2 GPS コンポーネント入力部

### 2.3.2 出力部

本コンポーネントは，多くのロボットで利用できるナビゲーションとして距離・旋回角度を出力する．これらの出力を利用し、各ロボットは駆動部に固有な移動命令を生成することができる．GPS コンポーネントの出力利用のイメージを図 3 に示す．

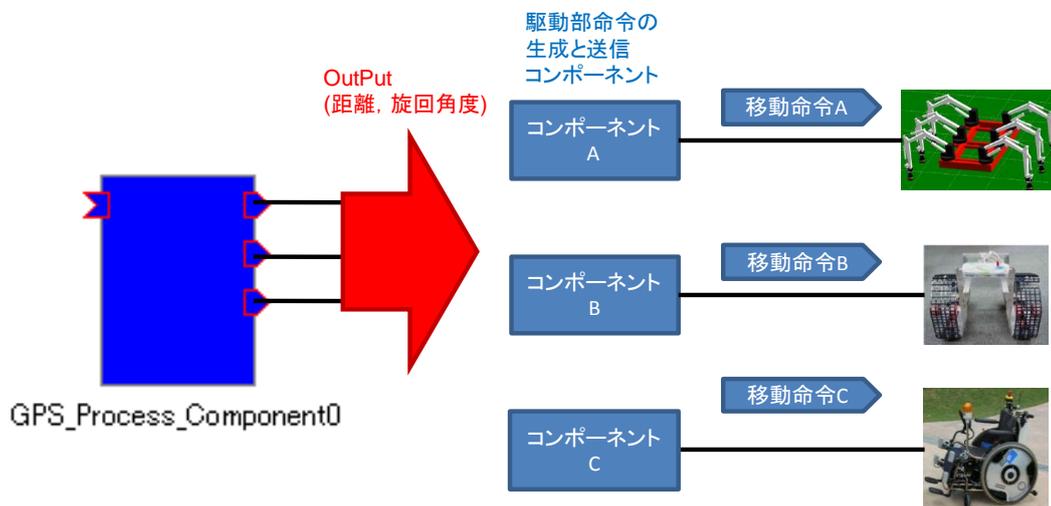


図 3 GPS コンポーネント出力の接続

### 3. GPS コンポーネントの使用手順

#### 3.1 GPS レシーバの設定

GPS コンポーネントは, 入力として NMEA-0183 規格の GPGGA センテンスを利用する. そこで, GPS レシーバを GPGGA が出力される設定に変更する必要がある(詳細は, 各 GPS レシーバのマニュアルを参照). GPGGA センテンスの例を図 4 に, データ内容を表 1 に示す.

```
$GPGGA,34557.4,3604.61439,N,14007.00612,E,2,5,1.5,23.1,M,39.1,M,7.4,129,72.32965,*79
```

図 4 GPGGA センテンス例

表 1 GPGGA センテンス, データ仕様

\$GPGGA	センテンス名
34557.4	測位時刻(UTC)
3604.61439	緯度
N	南北
14007.00612	経度
E	東西
2	受信状態
5	受信衛星数
1.5	HDOP(誤差の指標)
23.1, M	平均海水面からのアンテナ高度(m)
39.1, M	WGS-楕円体から平均海水面の高度差(m)
7.4	DGPS データのエージ(秒)
129	DGPS 基準局の ID
*79	チェックサム

### 3.2 GPS レシーバとの接続

再利用性の観点から、GPS コンポーネントは GPS レシーバと直接接続する機能をもっていない。そのため GPS レシーバと GPS コンポーネントを接続するための通信コンポーネントが必要になる。通信コンポーネントは、GPS レシーバからの情報を表 2 の仕様に従って出力するものとする。

表 2 通信コンポーネント，出力ポート仕様

ポート名称	データ型	説明
OutPort	TimedCharSeq	GPS から取得したデータを出力する。 出力するデータの規格は NMEA-0183 である。

### 3.3 マップデータの作成

GPS コンポーネントでは、通過する目標点の緯度経度・目標点を切り替える範囲を記述したマップデータが必要である。マップデータの詳細については本マニュアル 7 章「マップの作成」を参照。使用するマップデータは GPS コンポーネントの実行ファイルと同じフォルダに移動しておく。

