

次世代ロボット知能化技術開発プロジェクト
ロボット知能ソフトウェア再利用性向上技術の開発

来訪者受付システム
(RS002)

RH 詳細設計書

0.1 版

2011 年 3 月 23 日

RTC 再利用技術研究センター

目次

1. はじめに.....	1
1.1. 目的.....	1
1.2. 本書での書式.....	1
1.3. 用語の定義、略語.....	1
1.4. 参考資料.....	1
2. システム構成.....	2
2.1.1. a. システム構成.....	2
2.1.2. b. モジュール構成.....	3
2.1.2.1. 1.全体.....	3
2.1.2.2. 2.移動(台車).....	5
2.1.2.3. 3.作業(アーム).....	5
2.1.3. c. フォルダ構成.....	6
3. 動作シーケンス.....	8
4. モジュール仕様.....	14
4.1. RTC 仕様.....	14
4.1.1. RHCtrl.....	14
4.1.2. BatteryWatcher.....	17
4.1.3. Management Of Area Map.....	18
4.1.4. HazardMesh.....	25
4.1.5. Dijkstra.....	28
4.1.6. PathCnv.....	31
4.1.7. Navigation.....	32
4.1.8. PathFollower.....	34
4.1.9. SwitchInputRTC.....	36
4.1.10. RefHardRh2.....	37
4.1.11. Odometry.....	38
4.1.12. LocalizeCenter.....	39
4.1.13. CameraEye.....	40
4.1.14. CeilingNavigation.....	41
4.1.15. GamePad.....	44
4.1.16. DispPosition.....	46
4.1.17. RH_RMRC.....	47
4.1.18. RH_HandWrapper.....	50
4.1.19. RH2ArmControl.....	54
4.1.20. tempmatch.....	55
4.1.21. pyOpenJTalk.....	56
4.1.22. PulseAudioOutput.....	57
4.1.23. SoundPlayer.....	58
4.2. スクリプト仕様.....	60
4.2.1. シナリオ系.....	61
4.2.2. RTC/RTS 制御系.....	61
4.2.3. その他.....	65
5. 画面仕様.....	66
6. データ仕様.....	67
7. その他.....	68
7.1. 延期要求.....	68
7.2. その他の要件.....	68
7.3. 特記事項.....	68

1. はじめに

1.1. 目的

本書は、「次世代ロボット知能化技術開発プロジェクト」の「ロボット知能ソフトウェア再利用性向上技術の開発」における、来訪者受付システムの関するRHに関するシステムや機能など設計に関連した内容を記載した文書である。

1.2. 本書での書式

本文書で使用している記号・書式の目的を下表に示す。

表.書式一覧

No.	記号・書式	目的
1	※	注意書き
2	赤色の文字	注記

1.3. 用語の定義、略語

表.用語の定義、略語一覧

No.	表記	意味
1	本システム	来訪者受付システム
2	プロジェクト	次世代ロボット知能化技術開発プロジェクト
3	センター	RTC再利用技術研究センター
4	現時点	本書作成時点(2010/03/01)
5	在籍者	センター内勤務者
6	OS	動作対象プラットフォーム
7	RTミドルウェア	OpenRTM-Aist
8	RTM	RTミドルウェア
9	OSS	オープンソースソフトウェア
10	障害物	人及び、人が一人で運ぶ事の出来る物体

1.4. 参考資料

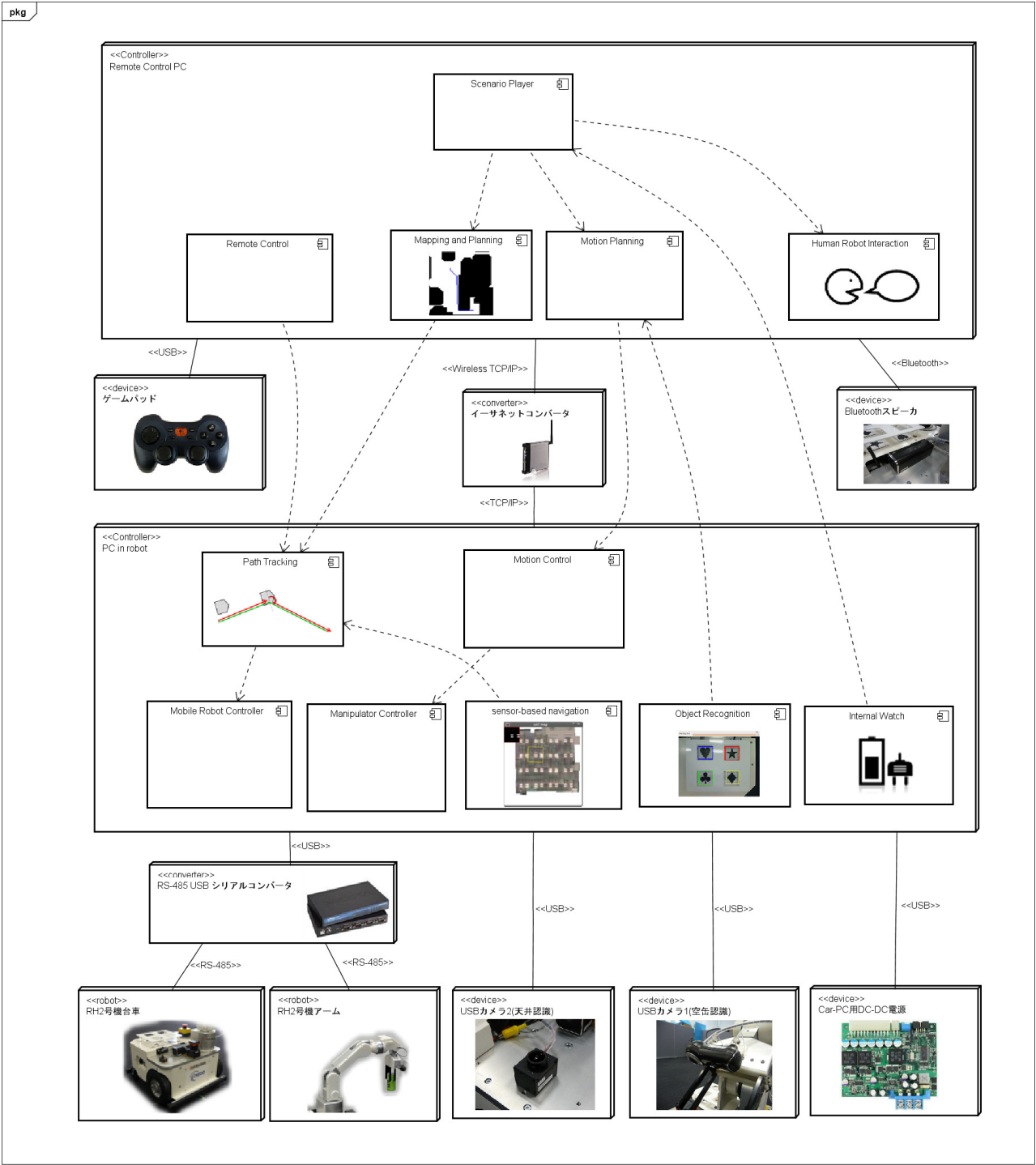
本書を作成するにあたり参照した文書・資料を下表に示す。

表.参考資料一覧

No.	文書名	備考 / URL
1	OpenRTM-aist Official Website	http://www.is.aist.go.jp/rt/OpenRTM-aist/

