

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
(RS003)

PA10 詳細設計書

1.0 版

2011 年 6 月 30 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. システム配置図	2
2.1.2. b. モジュール構成図	3
2.1.3. c. フォルダ構成	4
3. 動作シーケンス	6
4. モジュール仕様	10
4.1.1. RTC 仕様	10
4.1.1.1. PA10SystemController	10
4.1.1.2. HandCtrl	13
4.1.1.3. pa10vel	15
4.1.1.4. ACT_PA10_RH	18
4.1.1.5. RMRC	22
4.1.2. RTS 仕様	27
4.1.2.1. OpenVGR	27
4.1.3. スクリプト仕様	30
4.1.3.1. キャリブレーション用 PA10 動作スクリプト (VGRcalib)	30
4.1.3.2. PA10 給仕シナリオスクリプト (PA10DrinkServe)	30
4.1.3.3. PA10 制御スクリプト (pa10act_ctrl、pa10sim_ctrl)	31
4.1.3.4. ハンド制御スクリプト (hand_ctrl)	32
4.1.3.5. 飲み物認識スクリプト (VGRdrink_ctrl)	33
4.1.3.6. ホルダ認識スクリプト (VGRholder_ctrl)	34
5. その他	35
5.1. その他の要件	35
5.2. 特記事項	35

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に関する PA10 のシステムや機能などの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(2010/10/01)
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	RTC	RTコンポーネント
10	RTS	RTシステム
11	OSS	オープンソースソフトウェア
12	障害物	人及び、人が一人で運ぶ事の出来る物体
13	RH	リファレンスハードウェア
14	PA10システム	PA10、RH707および作業対象物認識モジュール群を用いたハンドアイシステム