## 4. GPS コンポーネントの仕様

### 4.1 マップデータ

GPS コンポーネントが利用するマップデータの仕様を表 3 に示す。これらのデータが記録されたテキストファイルをマップデータとして利用する。

表 3 GPS コンポーネント, マップデータ仕様

データ名称	説明
緯度, 南北	通過する緯度を記述する。南北は, 北半球であれば「N」, 南半球であれば「S」とする
経度, 東西	通過する経度を記述する。東西は, 東半球であれば「E」, 西半球であれば「W」とする
判定範囲	記述した緯度, 経度に何[m]近づいたら次の目標点に切り替えるか, を設定する

### 4.2 プロファイル

GPS コンポーネントのプロファイルを表 4 に示す。

表 4 GPS コンポーネント, プロファイル

プロファイル項目	値
モジュール名 (RTC_MODULE_NAME)	GPS_Process_Component
概要 (RTC_MODULE_DESC)	GPS_Component
バージョン (RTC_MODULE_VERSION)	1.0.0
作成者 (RTC_MODULE_VENDOR)	sato&tanaka
カテゴリ (RTC_MODULE_CATEGORY)	GPS
コンポーネント型 (RTC_MODULE_COMP_TYPE)	SPORADIC
アクティビティ型 (RTC_MODULE_ACT_TYPE)	DataFlowComponent
最大インスタンス数 (RTC_MODULE_MAX_INST)	10
モジュール記述言語名 (RTC_MODULE_LAN)	C++(Windows)

### 4.3 インタフェース

GPS コンポーネントのインタフェースを以下の表に示す。また、GPS コンポーネントの概要を図5に示す。

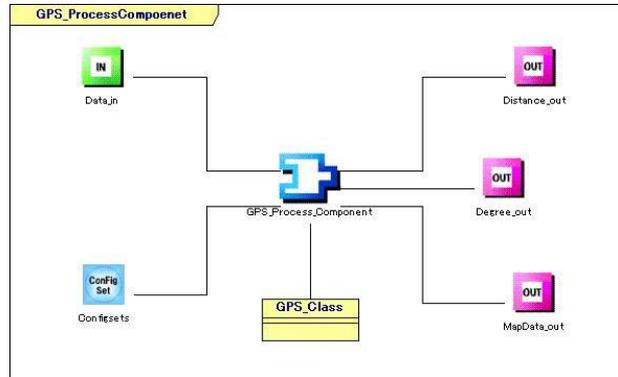


図5 GPS コンポーネント概要

表5 GPS コンポーネント，入力ポート(InPort)仕様

ポート名称	データ型	説明
Data_in	TimedCharSeq	GPS より受信したデータを入力する。 入力されるデータの規格は NMEA0183 である。

表6 GPS コンポーネント，出力ポート(OutPort)仕様

ポート名称	データ型	説明	備考
Distance_out	TimedDouble	入力されたデータとマップデータより目的点までの距離を計算し出力する。	単位[m]
Degree_out	TimedDouble	入力されたデータとマップデータより目的点までの角度を計算し出力する。	単位[degree] レンジ 0° ~ 359°
MapData_out	TimedShort	現在の目標点としているマップの番号を出力する。最後の目標点に達した場合は-1を出力する。	

表7 GPS コンポーネント，コンフィギュレーションインタフェース仕様

名称	データ型	初期値	説明
str_mapname	string	map.txt	ロードするマップデータを指定する

## 5. GPS コンポーネントの使用方法

本章では、GPS コンポーネントの使用方法を説明する。本コンポーネントは Windows 環境で開発した。そこで、本章では Windows 上で Visual C++ 8.0 を用いたビルド方法を説明する。

### 5.1 ビルド方法

GPS コンポーネントのプロジェクトファイルとソースファイルをビルドする手順を以下に示す。

(1) 以下のファイルを同一フォルダ内に移動する

- GPS\_Process\_Component.vcproj
- OpenRTM-aist.vsprops
- GPS\_Process\_Component.cpp
- GPS\_Process\_Component.h
- GPS\_Process\_ComponentComp.cpp
- GPS\_Class.cpp
- GPS\_Class.h