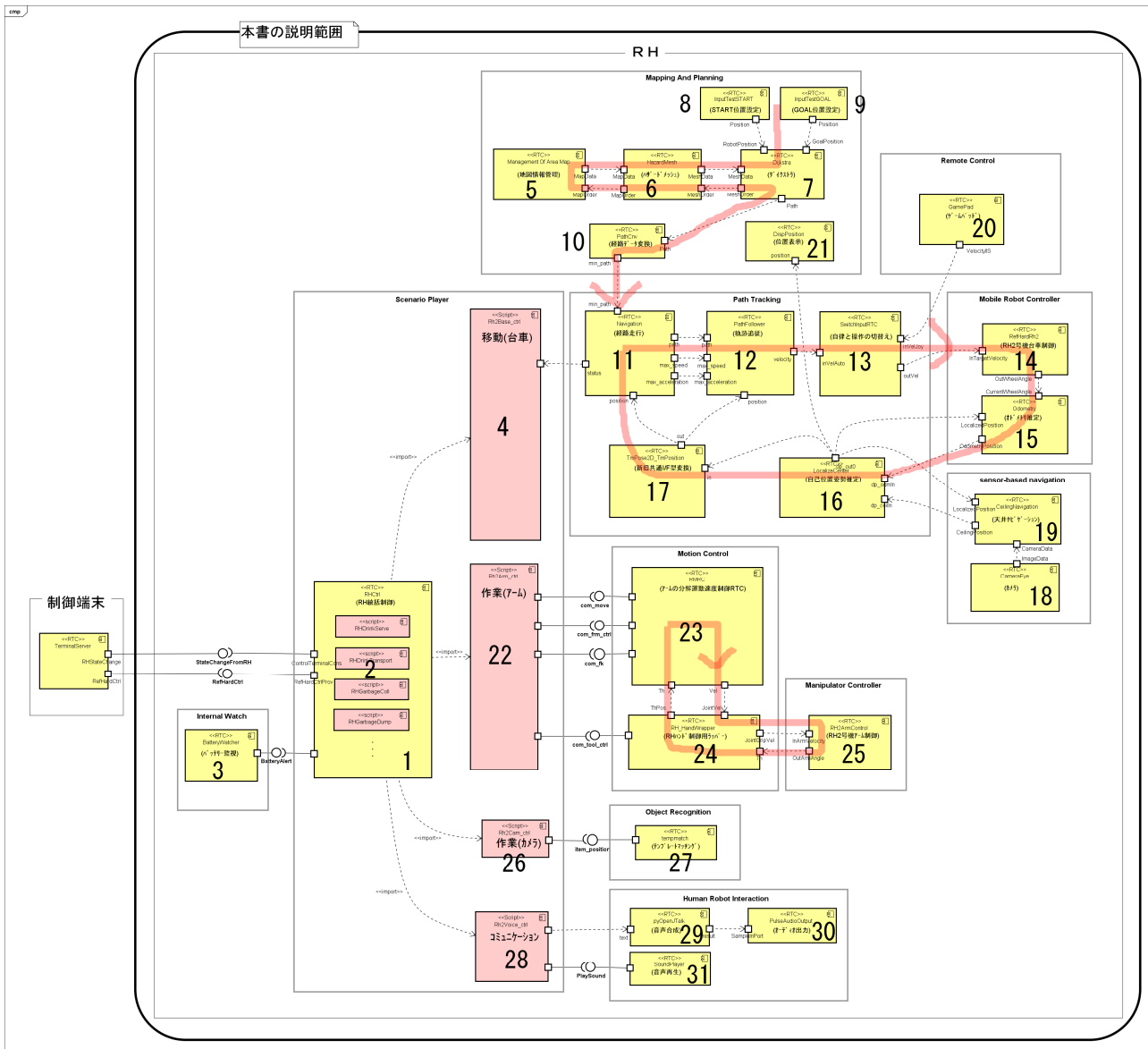
2. システム構成

2.1.1. a. システム構成



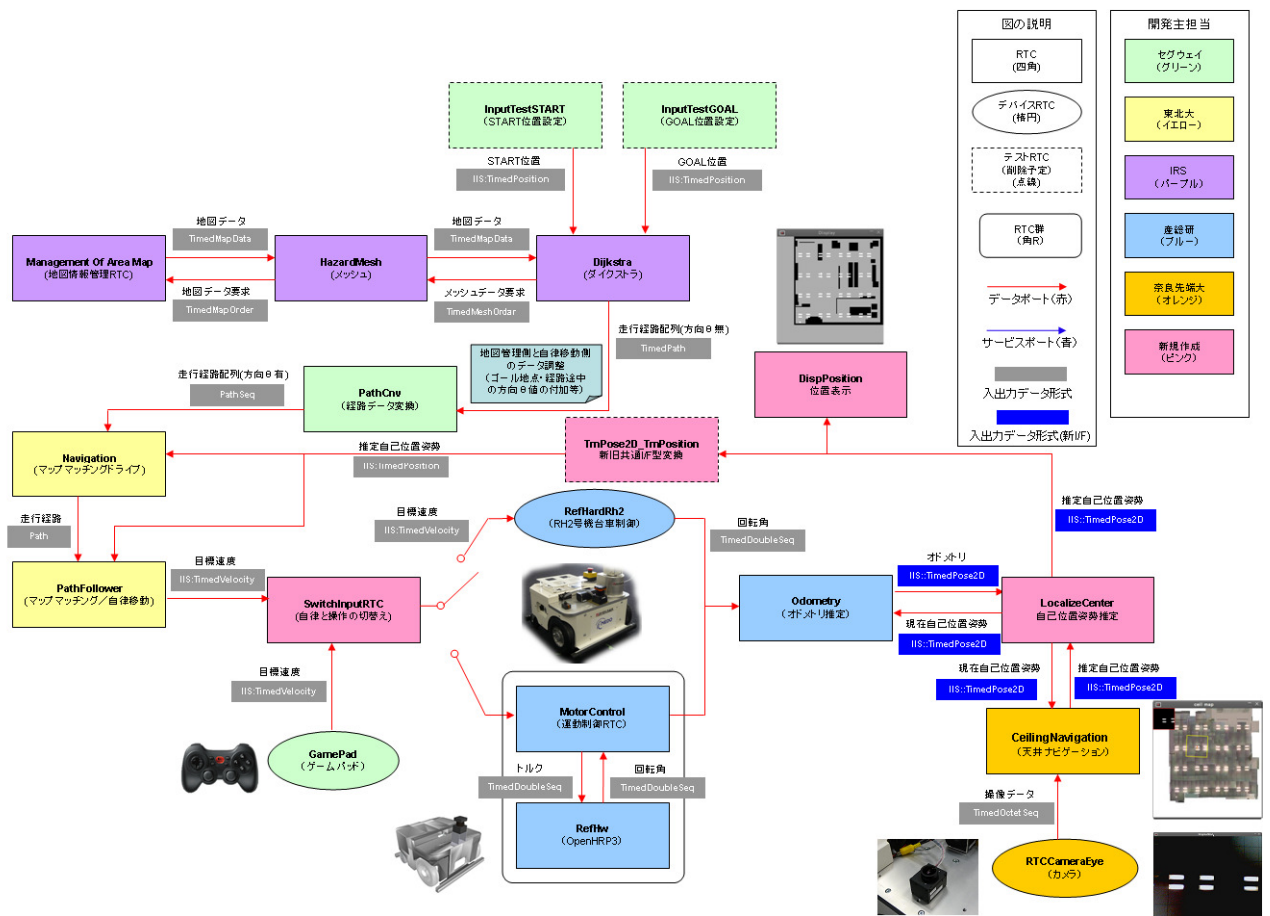
2.1.2. b. モジュール構成

2.1.2.1. 1.全体

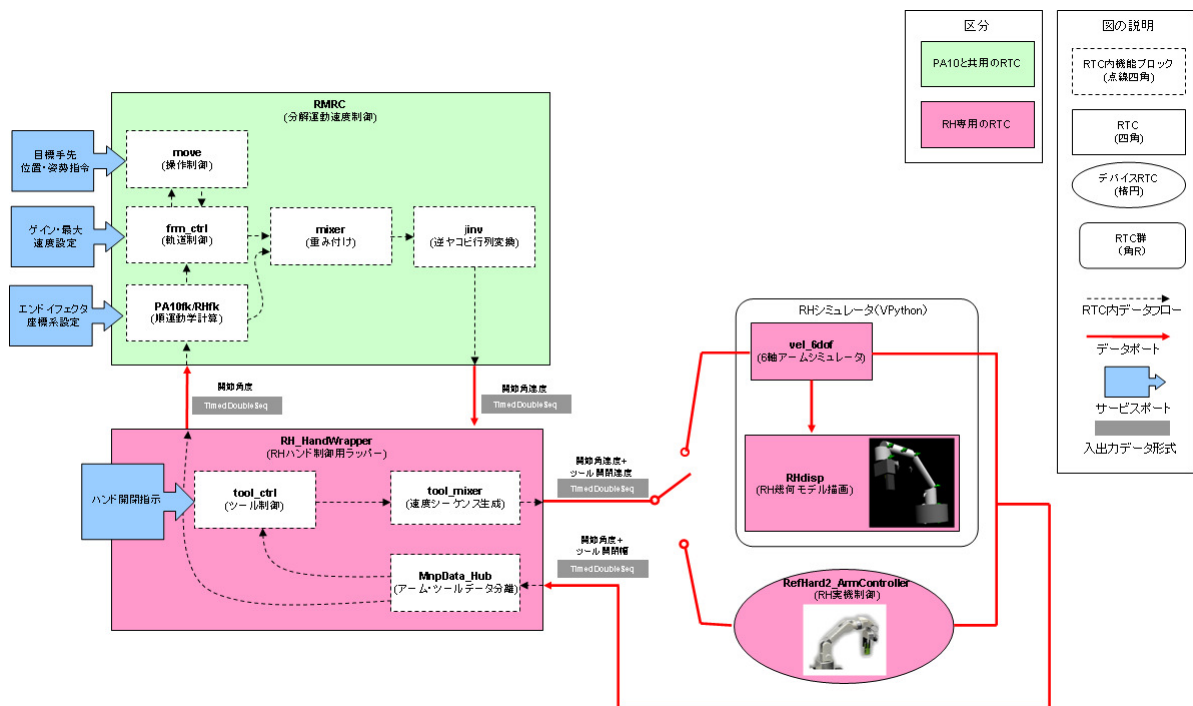


NO.	モジュール名	言語	種別	機能	概要
1	RHCtrl	C++	RTC	RH管理	RH統括制御
2	RHDrinkSeve、RHDrinkTransport、RHGarbageColl、RHGarbageDumpr・・・	Python	script		RHシナリオ実行スクリプト
3	BatteryWatcher	C++	RTC		バッテリー監視
4	Rh2Base_ctrl	Python	script	RH移動	RH台車制御
5	Management Of Area Map	C++	RTC		地図情報管理
6	HazardMesh	C++	RTC		ハザードメッシュ
7	Dijkstra	C++	RTC		ダイクストラ
8	InputTestSTART	C++	RTC		START位置設定
9	InputTestGOAL	C++	RTC		GOAL位置設定
10	PathCnv	C++	RTC		経路データ変換
11	Navigation	C++	RTC		経路走行
12	PathFollower	C++	RTC		軌跡追従
13	SwitchInputRTC	C++	RTC		自律と操作の切り替え
14	RefHardRh2	C++	RTC		RH2号機台車制御
15	Odometry	C++	RTC		オドメトリ推定
16	LocalizeCenter	C++	RTC		自己位置姿勢推定
17	TmPose2D_TmPosition	C++	RTC		新旧共通I/F型変換
18	CameraEya	C++	RTC		カメラ
19	CeilingNavigation	C++	RTC		天井ナビゲーション
20	GamePad	C++	RTC		ゲームパッド
21	DispPosition	C++	RTC		位置表示
22	Rh2Arm_ctrl	Python	script	RH作業(アーム)	RHアーム制御
23	RMRC	C++	RTC		アームの分解運動速度制御RTC
24	RH_HandWrapper	C++	RTC		RHハンド制御用ラッパー
25	RH2ArmControl	C++	RTC	RH作業(カメラ)	RH2号機アーム制御
26	Rh2Cam_ctrl	Python	script		RHカメラ制御
27	tempmatch	C++	RTC	コミュニケーション	テンプレートマッチング
28	Rh2Voice_ctrl	Python	script		コミュニケーション
29	pyOpenJTalk	C++	RTC		音声合成
30	PulseAudioOutput	C++	RTC		オーディオ出力
31	SoundPlayer	C++	RTC		音声再生

2.1.2.2. 2.移動(台車)



2.1.2.3. 3.作業(アーム)



2.1.3. c. フォルダ構成

端末毎の配布パッケージのフォルダ構成は以下の通りとする。

・RH 内蔵 PC

ディレクトリ	内容	備考
PCinRH	RH内蔵PC	
├─bin	実行ファイル	
│├─comp	バイナリ(Comp)	
│├─script	スクリプト	
││├─control	シナリオスクリプト	
│││├─Rh2Arm	作業系(アーム) RTC制御スクリプト	
│││├─Rh2Base	移動系(台車) RTC制御スクリプト	
│││├─Rh2Cam	作業系(カメラ) RTC制御スクリプト	
│││├─Rh2Ctrl	RH統括RTC制御スクリプト	
│││├─Rh2Voice	コミュニケーション系RTC制御スクリプト	
││└─openh3	OpenHRP3シミュレーション用動作スクリプト	
└─so	バイナリ(so)	
├─etc	設定ファイル	
│├─conf	rtc.conf	
│├─data	データファイル	
││├─CeilingMap	RTC再利用技術研究センターの天井地図データ	
││├─Model	OpenHRP3シミュレーション用モデル	
│││├─EnvModels	各種環境モデル	
││││├─etc	床・家具等のモデル	
││││├─osl-model	産総研OSL棟のモデル	
││││├─rtc-center	RTC再利用技術研究センターのモデル	
│││└─Robots	各種ロボットモデル	
││││├─RH0	リファレンスハードウェア0号機モデル	
││││├─RH1	リファレンスハードウェア1号機モデル	
││││└─RH2	リファレンスハードウェア2号機モデル	
││└─Project	OpenHRP3プロジェクトファイル	
└─env	pythonスクリプト用環境変数設定ファイル	
├─idl	IDLファイル	
├─include	共通ヘッダファイル	
├─lib	共通ライブラリ	
└─src	ソースファイル	
│├─DataConverter	新旧共通I/F型変換	
││├─TimedPosition2TimedPose2D	TimedPosition→TimedPose2D変換RTC	未使用
││├─TimedVelocity2TimedVelocity2D	TimedVelocity→TimedVelocity2D変換RTC	未使用
││├─TimedPose2D_TimedPosition	TimedPose2D→TimedPosition変換RTC	
││└─Tmvel2D2Tmvel	TimedVelocity2D→TimedVelocity変換RTC	未使用
├─InternalWatch	内部監視	
│├─BatteryWatcher	バッテリー監視RTC	
│└─m4api	Mini-BOX M4-ATX用API	
├─ManipulatorController	アーム制御	
│└─RefHard2_ArmController_ver3.0.0	RH2号機アームコントロールRTC	
├─MobileRobotController	台車制御	
│├─Bridge	OpenHRP3用コントローラブリッジ	シミュレータ動作
│├─InitOdometry	オドメトリ初期値設定RTC	
│├─MotorControl	モータ制御RTC	
│├─Odometry	オドメトリRTC	
│└─RefHard2	RH2号機台車制御	
├─MotionControl	動作制御	
│├─RHdisp	幾何モデル描画RTC	
│├─RH_HandWrapper	RHハンド制御用ラッパー	シミュレータ動作
│├─RMRC	アームの分解運動制御RTC	
│└─vel_sim	シミュレータRTC	シミュレータ動作
├─ObjectRecognition	物体認識	
│└─tempmatch	テンプレートマッチングRTC	
├─PathTracking	経路追従	
│├─BackInput	後進指示入力RTC(試作版)	未使用
│├─LocalizeCenter	推定自己位置姿勢融合RTC	
│├─Navigation_0.1	経路走行RTC	※
│├─PathFollower_0.1	軌跡追従RTC	※
│└─SwitchJoyStickRTC	自律と操作の切替えRTC	
├─Sensor-BasedNavigation	外界センサを用いた自己位置同定	
│├─CeilingNavigation	天井ナビゲーションRTC	
│└─RTCCameraEye	天井カメラRTC	

※ 本版はバイナリのみ提供となります。

・遠隔操作 PC

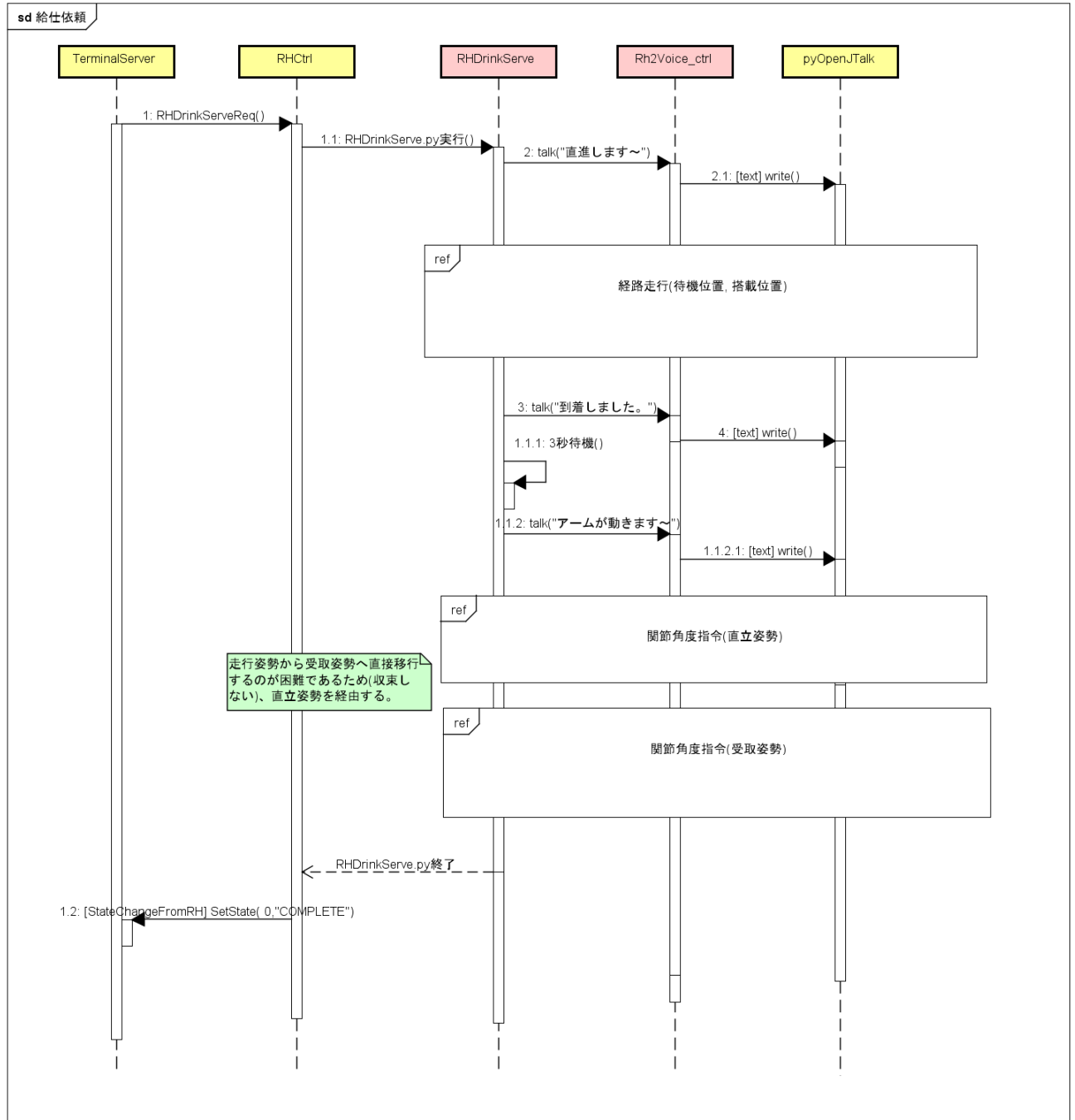
ディレクトリ	内容	備考
RemotePC	遠隔操作PC	
├── bin	実行ファイル	
│ ├── comp	バイナリ(Comp)	
│ ├── gui	スクリプト	
│ └── script	シナリオスクリプト	
│ ├── control	作業系(アーム)RTC制御スクリプト	
│ ├── Rh2Arm	移動系(台車)RTC制御スクリプト	
│ ├── Rh2Base	作業系(カメラ)RTC制御スクリプト	
│ ├── Rh2Cam	RH統括RTC制御スクリプト	
│ └── Rh2Voice	コミュニケーション系RTC制御スクリプト	
│ └── openhrp3	OpenHRP3シミュレーション用動作スクリプト	
├── so	バイナリ(so)	
├── etc	設定ファイル	
│ ├── conf	rtc.conf	
│ ├── data	データファイル	
│ │ ├── mapdata	経路計画用地図データベース	
│ │ │ └── robot		
│ │ │ ├── kobe_green_space_park	神戸中央緑地公園	
│ │ │ ├── kobe_kimec_center_building	神戸キメックセンタービル	
│ │ │ ├── rtc-center	RTC再利用技術研究センター	
│ │ │ └── virtual_space	仮想空間	
│ └── sound	音声ファイル	
├── env	pythonスクリプト 用環境変数設定ファイル	
├── idl	IDLファイル	
├── include	共通ヘッダファイル	
├── lib	共通ライブラリ	
└── src	ソースファイル	
├── HumanRobotInteraction	コミュニケーション系RTC	
│ └── SoundPlayer	音声再生RTC	
├── MappingAndPlanning	経路計画	
│ ├── dijkstra	ダイクストラRTC	※
│ ├── DispPosition	現在自己位置表示RTC	
│ ├── hazardmesh	ハザードメッシュRTC	※
│ ├── managementofareamap	地図情報管理RTC	※
│ └── PathEnv	パス情報変換RTC	※
├── MotionPlanning	運動計画	
├── RemoteControl	遠隔操作	
│ └── GamePad	ゲームパッド RTC	※
├── ScenarioPlayer	シナリオ制御	
└── RHCtrl	RH統括制御RTC	