1.4. 参考資料

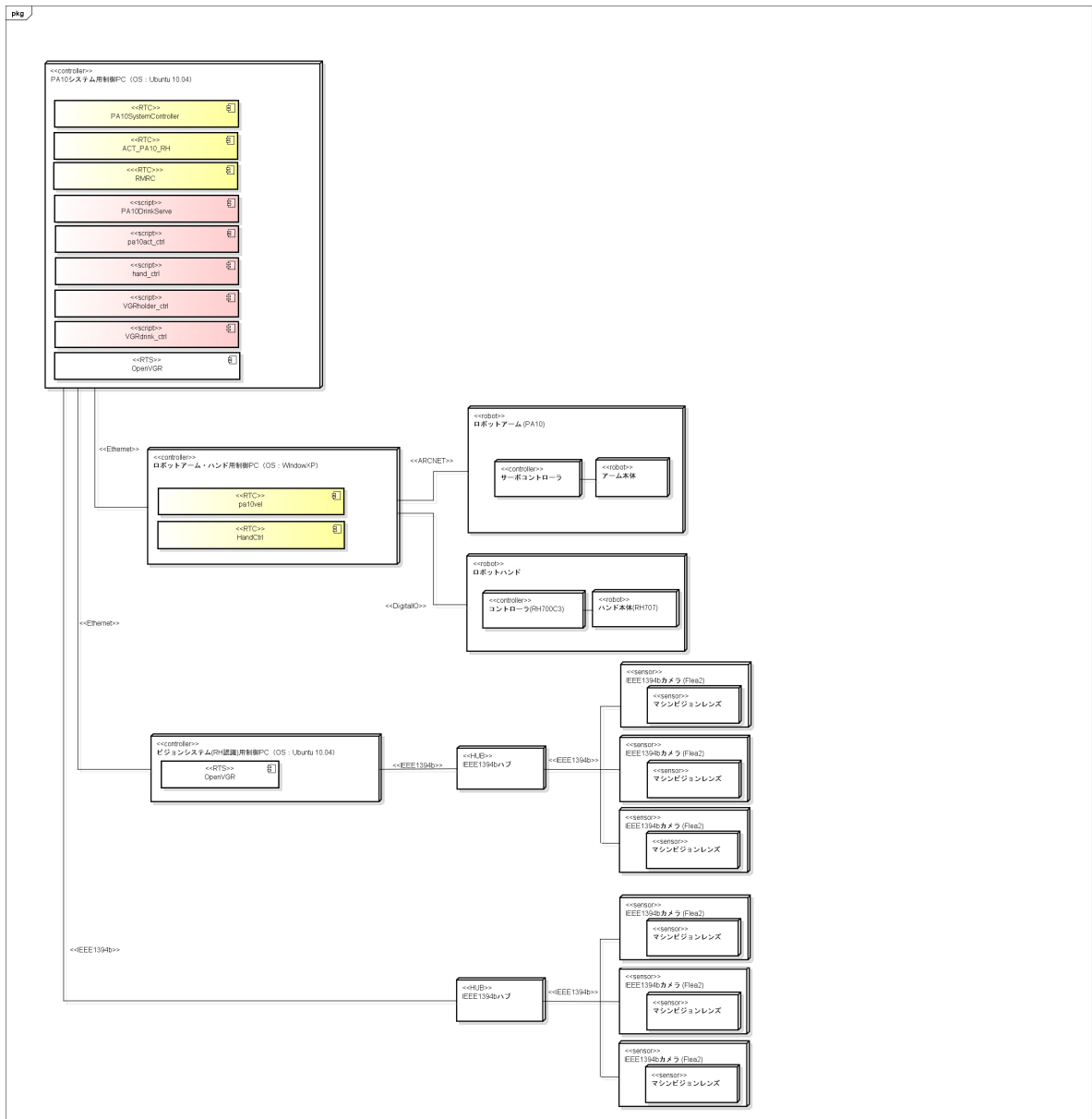
本書を作成するにあたり参照した文書・資料を下表に示す。

表.参考資料一覧

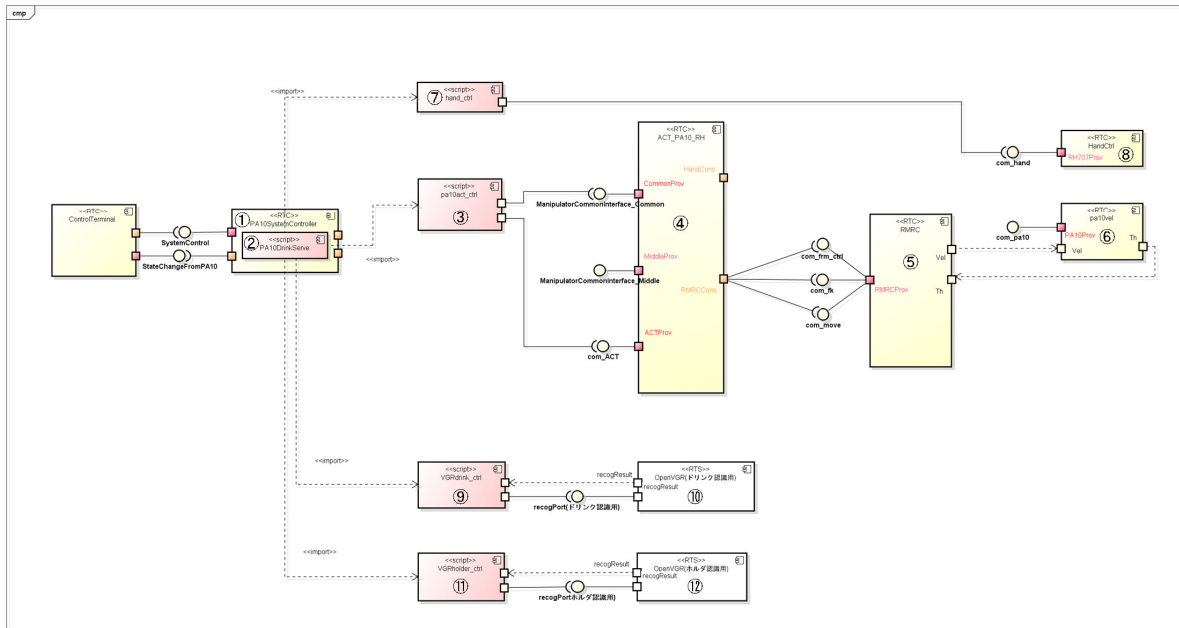
No.	文書名	備考 / URL
1	OpenRTM-aist Official Website	http://www.is.aist.go.jp/rt/OpenRTM-aist/
2	機能仕様書 PA10システム制御モジュール	RTC再利用技術研究センターHP→再利用検証成果公開ページ→統合検証 成果報告 (RS003) 詳細ページ→参考資料 http://210.154.184.16/RS003/RS003.html
3	機能仕様書 RH707制御モジュール	
4	機能仕様書 PA10制御モジュール	
5	機能仕様書 分解運動速度制御モジュール	
6	機能仕様書「作業対象認識モジュール群」(産総研)	
7	作業対象認識モジュールドキュメント (産総研)	
8	作業対象認識モジュールドキュメント 座標系変換ツール(産総研)	