(2) GPS\_Process\_Component.vcproj をダブルクリックすることで、Visual C++のプロジェクトが開かれる。ここで、本コンポーネントの開発は OpenRTM-aist0.4.0 で行ったので、0.4.1 の環境で使用する場合は、依存ファイルを 0.4.1 に変更する必要がある。

(3) ツールバーからビルド→プロジェクトのビルドを選択しビルドを実行する。

(4) 上記フォルダ内の Debug フォルダに実行ファイル「GPS\_Process\_Component.exe」が生成される。

### 5.2 実行方法

(1) RT コンポーネント設定ファイル rtc.conf の設定を行う。設定項目 corba.nameservers で GPS コンポーネントを登録するネームサーバを指定する。このファイルを実行ファイルと同一のフォルダに移動しておく。

(2) マップデータを実行ファイルと同一のフォルダに移動する。

(3) 実行ファイルから GPS コンポーネントを起動する。

### 5.3 GPS コンポーネントの接続

使用するマップデータの名前をコンフィギュレーションで指定する。その後、Apply をクリックして値を適用する。GPS コンポーネントを **Activate** するとマップデータが読み込まれナビゲーションが開始される。

### 5.4 GPS コンポーネントによるロボット制御

GPS コンポーネントの利用例として、GPS コンポーネントによるナビゲーションのみでロボットが移動する場合を考える。ロボットは 1[m]の直進を行うごとに出力されるナビゲーションに応じて旋回するよう設定する。この繰り返しによって目的地までの移動が可能になる。

**注意：**GPS コンポーネントは GPGGA センテンスを利用しているため、GPS レシーバの設定で GPGGA センテンスを出力するように設定する。

GPS コンポーネントの **Activate** 直後には移動距離が 0 となり、進行方向が算出できないことから旋回角度は 0 を出力する。

## 6. GPS コンポーネントのシステム例

### 6.1.1 出力表示システム

距離・角度・マップ番号の各出力を受け取りコンソール画面に表示するコンポーネントを GPS コンポーネントの検証のために作成した。また、シリアル通信により GPS レシーバと接続する通信コンポーネントを作成した。これらのコンポーネントを利用して構成したシステムと動作を図 6, 図 7 に示す。

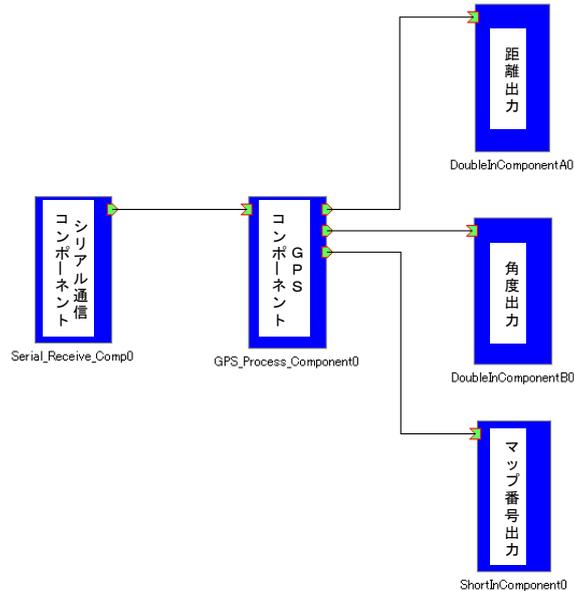


図 6 出力表示システム構成



図 7 GPS コンポーネント動作例

## 6.1.2 出力表示システム用コンポーネント仕様

出力表示システムにて利用したコンポーネントの仕様を以下の表に示す。

### (1)Serial\_Receive\_Comp コンポーネント

表 8 出力ポート(OutPort)仕様

ポート名称	データ型	説明
OutPort	TimedCharSeq	シリアル通信により受信したデータを出力する。

表 9 コンフィギュレーションインタフェース仕様

名称	データ型	初期値	説明
int_baudrate	int	19200	シリアル通信のボーレートを設定する。
str_portname	string	COM1	使用する COM ポートを設定する