※ 本版はバイナリのみの提供となります。

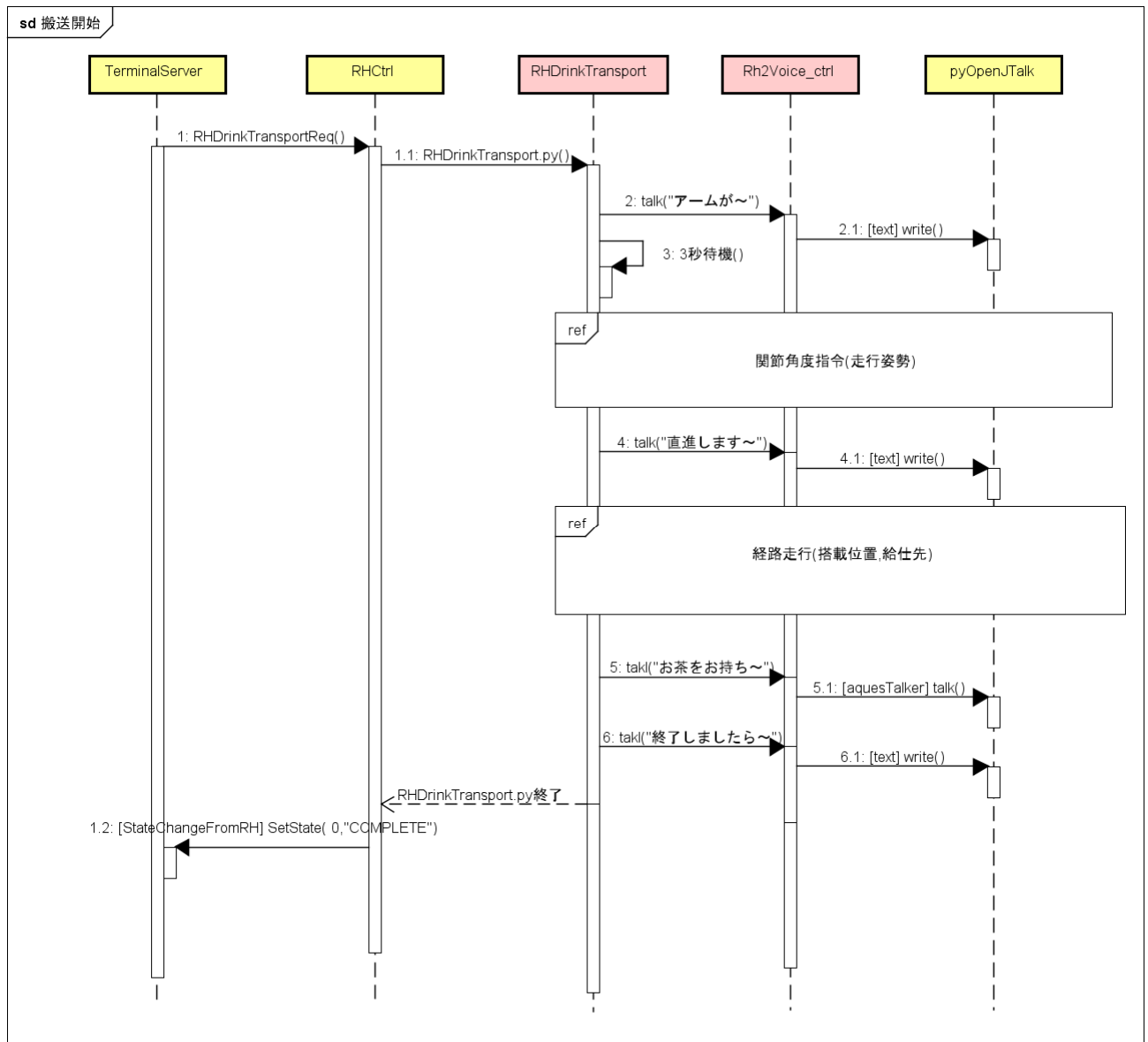
3. 動作シーケンス

1. 給仕

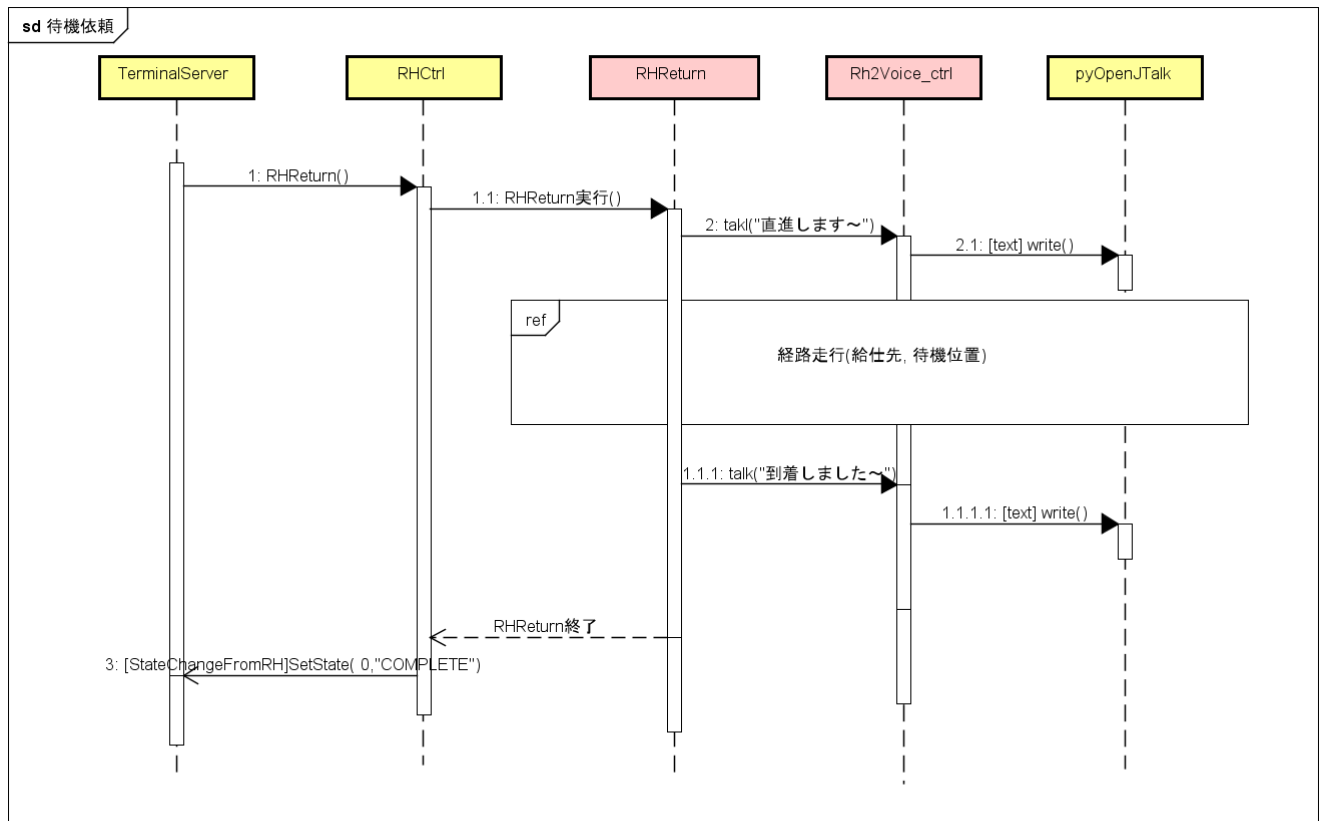
1_1. 給仕依頼



1_2.搬送開始

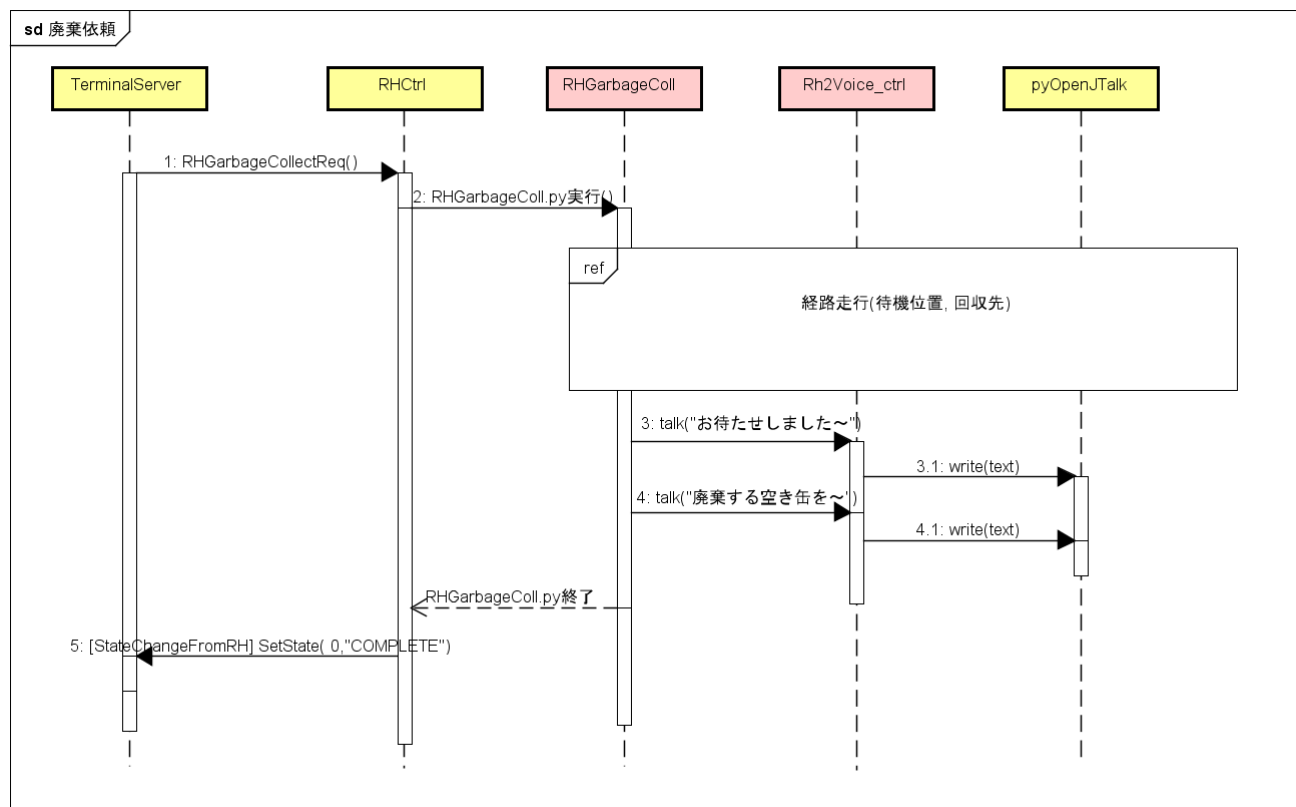


1_3.待機依頼

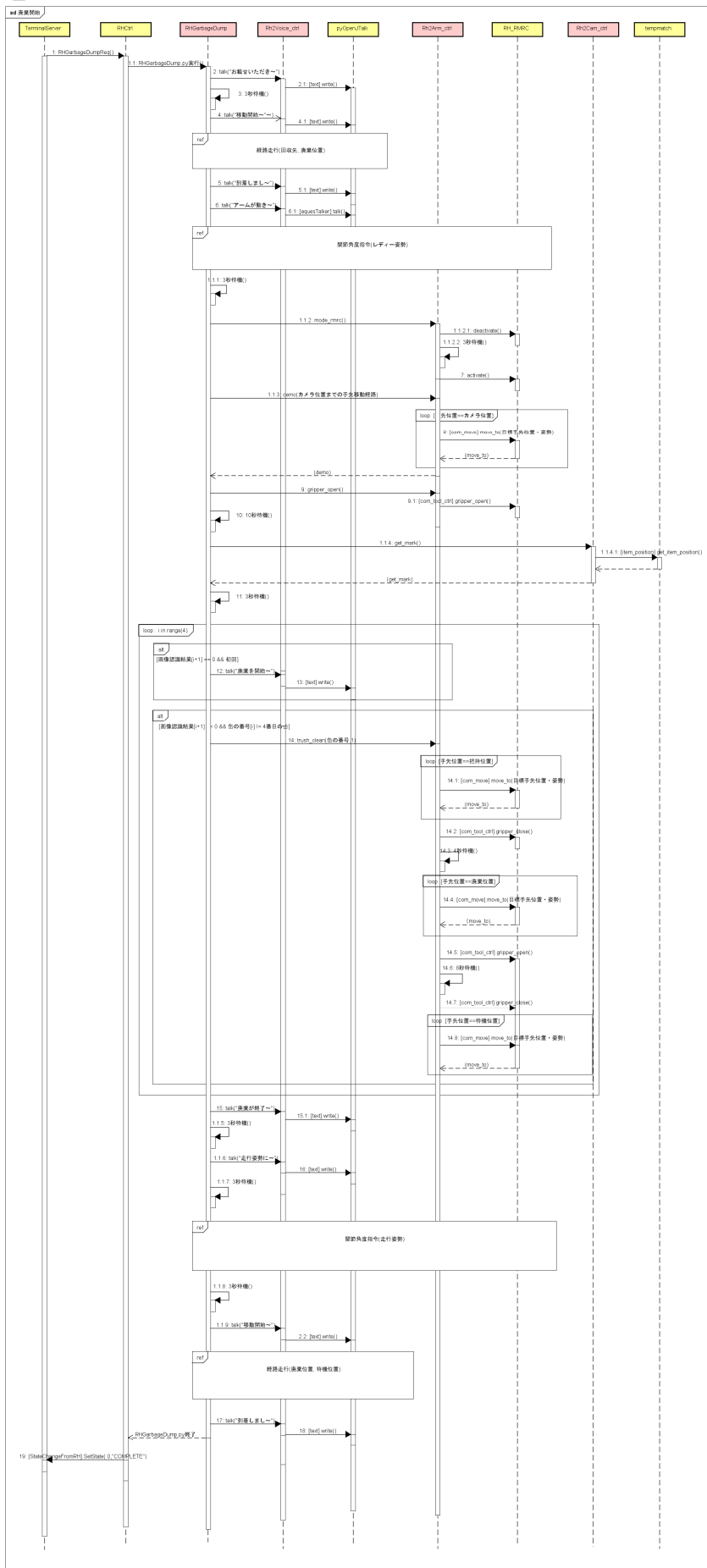


2. 廃棄

2_1.廃棄依頼

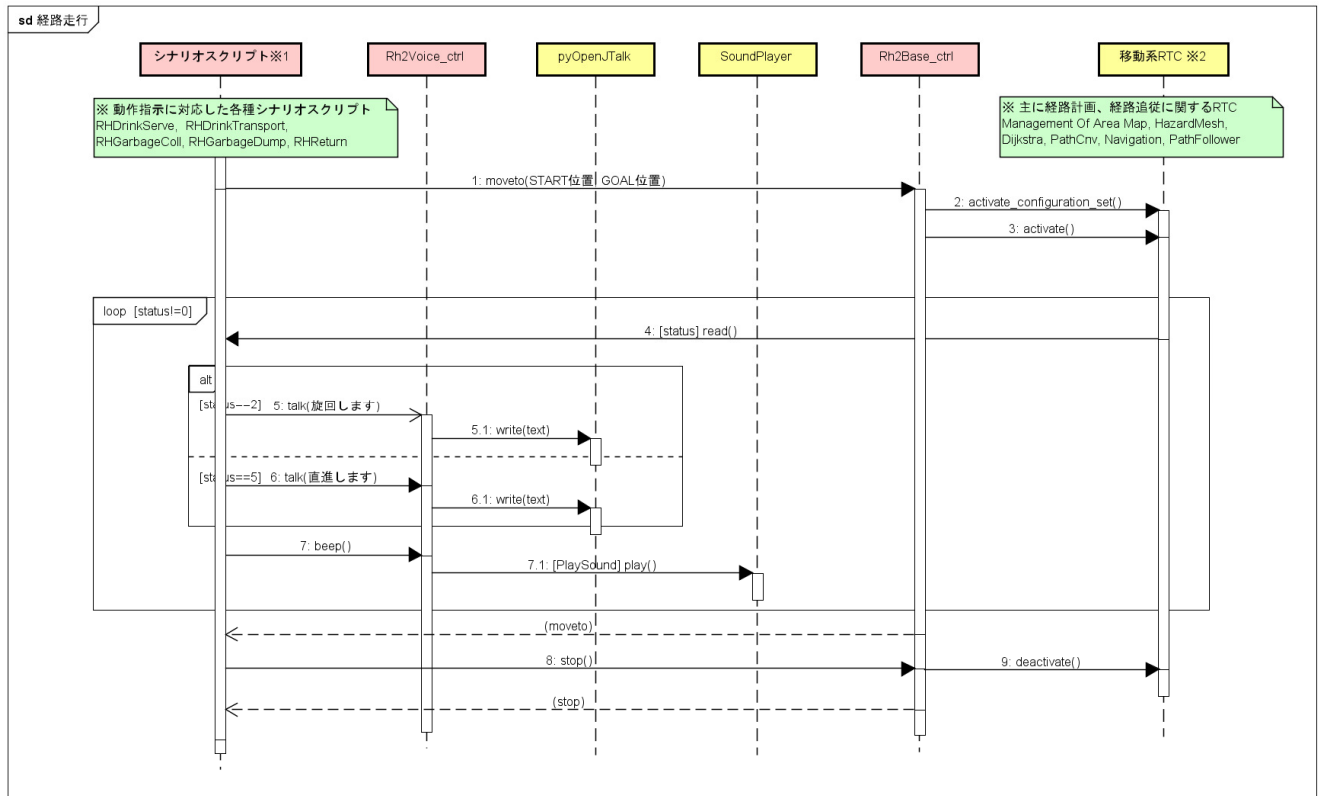


2_2.廃棄開始

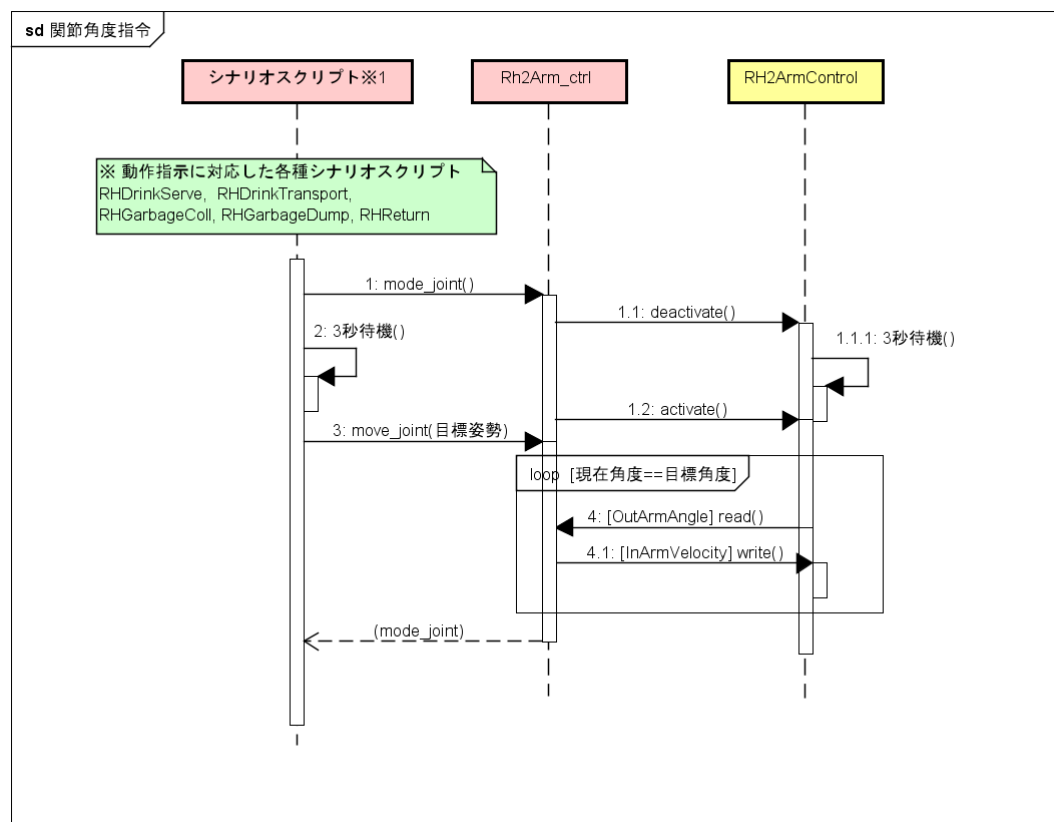


3.共通

3_1.経路走行



3_2.関節角度指令



4. モジュール仕様

4.1. RTC 仕様

4.1.1. RHCtrl

1. 機能概要

制御端末 RTC からのリクエストをもとにシナリオ動作(給仕・廃棄)の実行を行う。
その他 RH の状態管理、バッテリー管理を行う。

2. 基本情報

2.1.動作環境

種別	RTC
ベンダ名	RTC 再利用技術研究センター
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	特になし

2.2.動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	RemotePC/src/ScenarioPlayer/RHCtrl
-------	------------------------------------

3. ポート情報

・データポート(InPort)

なし

・データポート(OutPort)

なし

・サービスポート(Provider)

項目	値
サービスポート名	RefHardCtrlProv
インターフェイス名	RefHardCtrl
サービスの型	RefHardCtrl

・サービスポート(Consumer)

項目	値
サービスポート名	ControlTerminalCons
インターフェイス名	ControlTerminal
サービスの型	StateChangeFromRH

・コンフィグレーション(Configuration set)
なし

4. サービスの型:RefHardCtrl 定義

・RHReturn

関数名	RHReturn			
引数	名称	型	I/O	説明
	base	Position2D	入力	待機先
戻り値	値			説明
	void			-
説明	待機先へ移動をおこなう RHReturn.py を実行			
備考				

・RHDrinkServeReq

関数名	RHDrinkServeReq			
引数	名称	型	I/O	説明
	visitor	Position2D	入力	待機先
	warehouse	Position2D	入力	ドリンク 格納庫先
	name	string	入力	来訪者情報－氏名
戻り値	値			説明
	void			-
説明	RH ドリンク 格納をおこなう RHDrinkServe.py を実行			
備考				

•RHDrinkTransportReq

関数名	RHDrinkTransportReq			
引数	名称	型	I/O	説明
	dist	Position2D	入力	給仕先
	name	string	入力	来訪者情報－氏名
戻り値	値			説明
	void			-
説明	RH ドリンク給仕をおこなう RHDrinkTransport.py を実行			
備考				

•RHGarbageCollectReq

関数名	RHGarbageCollectReq			
引数	名称	型	I/O	説明
	dist	Position2D	入力	ドリンク回収先
	name	string	入力	来訪者情報－氏名
戻り値	値			説明
	void			-
説明	RH ドリンク回収をおこなう RHGarbageColl.py を実行			
備考				

•RHGarbageDumpReq

関数名	RHGarbageDumpReq			
引数	名称	型	I/O	説明
	dumpsite	Position2D	入力	廃棄先
	name	string	入力	来訪者情報－氏名
戻り値	値			説明