2. システム構成

2.1.1. a. システム配置図



2.1.2. b. モジュール構成図



※ControlTerminal RTC は PA10 システムの外部とする。

表:PA10 システムモジュール一覧

No.	モジュール名	種別	機能	概要
①	PA10SystemController	RTC	給仕処理	PA10 システム制御
②	PA10DrinkServe	script		PA10 給仕シナリオ実行スクリプト
③	pa10act_ctrl	script	PA10 制御	PA10 制御スクリプト
④	ACT_PA10_RH	RTC	ハンド制御	ACT（中レベル）共通インターフェース提供 RTC
⑤	RMRC	RTC		PA10 分解運動速度制御 RTC
⑥	pa10vel	RTC		PA10 制御 RTC
⑦	hand_ctrl	script		RH707 制御スクリプト
⑧	HandCtrl	RTC		RH707 制御 RTC
⑨	vGRdrink_ctrl	script	飲み物認識	飲み物認識スクリプト
⑩	OpenVGR	RTS		作業対象認識モジュール群
⑪	VGRholder_ctrl	script	ホルダ認識	ホルダ認識スクリプト
⑫	OpenVGR	RTS		作業対象認識モジュール群

2.1.3. c. フォルダ構成

各制御 PC におけるモジュールのフォルダ構成は以下の通りとする。

● PA10 システム用制御 PC

ディレクトリ	内容	備考
~/opt/OpenVGR	作業対象認識モジュール	-
~/PCforSystem Control	PA10 システム制御コンポーネント	-
└ bin	-	-
├ comp	各 RTC のバイナリファイル	-
├ script	各種制御スクリプト	-
└ so	各 RTC(C++実装のみ)の so ファイル	-
└ etc	各 RTC の conf ファイル	-
└ idl	各 RTC の idl ファイル	-
└ include	-	-
└ lib	rtc_handle	-
└ log	各 RTC ログ出力ファイル	-
└ src	ソースファイル	-
└ GeneralController	統括制御	-
└ PA10SystemController	PA10 システム制御コンポーネント	-
└ MotionController	動作制御	-
└ ACT_PA10_RH	ACT (中レベル) 共通インターフェース提供コンポーネント	-
└ RMRC	分解運動速度制御コンポーネント	-
└ pa10disp_py	PA10 幾何モデル描画コンポーネント	-
└ pa10hand_disp	ハンド付き PA10 幾何モデル描画コンポーネント	-
└ slider	データ送出コンポーネント	-
└ tools	-	-
└ geo	幾何学演算ライブラリ	-
└ nr	行列演算ライブラリ	-
└ v_robot	PA10 シミュレータ描画処理	-
└ vel_sim	PA10 シミュレータコンポーネント	-