### (2)DoubleIn コンポーネント

表 10 入力ポート(InPort)仕様

ポート名称	データ型	説明
InPort	TimedDouble	TimedDouble 型のデータを入力する。 入力はコンソール画面に出力する。

### (3)ShortIn コンポーネント

表 11 入力ポート(InPort)仕様

ポート名称	データ型	説明
InPort	TimedShort	TimedShort 型のデータを入力する。 入力はコンソール画面に出力する。

### 6.1.3 GPS レシーバ通信仕様

検証に用いた GPS レシーバ Crescent A100 Smart Antenna(Hemisphere)の通信仕様を表 12 に示す.

表 12 GPS レシーバ通信仕様

項目	説明
PC インタフェース	シリアルポート,全二重 RS232S 2ポート
ボーレート	4800-57600
データビット	8
パリティ	None
ストップビット	1
出力仕様	NMEA0183
出力頻度	1-10[Hz]

## 6.2.1 ロボット実装例

本研究室ではリアルワールドロボットチャレンジ(RWRC：つくばチャレンジ，財団法人ニューテクノロジー振興財団主催)での走行を目的とした屋外走行用ロボット PAR-NE07を開発している。PAR-NE07の外観を図8に示す。そこで，GPSコンポーネントの検証としてPAR-NE07への実装を行った。PAR-NE07は，体内ネットワークCANによりH8Sマイコンとの通信を行う分散制御を採用している。そこでCAN ControllerコンポーネントとCompose Messageコンポーネント(\*)を用いてナビゲーションシステムを構築した。このシステムはCANバスを通じて駆動部のH8Sマイコンへ移動命令を送信する。本実装におけるシステム構成を図9に示す。Compose Messageコンポーネントは入力されたデータからIDの付与やメッセージの生成などを行うコンポーネントである。CAN ControllerコンポーネントはCompose Messageコンポーネントからのデータを入力とし，対応するCANバスへ送信するコンポーネントである。



図8 PAR-NE07

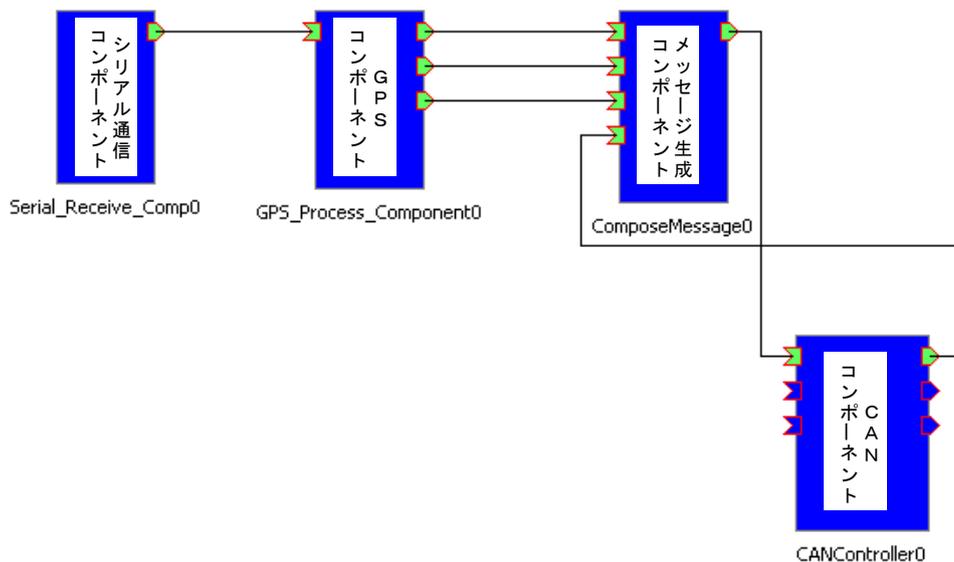


図9 ロボット実装例におけるシステム構成

(\*)参考文献 三浦俊宏他：分散制御ロボットにおけるCANコンポーネント，第8回計測自動制御学会システムインテグレーション部門 RTミドルウェアコンテスト

## 6.2.2 実装検証用コンポーネント

実装による検証で用いたコンポーネントを以下の表に示す。ただし、シリアル通信コンポーネント及び GPS レシーバの仕様は、6.1 節出力表示システムと同様なので省略する。