	void			-
説明	RH ドリンク廃棄をおこなう RHGarbageDump.py を実行			
備考				

4.1.2. BatteryWatcher

1. 機能概要

バッテリー監視を行う。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	RTC 再利用技術研究センター
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	m4api

2.2. 動作条件

実行周期	1Hz
------	-----

2.3. 格納場所

ソース一式	PCinRH/src/InternalWatch/BatteryWatcher
-------	---

3. ポート情報

・データポート (InPort)

なし

・データポート (OutPort)

名称	型	データ長	説明
RH_VIN	TimedFloat	—	電源ボードへの入力電圧 (バッテリー電圧)
Voltage3	TimedFloat	—	3.3V 出力の実際の出力電圧
Voltage5	TimedFloat	—	5V 出力の実際の電圧
Voltage12	TimedFloat	—	12V 出力の実際の電圧
Temperature	TimedShort	—	温度

・サービスポート (Provider)

なし

・サービスポート(Consumer)

項目	値
サービスポート名	BatteryAlert
インターフェイス名	BatteryAlert
サービスの型	BatteryAlert

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
ThresholdVoltage	float	21	電圧閾値

4. サービス関数定義(PlaySound)

・BatteryWatcher

関数名	BatteryWatcher			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	void			-
説明	コンフィグレーション値より電圧が低くなったら、BatteryAlert を呼ぶ			
備考				

4.1.3. Management Of Area Map

1. 機能概要

グリッドマップデータの情報を出力するためのモジュール。
読み出しマップの GUI 表示が可能。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	国際レスキューシステム研究機構
バージョン	0.1.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	gtk+-2.0, libglade-2.0, gmodule-2.0

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	RemotePC/src/MappingAndPlanning/managementofareamap
-------	---

※本版はバイナリのための提供となります。

3. ポート情報

・データポート(InPort)

名称	型	データ長	説明
MapOrder	TimedMapOrder	－	地図データ要求

・データポート(OutPort)

名称	型	データ長	説明
MapData	TimedMapData	－	地図データ
ArcNode	TimedArcNode	－	ラスターデータ
Landmark	TimedLandmark	－	ベクターデータ

・サービスポート(Provider)

なし

・サービスポート(Consumer)

なし

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
robotmapaddress	string	../mapdata/robot/	robot マップアドレス
gui	string	Off	GUI 動作[off, on]
guiaddress	string	../gui/managementofareamap	GUI プログラムアドレス
ip	string	127.0.0.1	IP アドレス
sendport	string	10004	送信ポート番号
receiveport	string	10003	受信ポート番号