● ロボットアーム・ハンド用制御 PC

ディレクトリ	内容	備考
C:\device	-	-
└ HandCtrl	RH707 制御コンポーネント	-
└ pal0vel	PA10 制御コンポーネント	

● ビジョンシステム(ホルダ認識)用制御 PC

ディレクトリ	内容	備考
~/opt/OpenVGR	作業対象認識モジュール	-

3. 動作シーケンス

図：給仕シーケンス

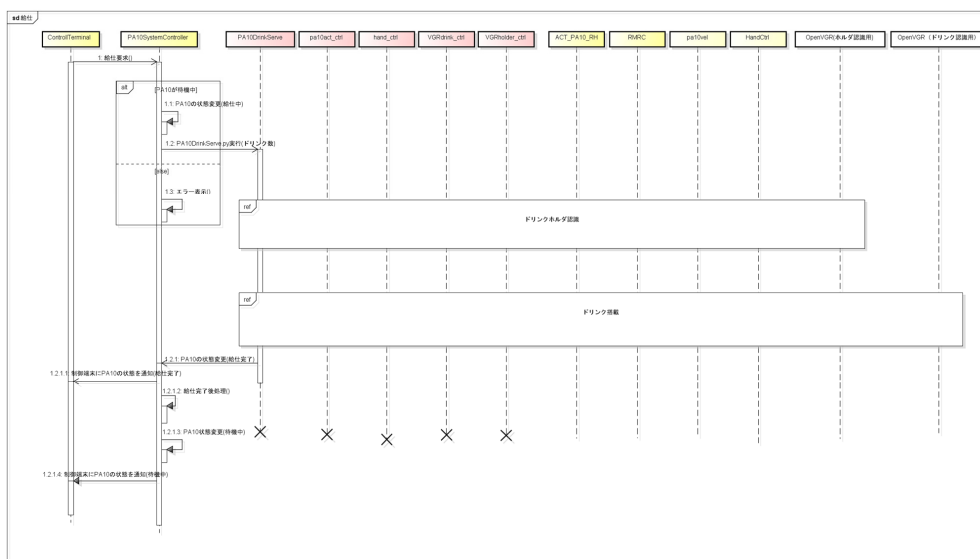


図:ドリンクホルダ認識シーケンス

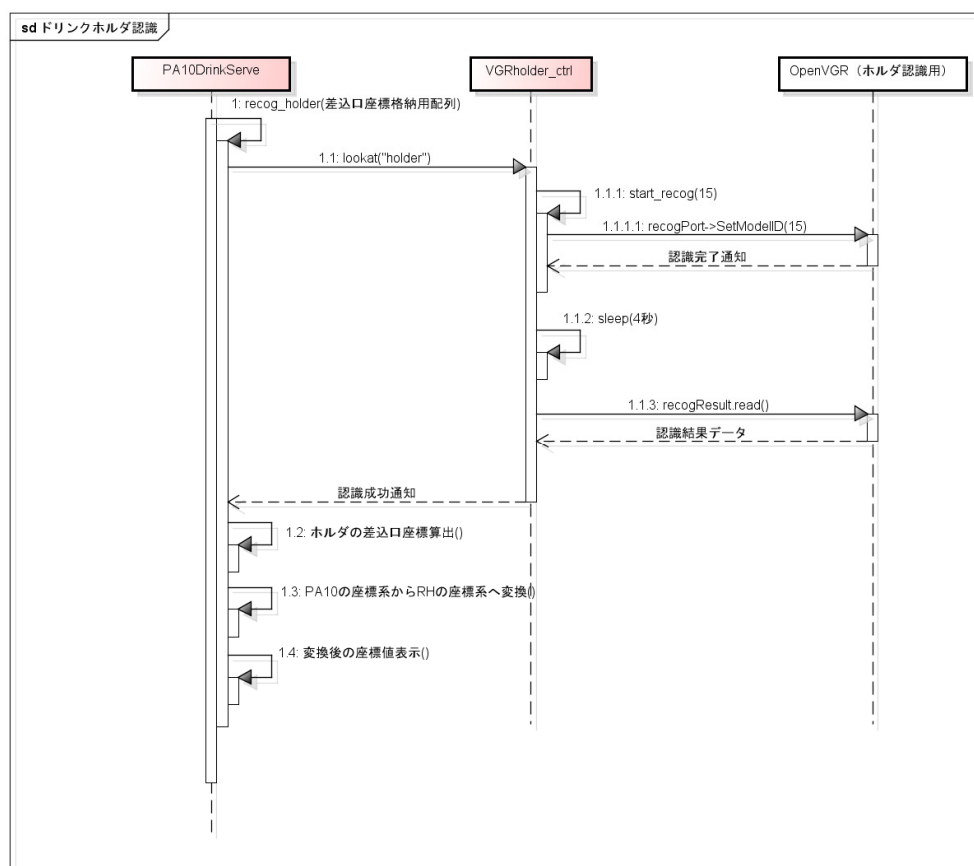