### (1)CAN Controller コンポーネント

表 13 入力ポート(InPort)仕様

ポート名称	データ型	説明
TxChannel1	TimedOctetSeq	入力されたデータを CAN バスのチャンネル 1 に送信する。
TxChannel2	TimedOctetSeq	入力されたデータを CAN バスのチャンネル 2 に送信する。
TxChannelAll	TimedOctetSeq	入力されたデータを CAN バスのチャンネル 1 と 2 に送信する。

表 14 入力ポート(InPort)仕様

ポート名称	データ型	説明
RxChannel1	TimedOctetSeq	CAN バスのチャンネル 1 で受信したデータを出力する。
RxChannel2	TimedOctetSeq	CAN バスのチャンネル 2 で受信したデータを出力する。
RxChannelAll	TimedOctetSeq	CAN バスのチャンネル 1 及び 2 で受信したデータを出力する。

## (2)ComposeMessage コンポーネント

表 15 入力ポート(InPort)仕様

ポート名称	データ型	説明
Distance_in	TimedDouble	GPS コンポーネントの距離データを入力とする.
Degree_in	TimedDouble	GPS コンポーネントの角度データを入力とする.
MapData_in	TimedShort	GPS コンポーネントのマップ番号データを入力とする
State_in	TimedShort	駆動部の状態(待機, 移動中, 移動完了)を入力とする.

表 16 出力ポート(OutPort)仕様

ポート名称	データ型	説明
TxMessage	TimedOctetSeq	CAN バスへ送信するデータを出力する.

### 6.2.3 CAN インタフェース通信仕様

CAN インタフェース CANcardXL(Vector Japan)の通信仕様を表に示す.

表 17 CANcardXL 通信仕様

項目	説明
PC インタフェース	PCMCIA Type II - 16 ビット
最大ボーレート	1[Mbps]
CAN コントローラ	Philips 社 SJA1000
CAN コントローラの数	2
タイムスタンプ精度	1[ $\mu$ s]

## 7. マップ作成手順

本章では「Map\_maker」(以下本ソフト)を用いたマップ作成手順について説明する。本ソフトは本研究室ホームページ(<http://www.hri.ee.shibaura-it.ac.jp/>)にてダウンロード可能にする予定である。

**注意：** インタフェースがシリアル通信でない GPS レシーバでは「Map\_maker」を利用できない。その場合はマップデータの仕様や例を参考にテキストエディタで作成すること

### 7.1 マップデータによるナビゲーション

GPS コンポーネントで利用するマップデータは、目標点の点列で構成される。また、目標点は緯度・経度による位置情報と目標点を切り替える距離によって記述される。マップデータから想定されるナビゲーションのイメージを図 10 に示す。