4. 独自データ型

•TimedMapOrder(intellirobot.idl)

```

/*****
/*  tm      -> gettimeofday 関数より生成したタイムスタンプ                      */
/*  trigger -> 動作を開始するためのトリガー [0 から 1 へ変化した場合のみ反応] */
/*  mode     -> アクセスモード                                                  */
/*          bit 0 : 0 -> raster image      1 -> vector image                    */
/*          bit 1 : 0 -> read              1 -> write                          */
/*          bit 2 : リザーブ                                                       */
/*          bit 3 : リザーブ                                                       */
/*          bit 4 : リザーブ                                                       */
/*          bit 5 : リザーブ                                                       */
/*          bit 6 : リザーブ                                                       */
/*          bit 7 : リザーブ                                                       */
/*          bit 8 : リザーブ                                                       */
/*          bit 9 : リザーブ                                                       */
/*          bit10 : リザーブ                                                       */
/*          bit11 : リザーブ                                                       */
/*          bit12 : リザーブ                                                       */
/*          bit13 : リザーブ                                                       */
/*          bit14 : リザーブ                                                       */
/*          bit15 : リザーブ                                                       */
/*  location -> 地域名                  [ディレクトリ名]                      */
/*  mapset    -> 利用者マップセット名 [ディレクトリ名]                      */
/*  type      -> データタイプ            [ファイル名]                        */
/*  origin_x  -> 対象原点 x 座標 [mm]                                          */
/*  origin_y  -> 対象原点 y 座標 [mm]                                          */
/*  size_x    -> 原点からの対象範囲 x 座標 [mm]                              */
/*  size_y    -> 原点からの対象範囲 y 座標 [mm]                              */
/*          [注意] (origin_x, origin_y)を原点(左下)として,                    */
/*                  (size_x)*(size_y)の長方形となる。                        */
*****/
struct TimedMapOrder{
    Time tm;                // タイムスタンプ
    unsigned short trigger; // 動作トリガー
    unsigned short mode;    // アクセスモード
    sequence<char> location; // 地域名
    sequence<char> mapset;   // 利用者マップセット名
    sequence<char> type;     // データタイプ
    long origin_x;           // 対象原点 x 座標 [mm]
    long origin_y;           // 対象原点 y 座標 [mm]
    unsigned long size_x;    // 対象範囲 x 座標 [mm]
    unsigned long size_y;    // 対象範囲 y 座標 [mm]
};

```

•TimedMapData(intellirobot.idl)

```

/*****
/*  tm          -> 動作トリガー起動時の入力タイムスタンプ          */
/*  mode        -> アクセスモード                                */
/*              bit 0 : 0 -> raster image    1 -> vector image      */
/*              bit 1 : 0 -> read            1 -> write              */
/*              bit 2 : リザーブ                                */
/*              bit 3 : リザーブ                                */
/*              bit 4 : リザーブ                                */
/*              bit 5 : リザーブ                                */
/*              bit 6 : リザーブ                                */
/*              bit 7 : リザーブ                                */
/*              bit 8 : リザーブ                                */
/*              bit 9 : リザーブ                                */
/*              bit10 : リザーブ                               */
/*              bit11 : リザーブ                               */
/*              bit12 : リザーブ                               */
/*              bit13 : リザーブ                               */
/*              bit14 : リザーブ                               */
/*              bit15 : リザーブ                               */
/*  location     -> 地域名          [ディレクトリ名]              */
/*  mapset       -> 利用者マップセット名 [ディレクトリ名]        */
/*  type         -> データタイプ      [ファイル名]                */
/*  origin_x     -> 対象原点 x 座標 [mm]                          */
/*  origin_y     -> 対象原点 y 座標 [mm]                          */
/*  size_x       -> 原点からの対象範囲 x 座標 [mm]                */
/*  size_y       -> 原点からの対象範囲 y 座標 [mm]                */
/*  explanation  -> 取得した地図の説明                            */
/*  map_origin_x -> 参照した地図の原点 x 座標 [mm]                */
/*  map_origin_y -> 参照した地図の原点 y 座標 [mm]                */
/*  map_size_x   -> 参照した地図の元のサイズ [x 座標:mm]          */
/*  map_size_y   -> 参照した地図の元のサイズ [y 座標:mm]          */
/*  pixel_size_x -> 取得した地図の 1 ピクセルあたりのサイズ [x 座標:mm] */
/*  pixel_size_y -> 取得した地図の 1 ピクセルあたりのサイズ [y 座標:mm] */
/*  category     -> 地図データのカテゴリ                          */
/*  mapdata      -> カテゴリに対応した地図データ                  */
/*
/*      [注意] (origin_x, origin_y)を原点(左下)として,
/*              (size_x)*(size_y)の長方形となる.
/*      [注意] category のデータ区切りはカンマとなる.
/*      [注意] category は, カテゴリ番号->カテゴリの順番に送信される.
/*
*****/

```

```

struct TimedMapData{
    Time tm;                // タイムスタンプ
    unsigned short mode;    // アクセスモード
    sequence<char> location; // 地域名
    sequence<char> mapset;   // 利用者マップセット名
    sequence<char> type;     // データタイプ
    long origin_x;          // 対象原点 x 座標 [mm]
    long origin_y;          // 対象原点 y 座標 [mm]
    unsigned long size_x;    // 対象範囲 x 座標 [mm]
    unsigned long size_y;    // 対象範囲 y 座標 [mm]
    sequence<char> explanation; // 地図説明
    long map_origin_x;       // 地図原点 x 座標 [mm]
    long map_origin_y;       // 地図原点 y 座標 [mm]
    unsigned long map_size_x; // 地図サイズ x 座標 [mm]
    unsigned long map_size_y; // 地図サイズ y 座標 [mm]
    unsigned long pixel_size_x; // ピクセルサイズ x 座標 [mm]
    unsigned long pixel_size_y; // ピクセルサイズ y 座標 [mm]
    sequence<char> category;  // カテゴリデータ
    sequence<unsigned long> mapdata; // 地図データ
};

```


• TimedArcNode(intellirobot.idl)

```

/*****
/*  tm          -> タイムスタンプ                      */
/*  arc_data    -> アークデータ                        */
/*  node_data   -> ノードデータ                        */
*****/
struct TimedArcNode{
    Time tm;                      // タイムスタンプ
    sequence<Arc> arc_data;       // アークデータ
    sequence<Node> node_data;     // ノードデータ
};

// アーク
struct Arc{
    unsigned long start_id;       // 接続開始ノード ID
    unsigned long end_id;        // 接続終了ノード ID
    double cost;                 // コスト
};

// ノード
struct Node{
    unsigned long id;            // ノード ID
    double x;                   // x 座標 [m]
    double y;                   // y 座標 [m]
    double theta;               // 方向 [rad]
    sequence<char> explanation;
};

```

•TimedLandmark(intellirobot.idl)

```

/*****
/*  tm          -> タイムスタンプ                                     */
/*  line_data   -> 直線ランドマークデータ                             */
/*  circle_data -> 円ランドマークデータ                               */
*****/
struct TimedLandmark{
    Time tm;                // タイムスタンプ
    sequence<Line> line_data; // 直線ランドマークデータ
    sequence<Circle> circle_data; // 円ランドマークデータ
};

// 直線ランドマーク
struct Line{
    double x;                // x 座標 [m]
    double y;                // y 座標 [m]
    double theta;            // 角度 [rad]
    double r;                // 長さの 1/2 [m]
};

// 円ランドマーク
struct Circle{
    double x;                // x 座標 [m]
    double y;                // y 座標 [m]
    double r;                // 半径 [m]
};

```

4.1.4. HazardMesh

1. 機能概要

グリッドマップデータの縮小を行うためのモジュール。
地図情報管理モジュールのグリッドマップデータを用いて、
ダイクストラによる経路計画を立てるための下準備に使用。
マップの GUI 表示が可能。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	国際レスキューシステム研究機構
バージョン	0.1.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	gtk+-2.0, libglade-2.0, gmodule-2.0

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	RemotePC/src/MappingAndPlanning/hazardmesh
-------	--

※本版はバイナリのための提供となります。

3. ポート情報

・データポート(InPort)

名称	型	データ長	説明
MapData	TimedMapData	—	地図データ
MeshOrder	TimedMeshOrder	—	メッシュデータ要求

・データポート(OutPort)

名称	型	データ長	説明
MapOrder	TimedMapOrder	—	地図データ要求
MeshData	TimedMapData	—	メッシュデータ

・サービスポート(Provider)

なし

•サービスポート(Consumer)

なし

•コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
gui	string	Off	GUI 動作[off, on]
guiaddress	string	../gui/managementofareamap	GUI プログラムアドレス
ip	string	127.0.0.1	IP アドレス
sendport	string	10004	送信ポート番号
receiveport	string	10003	受信ポート番号

4. 独自データ型

•TimedMeshOrder(intellirobot.idl)

```

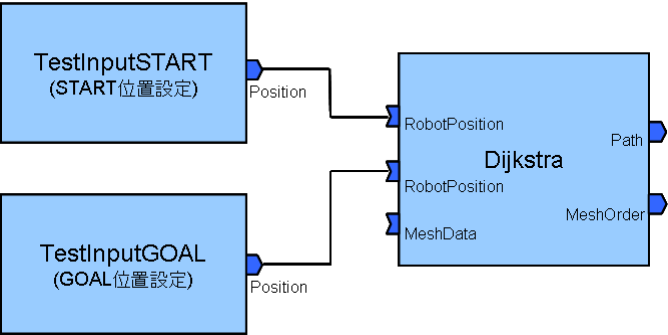
/*****
/*  tm          -> gettimeofday 関数より生成したタイムスタンプ          */
/*  trigger      -> 動作を開始するためのトリガー [0 から 1 へ変化した場合のみ反応] */
/*  mode         -> アクセスモード                                     */
/*               bit 0 : 0 -> raster image   1 -> vector image          */
/*               bit 1 : 0 -> read           1 -> write                  */
/*               bit 2 : リザーブ                                         */
/*               bit 3 : リザーブ                                         */
/*               bit 4 : リザーブ                                         */
/*               bit 5 : リザーブ                                         */
/*               bit 6 : リザーブ                                         */
/*               bit 7 : リザーブ                                         */
/*               bit 8 : リザーブ                                         */
/*               bit 9 : リザーブ                                         */
/*               bit10 : リザーブ                                        */
/*               bit11 : リザーブ                                        */
/*               bit12 : リザーブ                                        */
/*               bit13 : リザーブ                                        */
/*               bit14 : リザーブ                                        */
/*               bit15 : リザーブ                                        */
/*  location      -> 地域名                [ディレクトリ名]          */
/*  mapset        -> 利用者マップセット名 [ディレクトリ名]          */
/*  type          -> データタイプ            [ファイル名]            */
/*  origin_x      -> 対象原点 x 座標 [mm]                               */
/*  origin_y      -> 対象原点 y 座標 [mm]                               */
/*  size_x        -> 原点からの対象範囲 x 座標 [mm]                   */
/*  size_y        -> 原点からの対象範囲 y 座標 [mm]                   */
/*  pixel_size_x  -> 取得した地図の 1 ピクセルあたりのサイズ [x 座標:mm] */
/*  pixel_size_y  -> 取得した地図の 1 ピクセルあたりのサイズ [y 座標:mm] */
/*               [注意] (origin_x, origin_y)を原点(左下)として,          */
/*               (size_x)*(size_y)の長方形となる.                      */
/*               *****/
struct TimedMeshOrder{
    Time tm;                // タイムスタンプ
    unsigned short trigger; // 動作トリガー
    unsigned short mode;    // アクセスモード
    sequence<char> location; // 地域名
    sequence<char> mapset;   // 利用者マップセット名
    sequence<char> type;     // データタイプ
    long origin_x;           // 対象原点 x 座標 [mm]
    long origin_y;           // 対象原点 y 座標 [mm]
    unsigned long size_x;    // 対象範囲 x 座標 [mm]
    unsigned long size_y;    // 対象範囲 y 座標 [mm]
    unsigned long pixel_size_x; // ピクセルサイズ x 座標 [mm]
    unsigned long pixel_size_y; // ピクセルサイズ y 座標 [mm]
};

```

4.1.5. Dijkstra

1. 機能概要

グリッドマップデータからダイクストラ法によって、経路を生成するためのモジュールです。
経路マップの GUI 表示が可能。



START／GOAL 位置はデータポート経由で設定を行うこととし、
本 RTC には設定用のテストコンポーネントを付属する。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	国際レスキューシステム研究機構
バージョン	0.1.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	gtk+-2.0libglade-2.0gmodule-2.0

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	RemotePC/src/MappingAndPlanning/dijkstra
-------	--

※本版はバイナリのための提供となります。

3. ポート情報

・データポート(InPort)

名称	型	データ長	説明
RobotPosition	IIS::TimedPosition	—	START 位置
RobotPosition	IIS::TimedPosition	—	GOAL 位置
MeshData	TimedMapData	—	メッシュデータ

・データポート(OutPort)

名称	型	データ長	説明
Path	PathSeq	—	走行経路配列
max_acceleration	TimedMeshOrdar	—	メッシュデータ要求

・サービスポート(Provider)

なし

・サービスポート(Consumer)

なし

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
location	string	rtc-center	地域名
mapset	string	fsi	利用者マップセット名
type	string	hazard	説明
originx	string	10	対象原点 x 座標 [mm]
originy	string	5	対象原点 y 座標 [mm]
sizeX	string	10000	対象範囲 x 座標 [mm]
sizeY	string	10500	対象範囲 y 座標 [mm]
pixelsizeX	string	100	ピクセルサイズ x 座標 [mm]
pixelsizeY	string	100	ピクセルサイズ y 座標 [mm]
gui	string	Off	GUI 動作[off, on]
guiaddress	string	../gui/dijkstra	GUI プログラムアドレス
ip	string	127.0.0.1	IP アドレス
sendport	string	10008	送信ポート番号
receiveport	string	10007	受信ポート番号

4. 独自データ型

• IIS:TimedPosition(intellirobot.idl)

```
struct TimedPosition {  
    RTC::Time tm;  
    sequence<long> id;  
    double x;  
    double y;  
    double theta;  
    sequence<double> error;  
};
```

• PathSeq(intellirobot.idl)

```
struct PathSeq{  
    sequence<Path> path_list;  
    Time tm;  
};
```

```
struct Path{  
    short type;  
    short coordinate;  
    double x;  
    double y;  
    double theta;  
    double v;  
    double w;  
    double r;  
    Time tm;  
};
```

Path の type は現在は数値により指定(変更予定)

RUN_FREE	0
RUN_STOP	1
RUN_LINEFOLLOW	2
RUN_TO_POINT	3
RUN_CIRCLEFOLLOW	4
RUN_SPIN	5
RUN_VEL	6
RUN_WHEEL_VEL	7

4.1.6. PathCnv

1. 機能概要

経路計画 RTC と経路走行 RTC 間データ調整を行う。
(ゴール地点・経路途中の方向情報の付加)

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	セグウェイジャパン
バージョン	1.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	特になし

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	RemotePC/src/MappingAndPlanning/PathCnv
-------	---

※本版はバイナリのための提供となります。

3. ポート情報

・データポート(InPort)

名称	型	データ長	説明
Path	PathSeq	—	走行経路配列

・データポート(OutPort)

名称	型	データ長	説明
min_path	PathSeq	—	走行経路配列

・サービスポート(Provider)

なし

・サービスポート(Consumer)

なし

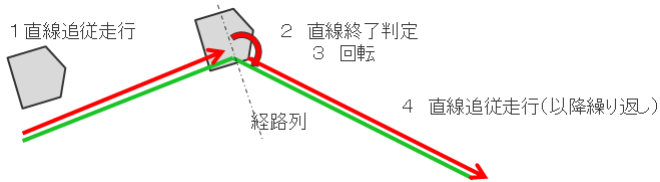
・コンフィグレーション(Configuration set)

なし

4.1.7. Navigation

1. 機能概要

指定された経路列(ウェイポイント群)を走行するように、走行コマンド(直線・回転・停止等)を出力する。



2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	東北大学
バージョン	1.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	特になし

2.2. 動作条件

実行周期	50Hz
------	------

2.3. 格納場所

ソース一式	PCinRH/src/PathTracking/Navigation_0.1
-------	--

※本版はバイナリのための提供となります。

3. ポート情報

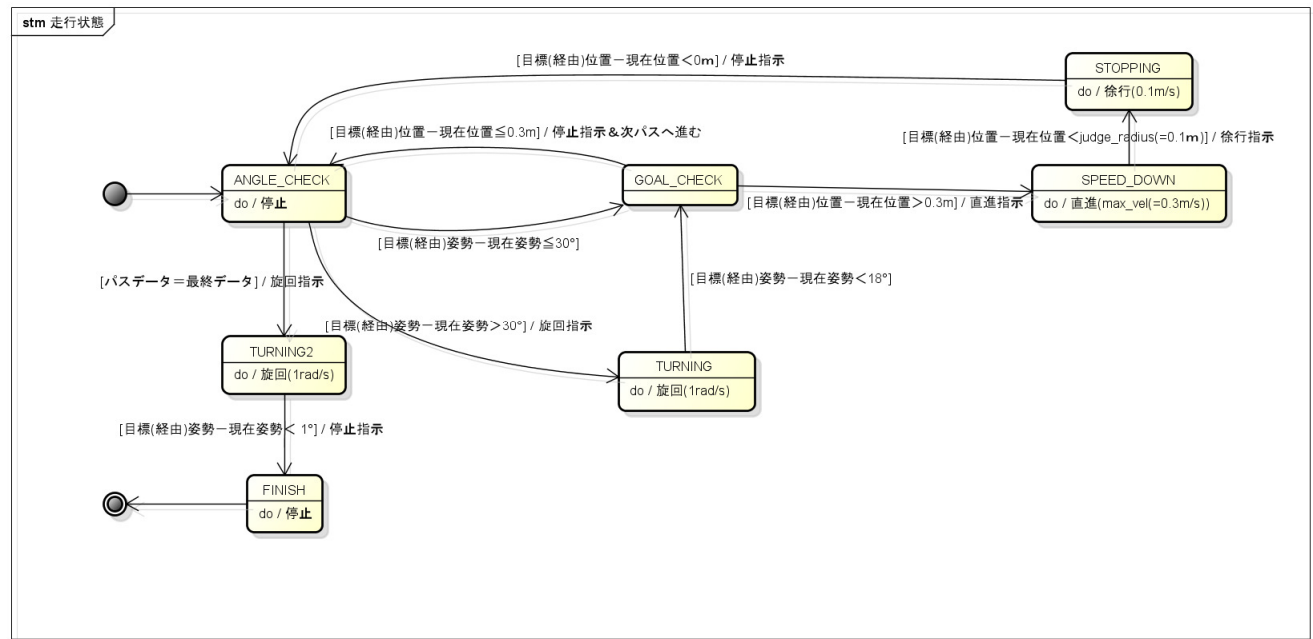
・データポート(InPort)

名称	型	データ長	説明
<code>position</code>	<code>IIS::TimedPosition</code>	—	現在位置
<code>min_path</code>	<code>PathSeq</code>	—	目標経路リスト

・データポート(OutPort)

名称	型	データ長	説明
<code>path</code>	<code>Path</code>	—	現在の経路
<code>max_speed</code>	<code>IIS::TimedVelocity</code>	—	最大速度
<code>max_acceleration</code>	<code>IIS::TimedVelocity</code>	—	最大加速度
<code>status</code>	<code>TimedState</code>	—	走行状態※

※走行状態



・サービスポート(Provider)
なし

・サービスポート(Consumer)
なし

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
judge_radius	double	0.5	ウェイポイントに到達したと判断する半径
max_vel	double	0.3	最大速度

4. 独自データ型

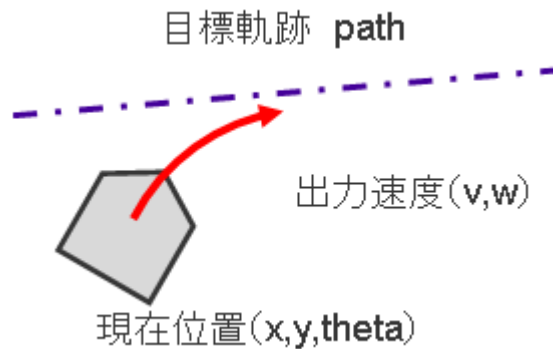
・IIS:TimedVelocity(intellirobot.idl)

```
struct TimedVelocity {
    Time tm;
    double v;
    double w;
};
```

4.1.8. PathFollower

1. 機能概要

指定された直線、円弧、回転等に沿うような速度を出力する。



2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	東北大学
バージョン	1.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	特になし

2.2. 動作条件

実行周期	50Hz
------	------

2.3. 格納場所

ソース一式	PCinRH/src/PathTracking/PathFollower_0.1
-------	--

※本版はバイナリのための提供となります。

3. ポート情報

・データポート (InPort)

名称	型	データ長	説明
position	IIS::TimedPosition	—	現在位置(x[m],y[m],θ[rad])
path	Path	—	目標経路
max_speed	IIS::TimedVelocity	—	最大速度
max_acceleration	IIS::TimedVelocity	—	最大加速度

•データポート(OutPort)

名称	型	データ長	説明
velocity	IIS:TimedVelocity	－	目標速度($v[m/s]$, $\omega[rad/s]$)

•サービスポート(Provider)

なし

•サービスポート(Consumer)

なし

•コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
max_v	double	0.8	最大速度
max_w	double	0.5	最大角速度
max_acc_v	double	0.1	最大加速度
max_acc_w	double	0.1	最大角加速度
line_C1	double	0.01	経路追従パラメータ
line_K1	double	800	経路追従パラメータ 経路からの距離に関する係数
line_K2	double	300	経路追従パラメータ 経路からの角度に関する係数
line_K3	double	400	経路追従パラメータ 角速度に関する係数
line_Dist	double	0.1	経路追従パラメータ 経路からの距離の最大値
control_cycle	double	0.02	制御周期[s]

4.1.9. SwitchInputRTC

1. 機能概要

手動操作モードと自律走行モードの入力切替を行う。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	産業技術総合研究所
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	特になし

2.2. 動作条件

実行周期	50Hz
------	------

2.3. 格納場所

ソース一式	PCinRH/src/PathTracking/SwitchJoyStickRTC
-------	---

3. ポート情報

・データポート(InPort)

名称	型	データ長	説明
inVelJoy	IIS:TimedVelocity	—	ゲームパッド（手動操作）からの目標速度($v[m/s]$, $\omega[rad/s]$)
inVelAuto	IIS:TimedVelocity	—	自律走行の目標速度($v[m/s]$, $\omega[rad/s]$)

・データポート(OutPort)

名称	型	データ長	説明
outVel	IIS:TimedVelocity	—	ロボットへの目標速度($v[m/s]$, $\omega[rad/s]$)

・サービスポート(Provider)

なし

・サービスポート(Consumer)

なし

- ・コンフィグレーション(Configuration set)
なし

4.1.10. RefHardRh2

- 機能概要
前川製作所製リファレンスハードウェア 2 号機実機(台車)を制御する。

- 基本情報

- 2.1. 動作環境

種別	RTC
ベンダ名	産業技術総合研究所 OpenINVENT Team
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	libsvm

- 2.2. 動作条件

実行周期	デフォルト(50Hz)
------	-------------

- 2.3. 格納場所

ソース一式	PCinRH/src/MobileRobotController/RefHard2
-------	---

3. ポート情報

- ・データポート(InPort)

名称	型	データ長	説明
InTargetVelocity	TimedVelocity		$V[\text{m/sec}], \omega[\text{rad/sec}]$

- ・データポート(OutPort)

名称	型	データ長	説明
OutWheelAngle	TimedDoubleSeq		L と R の Wheel 値

- ・サービスポート(Provider)
なし

- ・サービスポート(Consumer)
なし

- ・コンフィグレーション(Configuration set)
なし

4.1.11. Odometry

1. 機能概要

車輪エンコーダの回転角の積分計算により自己位置を推定する.

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	産業技術総合研究所 OpenINVENT Team
バージョン	4.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	特になし

2.2. 動作条件

実行周期	50Hz
------	------

2.3. 格納場所

ソース一式	PCinRH/src/MobileRobotController/Odometry
-------	---

3. ポート情報

•データポート(InPort)

名称	型	データ長	説明
CurrentWheelAngle	TimedDoubleSeq	—	車輪回転角[rad]
LocalizedPosition	IIS::TimedPose2D	—	推定現在位置(x[m],y[m],θ[rad])

•データポート(OutPort)

名称	型	データ長	説明
OdometryPosition	IIS::TimedPose2D	—	オドメトリによる推定位置 (x[m],y[m],θ[rad])

•サービスポート(Provider)

なし

•サービスポート(Consumer)

なし

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
leftWheelID	int	0	左車輪の ID
rightWheelID	int	1	右車輪の ID
radiusOfLeftWheel	double	0.1	左車輪の半径[m]
radiusOfRightWheel	double	0.1	右車輪の半径[m]
lengthOfAxle	double	0.441	車軸長[m]
radiusOfBodyArea	double	0.45	車軸中心から車体コーナーまでの長さ[m]

4.1.12. LocalizeCenter

1. 機能概要

オドメトリによる位置推定と天井カメラによる位置推定を融合して自己位置推定を行う。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	産業技術総合研究所
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	特になし

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.2. 格納場所

ソース一式	PCinRH/src/PathTracking/LocalizeCenter
-------	--

3. ポート情報

・データポート(InPort)

名称	型	データ長	説明
dp_odmIn	IIS::TimedPose2D	ー	オドメトリによる推定位置 (x[m],y[m],θ[rad])
dp_ceilIn	IIS::TimedPose2D	ー	天井画像マッチングによる 推定位置(x[m],y[m],θ[rad])

- ・データポート(OutPort)

名称	型	データ長	説明
dp_out0	IIS::TimedPose2D	－	推定位置(x[m],y[m],θ[rad])

※今版では旧 I/F である「IIS:TimedPosition」と混在しており、「PathFollower」「Navigation」への入力についてはデータ変換を行う。

- ・サービスポート(Provider)

なし

- ・サービスポート(Consumer)

なし

- ・コンフィグレーション(Configuration set)

なし

4.1.13. CameraEye

1. 機能概要

USB カメラからの画像を取得・処理・データ変換して天井画像位置推定コンポーネントに渡す。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	奈良先端科学技術大学院大学
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	opencv-1.0 以上

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	PCinRH/src/Sensor-BasedNavigation/RTCCameraEye
-------	--

3. ポート情報

- ・データポート(InPort)

なし

- ・データポート(OutPort)

名称	型	データ長	説明
ImageData	TimedOctetSeq	－	画像データ

- ・サービスポート(Provider)

なし

- ・サービスポート(Consumer)

なし

- ・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
Flip	int	2	画像反転の有無, 種類
CaribFile	string	ncm_cameraparam.xml	歪み補正用のパラメータファイル名
Magnification	double	1.00	画像拡大率 (不使用)
CaptureWindow	string	on	キャプチャした画像の表示/非表示
USBdevID	int	1	カメラのデバイス番号

4.1.14. CeilingNavigation

1. 機能概要

天井画像マッチングによる自己位置推定を行う。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	奈良先端科学技術大学院大学
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	opencv-1.0 以上

2.2. 動作条件

実行周期	30Hz
------	------

2.3. 格納場所

ソース一式	PCinRH/src/Sensor-BasedNavigation/CeilingNavigation
-------	---

3. ポート情報

•データポート(InPort)

名称	型	データ長	説明
CameraData	TimedOctetSeq	－	画像データ
LocalizedPosition	IIS::TimedPose2D	－	推定現在位置(x[m],y[m],θ[rad])

•データポート(OutPort)

名称	型	データ長	説明
CeilingOdometry	TimedOdometry	－	画像上での推定位置 (x[pixel],y[pixel],θ[rad])
CeilingOdometryd	TimedDoubleSeq		- 不使用 -
CeilingPosition	IIS::TimedPose2D		推定現在位置(x[m],y[m],θ[rad])

•サービスポート(Provider)

なし

•サービスポート(Consumer)

なし

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
NavigationMap	string	./rotRTCcenterCeiling.BMP	天井画像地図ファイル名
OfflineImage	string	- なし -	オフライン画像ファイル 格納フォルダ名
X_POS	int	38	画像上での初期 X 位置 [pixel]
Y_POS	int	239	画像上での初期 Y 位置 [pixel]
THETA	double	3.14159265	画像上での初期方位角 [rad]
BlockResolution	int	1	画像回転分解能[deg]
BlockSize	int	64	画像縮小サイズ[pixel]
BlockCoefficient	double	0.3	画像縮小補正值
SearchScope	int	2	画像探索範囲[pixel]
BlackWhiteValue	int	220	二値化の閾値
SearchScope	int	2	画像探索範囲[pixel]
Center_X	int	160	画像回転中心座標(X) [pixel]
Center_Y	int	120	画像回転中心座標(Y) [pixel]
RealMapHeight	double	100.0	実際の地図のサイズ(縦) - 不使用
RealMapWidth	double	100.0	実際の地図のサイズ(横) - 不使用
VirtualMapHeight	double	100.0	仮想地図のサイズ(縦) - 不使用
VirtualMapWidth	double	100.0	仮想地図のサイズ(横) - 不使用

4.1.15. GamePad

1. 機能概要

遠隔手動操作用のゲームパッドの入力をロボットへの速度指令に変換して出力する.
ロジタールの 10 ボタンゲームパッドに対応.