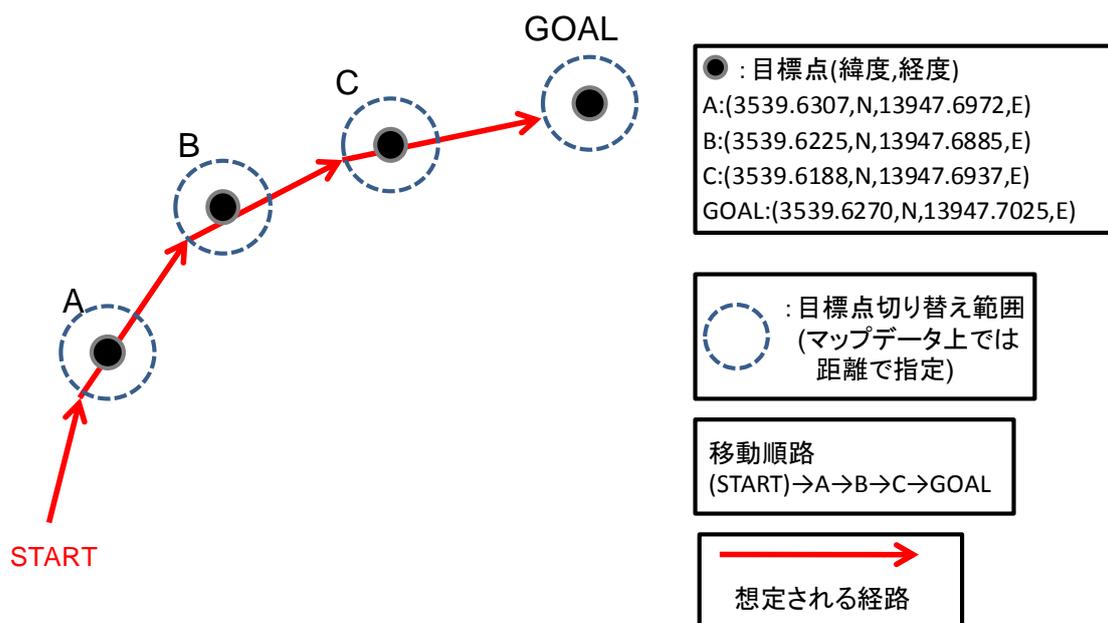


図 10 マップイメージ

## 7.2 「Map\_maker」機能

本ソフトはシリアル通信により GPS レシーバと接続し、現在位置の緯度経度を取得する。そして、GPS コンポーネントで利用できるマップデータとして保存するアプリケーションである。目標点を切り替える距離は目標点ごとに設定可能である。

## 7.3 概観

本ソフトの各表示の説明を図 11 に示す。

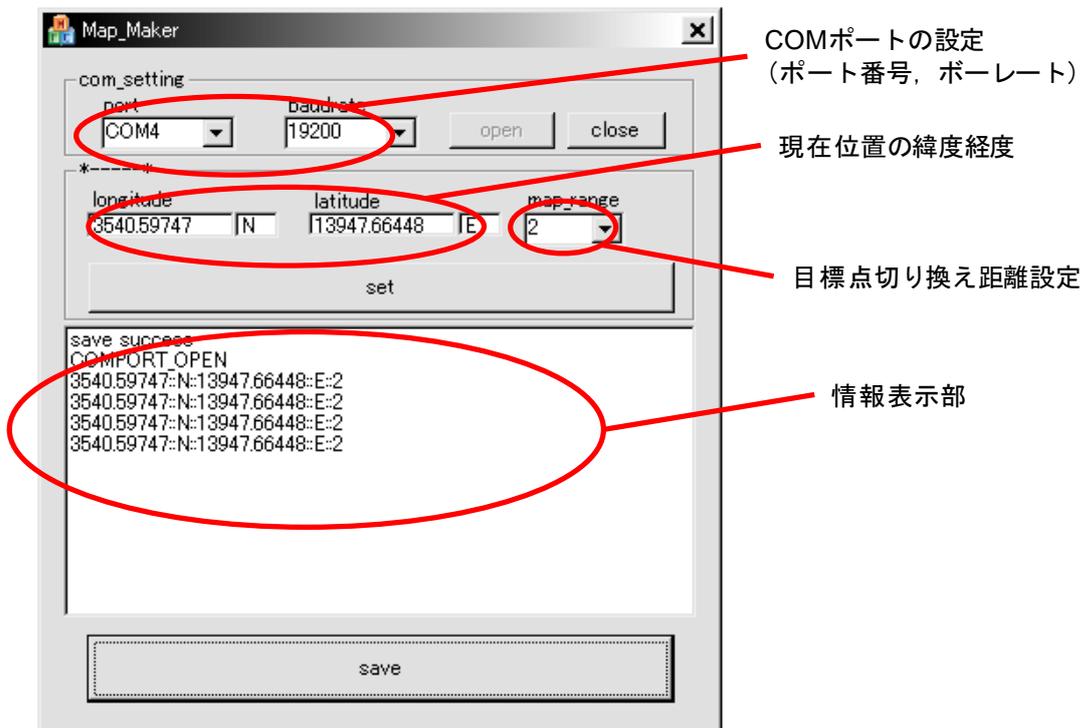


図 11 Map\_maker 概観

## 7.4 使用手順

### (1)GPS レシーバの設定

GPS コンポーネント利用時と同様, GPS レシーバが GPGGA センテンスを出力する設定に変更する.

### (2)GPS レシーバとの接続

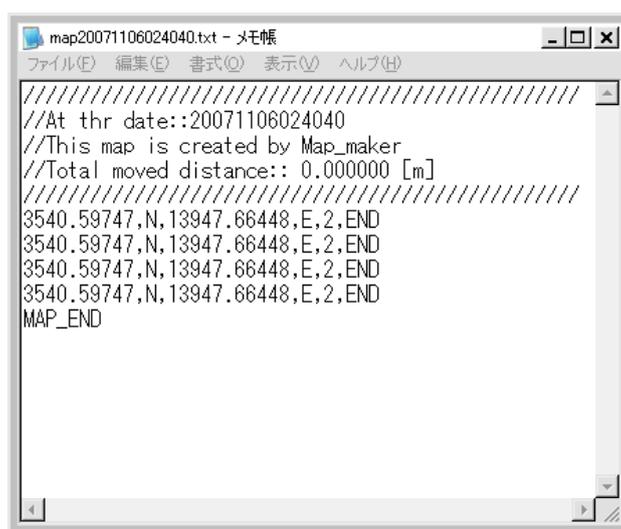
GPS レシーバを接続している COM ポートとボーレートをコンボボックスから指定し, 「open」 ボタンを押して接続する. COM ポートの接続に成功すると情報表示部に「COMPORT\_OPEN」と表示される. 「FAILED」の表示が出た場合は接続に失敗していることを示す. その場合はポート番号やボーレートを確認し再度設定する. GPS レシーバから正常にデータを受信できていれば, 現在位置の緯度・経度が表示される.

### (3)目標点の取得

始めに, 目標点に設定したい位置まで移動する. 次に「マップ切り替え距離設定」のコンボボックスでマップデータを切り替える距離を設定する. 「set」ボタンをクリックすると, それぞれのデータが情報表示部に表示される. これを繰り返し, 最終的な目的地までの経路の設定を行う.

### (4)マップデータの記録

「save」ボタンを押すと, 設定した目標点を記録したマップデータが作成される. ファイル名は「map 作成日時.txt」となる(例: map20071106015536.txt). その後, 「close」ボタンをクリックして COM ポートを閉じ, 本ソフトを終了させる. 作成されるマップデータの例を図 12 に示す.



```
map20071106024040.txt - メモ帳
ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)
//////////////////////////////////////////////////////////////////
//At thr date::20071106024040
//This map is created by Map_maker
//Total moved distance:: 0.000000 [m]
//////////////////////////////////////////////////////////////////
3540.59747,N,13947.66448,E,2,END
3540.59747,N,13947.66448,E,2,END
3540.59747,N,13947.66448,E,2,END
3540.59747,N,13947.66448,E,2,END
MAP_END
```

図 12 マップデータ例

## 8. 動作環境

### 動作対象環境

- ・ Windows 系 OS
- ・ RT ミドルウェア 「OpenRTM-aist-0.4.0」

### 動作確認済み環境

- ・ WindowsXP
- ・ RT ミドルウェア 「OpenRTM-aist-0.4.0」

### 開発環境

- ・ WindowsXP
- ・ Microsoft Visual Studio 2005(Visual C++でコンパイル)
- ・ RT ミドルウェア 「OpenRTM-aist-0.4.0」

## 9. その他

### 9.1 ライセンス

GPS コンポーネントに関する著作権は、芝浦工業大学ヒューマン・ロボット・インタラクション研究室に帰属する。

### 9.2 連絡先

芝浦工業大学 ヒューマン・ロボット・インタラクション研究室

E-mail: [hri\\_goiken@freeml.com](mailto:hri_goiken@freeml.com)

ホームページ

URL:<http://www.hri.ee.shibaura-it.ac.jp/>