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	セグウェイジャパン
バージョン	0.1.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	libSDL

2.2. 動作条件

実行周期	40Hz
------	------

2.3. 格納場所

ソース一式	RemotePC/src/RemoteControl/GamePad
-------	------------------------------------

※本版はバイナリのための提供となります。

3. ポート情報

- データポート (InPort)
なし

・データポート(OutPort)

名称	型	データ長	説明
Button	RTC::TimedULong	—	ボタン状態
PushedButton	RTC::TimedULong	—	押したボタン状態
StickLX	RTC::TimedFloat	—	アナログスティック状態 (左スティック X 方向)
StickLY	RTC::TimedFloat	—	アナログスティック状態 (左スティック Y 方向)
StickRX	RTC::TimedFloat	—	アナログスティック状態 (右スティック X 方向)
StickRY	RTC::TimedFloat	—	アナログスティック状態 (右スティック Y 方向)
StickLX	RTC::TimedDouble	—	アナログスティック状態 (左スティック X 方向)
StickLY	RTC::TimedDouble	—	アナログスティック状態 (左スティック Y 方向)
StickRX	RTC::TimedDouble	—	アナログスティック状態 (右スティック X 方向)
StickRY	RTC::TimedDouble	—	アナログスティック状態 (右スティック Y 方向)
Velocity	TimedVelocity	—	速度出力
VelocityIIS	IIS::TimedVelocity	—	IIS 形式速度出力 ロボットへの目標速度 ($v[m/s]$, $\omega[rad/s]$)

・サービスポート(Provider)

なし

・サービスポート(Consumer)

なし

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
Klx	float	1.0	左スティック X 方向に乗ずる係数
Kly	float	1.0	左スティック Y 方向に乗ずる係数
Krx	float	1.0	右スティック X 方向に乗ずる係数
Kry	float	1.0	右スティック Y 方向に乗ずる係数

4.1.16. DispPosition

1. 機能概要

ロボットの現在位置・方位角, 走行軌跡を地図画像に重畳表示する.

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	産業技術総合研究所
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	opencv-1.0 以上

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	RemotePC/src/MappingAndPlanning/DispPosition
-------	--

3. ポート情報

•データポート(InPort)

名称	型	データ長	説明
position	IIS::TimedPose2D	—	推定現在位置(x[m],y[m],θ[rad])

•データポート(OutPort)

なし

•サービスポート(Provider)

なし

•サービスポート(Consumer)

なし

•コンフィグレーション(Configuration set)

なし

4.1.17. RH_RMRC

1. 機能概要

アームの手先の位置・姿勢を3次元座標系で指示して動かすための RTC 群。

以下の2つの動作モードを切り替えることが可能。

- ・ 関節制御モード
- ・ 分解運動速度制御モード

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	RTC 再利用技術センター※
バージョン	1.0.0
動作 OS	Ubuntu10.04
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++,Python
コンパイラ	gcc 4.4.3
インタプリタ	Python2.6
依存ライブラリ	python-visual (シミュレータ動作時のみ) libgtkglext1 (シミュレータ動作時のみ) libgtkglext1-dev (シミュレータ動作時のみ)

※ 本モジュールは電気通信大学 末廣尚士氏によって PA10 向けに開発された「アームの分解運動速度制御 RTC 群」をベースに RTC 再利用技術センターにおいて RH 向けに単一 RTC 化を行ったものである。

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	PCinRH/src/MotionControl/RMRC
-------	-------------------------------

3. ポート情報

各 RTC の仕様及び内部で使用する I/F については以下の資料を参照のこと。

- ・「ロボットアーム(RH アーム)分解運動速度制御モジュール 機能仕様書」を参照。
- ・関節角速度制御アーム RTC の使い方 (<http://openrtm.sakura.ne.jp/cgi-bin/wiki/wiki.cgi/2009/1A32>)

4. ポート情報

- ・データポート (InPort)

名称	型	データ長	説明
Th	TimedDoubleSeq	n_size	現在の各軸角度情報 (単位: rad)

・データポート(OutPort)

名称	型	データ長	説明
Vel	TimedDoubleSeq	n_size	各軸の動作角速度 n_size：軸数 (単位：rad/sec)

・サービスポート(Provider)

項目	値
サービスポート名	ComMv
インターフェイス名	com_move
サービスの型	ComMv

項目	値
サービスポート名	ComFrmCtrl
インターフェイス名	com_frm_ctrl
サービスの型	ComFrmCtrl

項目	値
サービスポート名	ComFk
インターフェイス名	com_fk
サービスの型	ComFk

項目	値
サービスポート名	ComToolCtrl
インターフェイス名	com_tool_ctrl
サービスの型	ComToolCtrl

・サービスポート(Consumer)

なし

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
n_size	int	7	関節数 (≥ 1) 本モジュールではアーム自由度のデフォルト値を7とする。
m_max	int	100	特異値分解行列の最大行数 (≥ 1)

5. サービス関数定義

•moveto (ComMv)

関数名	move_to			
引数	名称	型	I/O	説明
	frm	Frame	入力	目標手先位置・姿勢
戻り値	値			説明
	void			-
説明	引数として渡された frm をデータポート (OutPort) に書き込む。			
備考				

•set_param (ComFrmCtrl)

関数名	set_param			
引数	名称	型	I/O	説明
	k_pv	double	入力	時定数 (k_gain) として設定される
	v_m	double	入力	最大速度 (v_max) として設定される (単位: mm/sec)
	k_pr	double	入力	並進・回転換算比率 (k_pos_rot) として設定される
戻り値	値			説明
	void			-
説明	並進速度、回転速度計算に使用するパラメータの設定を行う。			
備考	デフォルトでは k_pv = 1.0, v_m = 200, k_pr = 200 であり、基本的に k_pv と k_pr は変更を行わない。必要に応じて最大速度の調整を行う場合には、十分注意して適切なパラメータ設定をすること。			

•set_tool (ComFk)

関数名	set_tool			
引数	名称	型	I/O	説明
	frm	const Frame&	入力	手先の位置とツールの姿勢
戻り値	値			説明
	なし			-
説明	ツール長と手先座標系から見たツールの姿勢を設定する。			

4.1.18. RH_HandWrapper

1. 機能概要

主に前川製作所製リファレンスハードウェアを対象として、ハンドと一体式となっているアームを制御するためのラップ RTC である。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	RTC 再利用技術研究センター
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	特になし

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	PCinRH/src/MotionControl/RH_HandWrapper
-------	---

3. ポート情報

・データポート(InPort)

名称	型	データ長	説明
JointVel	TimedDoubleSeq	jointnum	アーム関節角速度 (単位: rad/sec)
ThPos	TimedDoubleSeq	jointnum+1	関節角度 (単位: rad) ツールの開口幅 (単位: m) ※

※今後共通インターフェース仕様に合わせ、長さの単位を mm へと修正する予定

・データポート(OutPort)

名称	型	データ長	説明
Th	TimedDoubleSeq	jointnum	現在のアーム関節角度 (単位: rad)
JointGripVel	TimedDoubleSeq	jointnum+1	アーム関節角速度 (単位: rad/sec) ツールの開閉速度 (単位: m/sec) ※

※今後共通インターフェース仕様に合わせ、長さの単位を mm へと修正する予定

・サービスポート(Provider)

項目	値
サービスポート名	ComToolCtrl
インターフェイス名	com_tool_ctrl
サービスの型	ComToolCtrl

・サービスポート(Consumer)

なし

・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
max_width	double	0.07 ※	最大開口幅（単位：m）※
grip_vel	double	0.01 ※	開閉速度（単位：m/sec）※
jointnum	int	6	アーム関節数（ ≥ 1 ） 本モジュールではアーム自由度のデフォルト値を 6 とする。

※今後共通インターフェース仕様に合わせ、長さの単位を mm へと修正する予定

4. サービス関数定義(ComToolCtrl)

・gripper_open

関数名	gripper_open			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	unsigned short			0 : 正常終了 その他 : 異常終了
説明	グリッパーを開く			
備考				

・gripper_close

関数名	gripper_close			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	unsigned short			0 : 正常終了 その他 : 異常終了
説明	グリッパーを閉じる			
備考				

・gripper_stop

関数名	gripper_stop			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	unsigned short			0 : 正常終了 その他 : 異常終了
説明	グリッパーの動作を止める			
備考				

•get_cond

関数名	get_cond			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	unsigned short			グリッパーの状態 -1:異常終了 0:閉完了 1:開完了 2:動作中 3:動作中断
説明	グリッパーの状態を取得			
備考				

•get_mode

関数名	get_mode			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	unsigned short			動作モード -1:異常終了 0:待機モード 1:開口モード 2:閉口モード
説明	グリッパーの動作モードを取得			
備考				

4.1.19. RH2ArmControl

1. 機能概要

前川製作所製リファレンスハードウェア 2 号機実機(アーム)を制御する。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	産業技術総合研究所 OpenINVENT Team
バージョン	4.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	libsvm

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	PCinRH/src/ManipulatorController/RefHard2_ArmController_ver3.0.0
-------	--

3. ポート情報

・データポート(InPort)

名称	型	データ長	説明
InArmVelocity	TimedDoubleSeq	指定なし	各軸の動作角速度 (単位: rad/sec) ツールの開閉速度 (単位: mm/sec)

・データポート(OutPort)

名称	型	データ長	説明
OutArmAngle	TimedDoubleSeq	7	関節角度 (単位: rad) ツールの開口幅 (単位: mm)

・サービスポート(Provider)

なし

・サービスポート(Consumer)

なし

・コンフィグレーション(Configuration set)

なし

4.1.20. tempmatch

1. 機能概要

与えられた 4 つのマーク(画像)の有無確認をおこなう。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	RTC 再利用技術研究センター
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
コンパイラ	gcc 4.4.3
依存ライブラリ	opencv-2.0

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	PCinRH/src/ObjectRecognition/tempmatchh
-------	---

3. ポート情報

・データポート(InPort)

なし

・データポート(OutPort)

なし

・サービスポート(Provider)

項目	値
サービスポート名	itemposition_service
インターフェイス名	item_position
サービスの型	item_position

・サービスポート(Consumer)

なし

- ・コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
int_disp	int	0	画像の表示/非表示
int_height	int	480	画像の高さ
int_width	int	640	画像の幅
int_match_level	int	70	認識率の閾値

4. サービスの型:item_position 定義

- ・get_item_position

関数名	get_item_position			
引数	名称	型	I/O	説明
	up_l	short	出力	上の段左側のマーク認識結果
	up_r	short	出力	上の段右側のマーク認識結果
	down_l	short	出力	下の段左側のマーク認識結果
	down_r	short	出力	下の段右側のマーク認識結果
戻り値	値			説明
	short			終了情報
説明	マークの認識ができたなら1、マークの認識ができなければ0を出力する。			
備考				

4.1.21. pyOpenJTalk

1. 機能概要

Open JTalk(<http://open-jtalk.sourceforge.net/>)は、フリーで利用できる日本語音声合成エンジンです。本コンポーネントは、受け取ったテキストデータを Open JTalk を使用して音声データに変換して出力します。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	産業総合技術研究所
バージョン	1.02
動作 OS	Ubuntu9.10 以上を推奨
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE(Python)
開発言語	Python
依存ライブラリ	OpenJTalk バージョン 1.00 以上

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

本モジュールのソースは SVN リポジトリから取得可能です。URL は以下の通り。

ソース一式	http://openhri.net/svn/OpenHRIVoice/
-------	---

※ その他詳細情報は OpenHRI ホームページ(<http://openhri.net/>)を参照。

4.1.22. PulseAudioOutput

1. 機能概要

標準音声出力デバイスに入力データを渡し、PulseAudio ライブラリを使用して音声を再生します。
本コンポーネントは、ubuntu9.04 以前のバージョンでは、正常動作しません。PortAudio ライブラリを用いたコンポーネントを使用してください。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	産業総合技術研究所
バージョン	1.02
動作 OS	Ubuntu9.10 以上を推奨
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	pulseaudio 0.9.21 以上を推奨

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

本モジュールのソースは SVN リポジトリから取得可能です。URL は以下の通り。

ソース一式	http://openhri.net/svn/OpenHRIAudio/
-------	---

※ その他詳細情報は OpenHRI ホームページ(<http://openhri.net/>)を参照。

4.1.23. SoundPlayer

1. 機能概要

音声ファイル(WAV)の再生を行う。

2. 基本情報

2.1. 動作環境

種別	RTC
ベンダ名	RTC 再利用技術研究センター
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C++
依存ライブラリ	sox

2.2. 動作条件

実行周期	デフォルト(1000Hz)
------	---------------

2.3. 格納場所

ソース一式	RemotePC/src/HumanRobotInteraction/SoundPlayer
-------	--

3. ポート情報

- データポート(InPort)
なし

- データポート(OutPort)
なし

- サービスポート(Provider)

項目	値
サービスポート名	PlaySound
インターフェイス名	PlaySound
サービスの型	PlaySound

- サービスポート(Consumer)
なし

- コンフィグレーション(Configuration set)

名称	型	デフォルト値	説明
sound_fname	string	sound.wav	再生する音声ファイル(WAV)名

4. サービス関数定義(PlaySound)

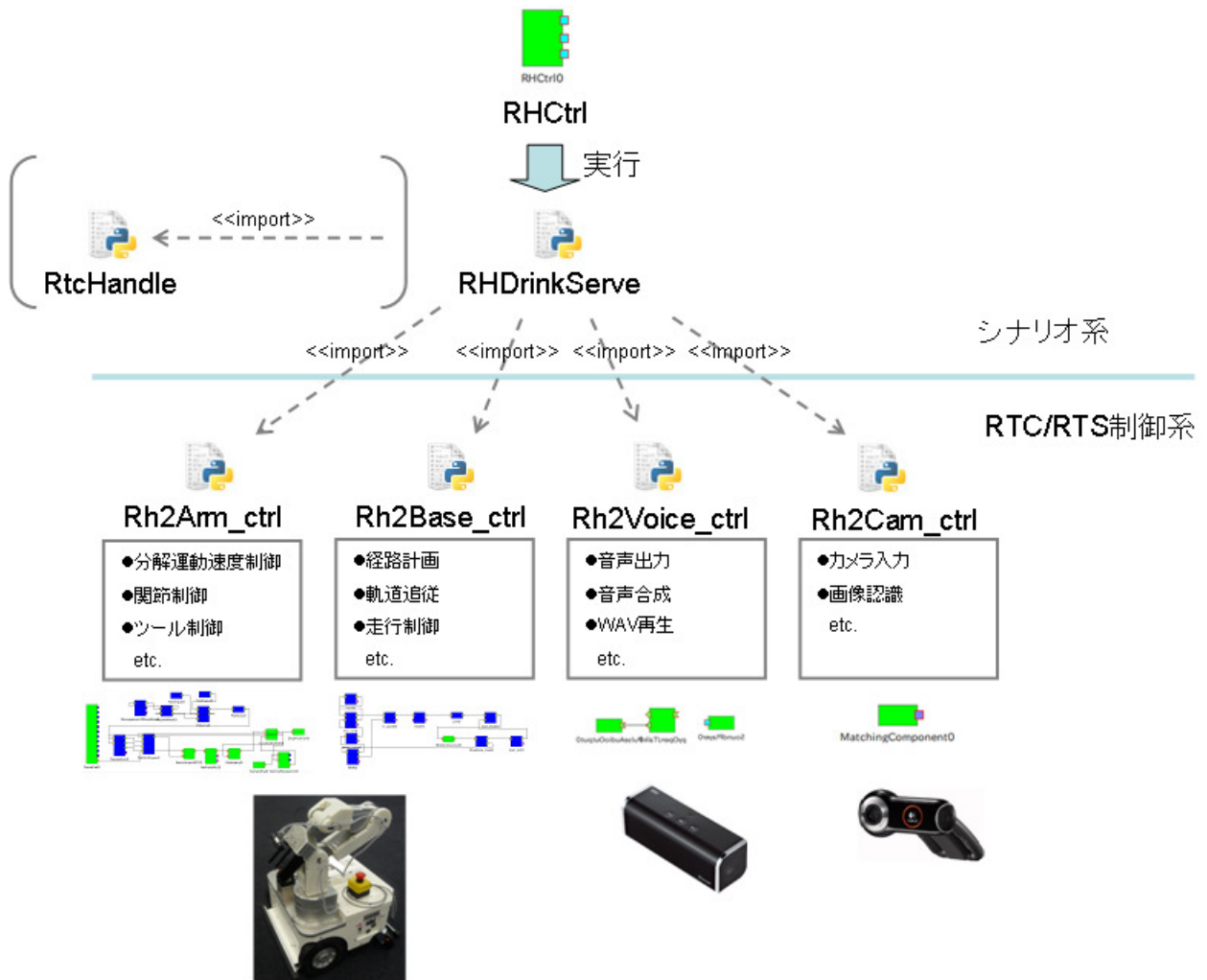
・Play

関数名	Play			
引数	名称	型	I/O	説明
	なし			
戻り値	値			説明
	void			-
説明	コンフィグレーションに設定されたファイル(sound_fname)を再生			
備考				

4.2. スクリプト仕様

本システムでは多数の RTC を制御する仕組みとして RtcHandle¹を用いた Python スクリプトを使用している。動作の仕組みは以下の図を参照のこと。

図.スクリプトの仕組み(RHDrinkServe の例)



¹ RTC を Python 環境から簡単に扱うことができる Python モジュール。参考 URL: “RtcHandle – 使い方とそれを用いた RTC 利用環境の構築” (http://staff.aist.go.jp/t.suehiro/rtm/rtc_handle.html)

今回使用するシナリオ系及び RTC/RTS 制御系スクリプトは以下の通り。

4.2.1. シナリオ系

単独での実行も可能であるが、基本的に RH 制御 RTC(RHCtrl)内で実行する。

1. RHDrinkServe (python)

◆ 機能概要

RH による給仕指示受付のシナリオ動作を行う。

2. RHDrinkTransport (python)

◆ 機能概要

RH による飲み物搭載～給仕終了のシナリオ動作を行う。

3. RHReturn (python)

◆ 機能概要

RH による給仕終了のシナリオ動作を行う。

4. RHGarbageColl (python)

◆ 機能概要

RH による廃棄指示～空き缶回収のシナリオ動作を行う。

5. RHGarbageDump (python)

◆ 機能概要

RH による空き缶廃棄のシナリオ動作を行う。

4.2.2. RTC/RTS 制御系

単独で実行も可能であるが、基本的にシナリオ系スクリプト内で実行する。

6. Rh2Base

6-1.Rh2Base_run (python)

◆ 機能概要

移動知能系 RTC の一括起動、自動接続、活性化／非活性化を行う。

◆ 関数仕様

関数名	引数	機能
run_components	なし	コンポーネント起動
assign_variables	Rtm 環境	RtcHandle の生成
make_connectors	Rtm 環境	接続する Port を登録
connect_all	Rtm 環境	全ての Connector を接続
disconnect_all	Rtm 環境	全ての Connector を切断
connect_list	Rtm 環境,Connector List	指定された Connector を接続
disconnect_list	Rtm 環境,Connector List	指定された Connector を切断
set_config_param	なし	コンフィグレーションを設定
activate_local	なし	自己位置推定関連の RTC の活性化
deactivate_local	なし	自己位置推定関連の RTC の非活性化
activate_all	なし	全 RTC の活性化
deactivate_all	なし	全 RTC の活性化

6-2.Rh2Base_ctrl (python)

◆ 機能概要

移動知能系 RTC に関してシナリオ動作のためのスクリプト(本章 1～5)にて使用するための I/F を提供する。

◆ 関数仕様

関数名	引数	機能
assign_variables	Rtm 環境	RtcHandle の生成
make_connectors	Rtm 環境	接続する Port を登録
connect_all	Rtm 環境	全ての Connector を接続
disconnect_all	Rtm 環境	全ての Connector を切断
connect_list	Rtm 環境,Connector List	指定された Connector を接続
disconnect_list	Rtm 環境,Connector List	指定された Connector を切断
set_config_param	なし	コンフィグレーションを設定
activate_local	なし	自己位置推定関連の RTC の活性化
deactivate_local	なし	自己位置推定関連の RTC の非活性化
activate_all	なし	全 RTC の活性化
deactivate_all	なし	全 RTC の活性化
moveto	START 位置,GOAL 位置	経路走行指示。位置は下記の 5 箇所の中から指定。 [start, entrance, room, drinkserve, dumpsite]
stop	なし	停止指示

7. Rh2Arm

7-1.Rh2Arm_run (python)

◆ 機能概要

作業知能系アーム制御 RTC の一括起動、自動接続、活性化／非活性化を行う。

◆ 関数仕様

関数名	引数	機能
run_components	なし	コンポーネント起動
assign_variables	Rtm 環境	RtcHandle の生成
make_connectors	Rtm 環境	接続する Port を登録
connect_components	Rtm 環境	全ての Connector を接続
disconnect_all	Rtm 環境	全ての Connector を切断
connect_list	Rtm 環境,Connector List	指定された Connector を接続
disconnect_list	Rtm 環境,Connector List	指定された Connector を切断
set_config_param	なし	コンフィグレーションを設定
activate_components	なし	全 RTC の活性化
deactivate_components	なし	全 RTC の活性化
mode_joint	なし	関節角度制御モード
mode_rmrc	なし	分解運動速度制御モード

7-2.Rh2Arm_ctrl (python)

◆ 機能概要

作業知能系アーム制御 RTC に関してシナリオ動作のためのスクリプト(本章 1～5)にて使用するための I/F を提供する。

◆ 関数仕様

関数名	引数	機能
assign_variables	Rtm 環境	RtcHandle の生成
make_connectors	Rtm 環境	接続する Port を登録
connect_components	Rtm 環境	全ての Connector を接続
disconnect_all	Rtm 環境	全ての Connector を切断
connect_list	Rtm 環境,Connector List	指定された Connector を接続
disconnect_list	Rtm 環境,Connector List	指定された Connector を切断
set_config_param	なし	コンフィグレーションを設定
activate_components	なし	全 RTC の活性化
deactivate_components	なし	全 RTC の活性化
mode_joint	なし	関節角度制御モード
mode_rmrc	なし	分解運動速度制御モード
demo	move の RtcHandle インスタンス,目標値のリスト	PTP 動作連続指令
go_to	move の RtcHandle インスタンス,目標値	PTP 動作指令
set_tool	ツール長	tool 長の設定
ready	0:待機姿勢、1:レディー姿勢	アーム走行可能姿勢へ移動
gripper_open	なし	グリッパーを開く
gripper_close	なし	グリッパーを閉じる
gripper_stop	なし	グリッパーを止める
get_cond	なし	グリッパーのコンディション取得
get_mode	なし	グリッパーのモード取得
holder_pickup	缶の位置,グリッパーの状態	ホルダーからの pickup
offset	座標、offset 座標	把持位置の調節
trush_clean	廃棄する缶、グリッパーの状態	ゴミ捨て

8. Re2Cam

8-1.Rh2Cam_run (shell)

◆ 機能概要

作業知能系画像認識 RTC の一括起動、活性化を行う。

8-2.Rh2Cam_end (shell)

◆ 機能概要

作業知能系画像認識 RTC の非活性化、終了を行う。

8-3.Rh2Cam_ctrl (python)

◆ 機能概要

作業知能系画像認識 RTC に関してシナリオ動作のためのスクリプト(本章 1～5)にて使用するための I/F を提供する。

◆ 関数仕様

関数名	引数	機能
assign_variables	Rtm 環境	RtcHandle の生成
get_mark	なし	マークの認識結果を取得

9. Rh2Voice

9-1.Rh2Voice_run1 (shell)

◆ 機能概要

コミュニケーション知能系 RTC のうち pyOpenJTalk,PulseAudioOutput の一括起動、自動接続、活性化を行う。

9-2.Rh2Voice_run2 (shell)

◆ 機能概要

コミュニケーション知能系 RTC のうち SoundPlayer の一括起動、活性化を行う。

9-3.Rh2Voice_run2 (shell)

◆ 機能概要

コミュニケーション知能系 RTC の非活性化、終了を行う。

9-4.Rh2Voice_ctrl (pyrthon)

◆機能概要

コミュニケーション知能系 RTC に関してシナリオ動作のためのスクリプト(本章 1～5)にて使用するための I/F を提供する。

◆ 関数仕様

関数名	引数	機能
assign_variables	Rtm 環境	RtcHandle の生成
activate	なし	全 RTC の活性化
deactivate	なし	全 RTC の活性化
talk	話す内容	発話をおこなう
beep	なし	ビーコン音を鳴らす

4.2.3. その他

単独での実行を想定したメンテナンス用のスクリプト。

10. RHArmReady (python)

◆ 機能概要

RH アームを走行姿勢に移行する。

11. RHArmStandUp (python)

◆ 機能概要

RH アームを直立姿勢に移行する。

12. RHBaseStop (python)

◆ 機能概要

RH の運動機能(台車のみ)を非活性化する。

13. RHStop (python)

◆ 機能概要

RH の運動機能(アーム、台車)を非活性化する。

14. RHRestart (python)

◆ 機能概要

RH の運動機能(アーム、台車)を活性化する。

5. 画面仕様

T.B.D

6. データ仕様

T.B.D

7. その他

7.1. 延期要求

- 画面仕様については RS004 仕様整理を行い、策定する。
- データ仕様については RS003 にて仕様整理を行い、策定する。
- 異常系及び準異常系の動作シーケンスについては RS003 にて対応する。

7.2. その他の要件

特になし。

7.3. 特記事項

本書をご利用される場合には、以下の記載事項・条件にご同意いただいたものとします。

- 本書は独立行政法人 新エネルギー・産業技術総合開発機構の「次世代ロボット知能化技術開発プロジェクト」内実施者向けに評価を目的として提供するものであり、商用利用など他の目的で使用することを禁じます。
- 本書に情報を掲載する際には万全を期していますが、それらの情報の正確性またはお客様にとっての有用性等については一切保証いたしません。
- 利用者が本書を利用することにより生じたいかなる損害についても一切責任を負いません。
- 本書の変更、削除等は、原則として利用者への予告なしに行います。また、止むを得ない事由により公開を中断あるいは中止させていただくことがあります。
- 本書の情報の変更、削除、公開の中断、中止により、利用者に生じたいかなる損害についても一切責任を負いません。

【連絡先】

RTC 再利用技術研究センター

〒101-0021 東京都千代田区外神田 1-18-13 秋葉原ダイビル 1303 号室

Tel/Fax:03-3256-6353 E-Mail:contact@rtc-center.jp