図:ドリンク搭載シーケンス

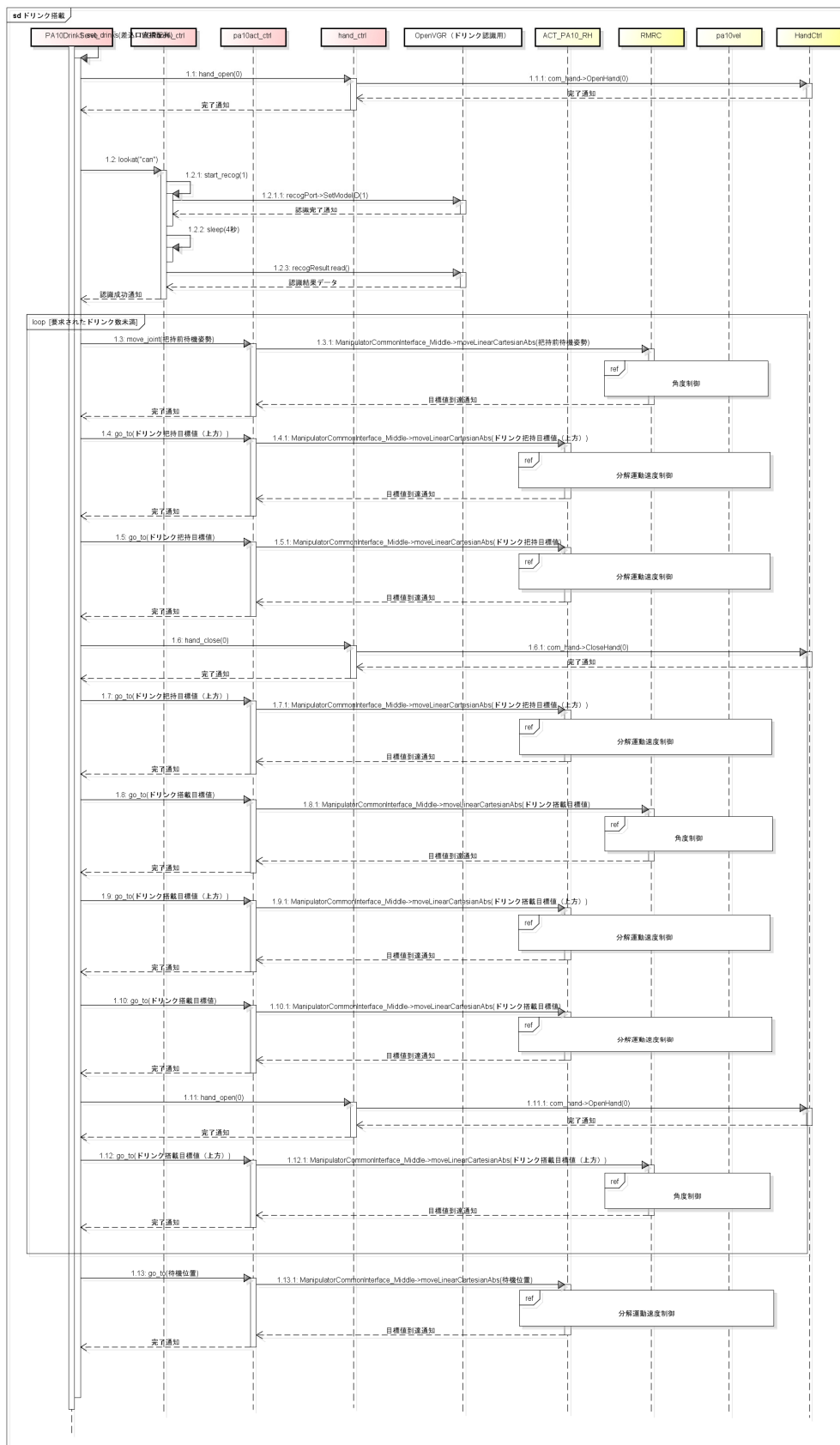


図: 分解運動速度制御シーケンス

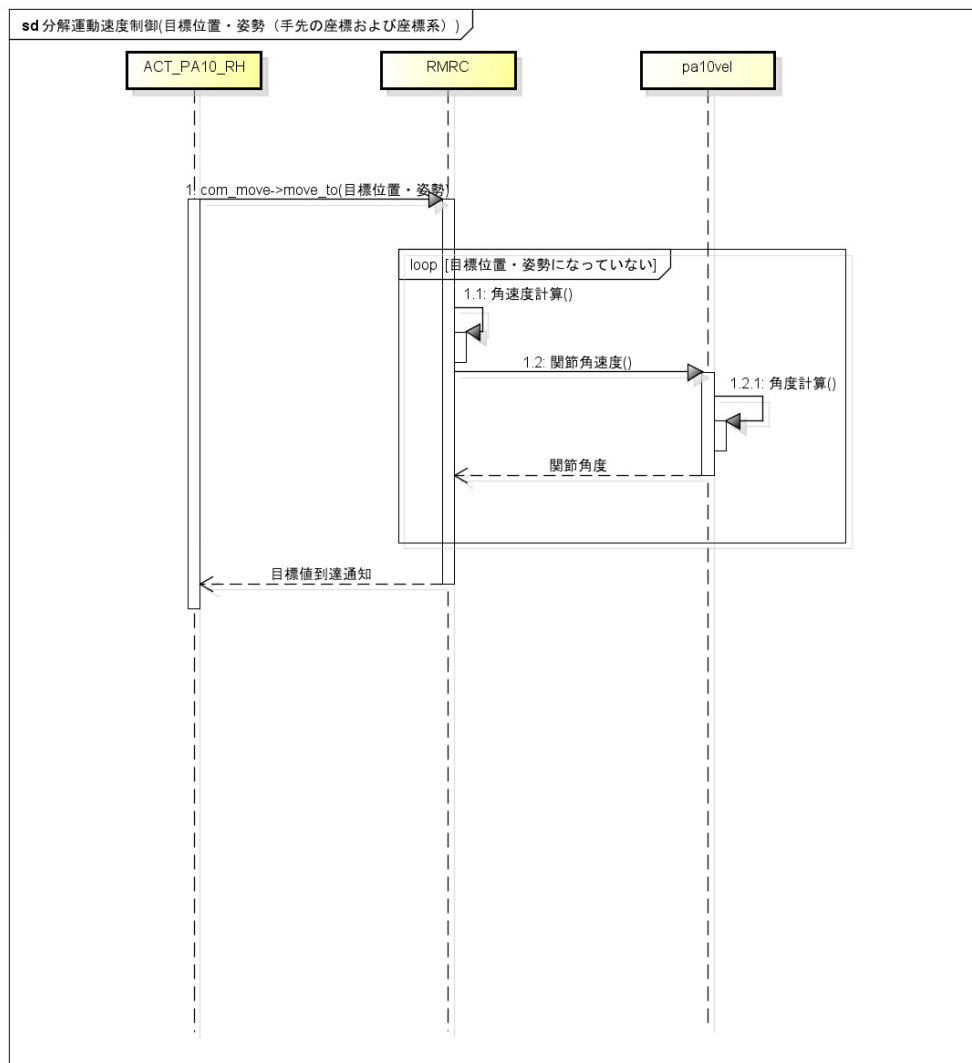
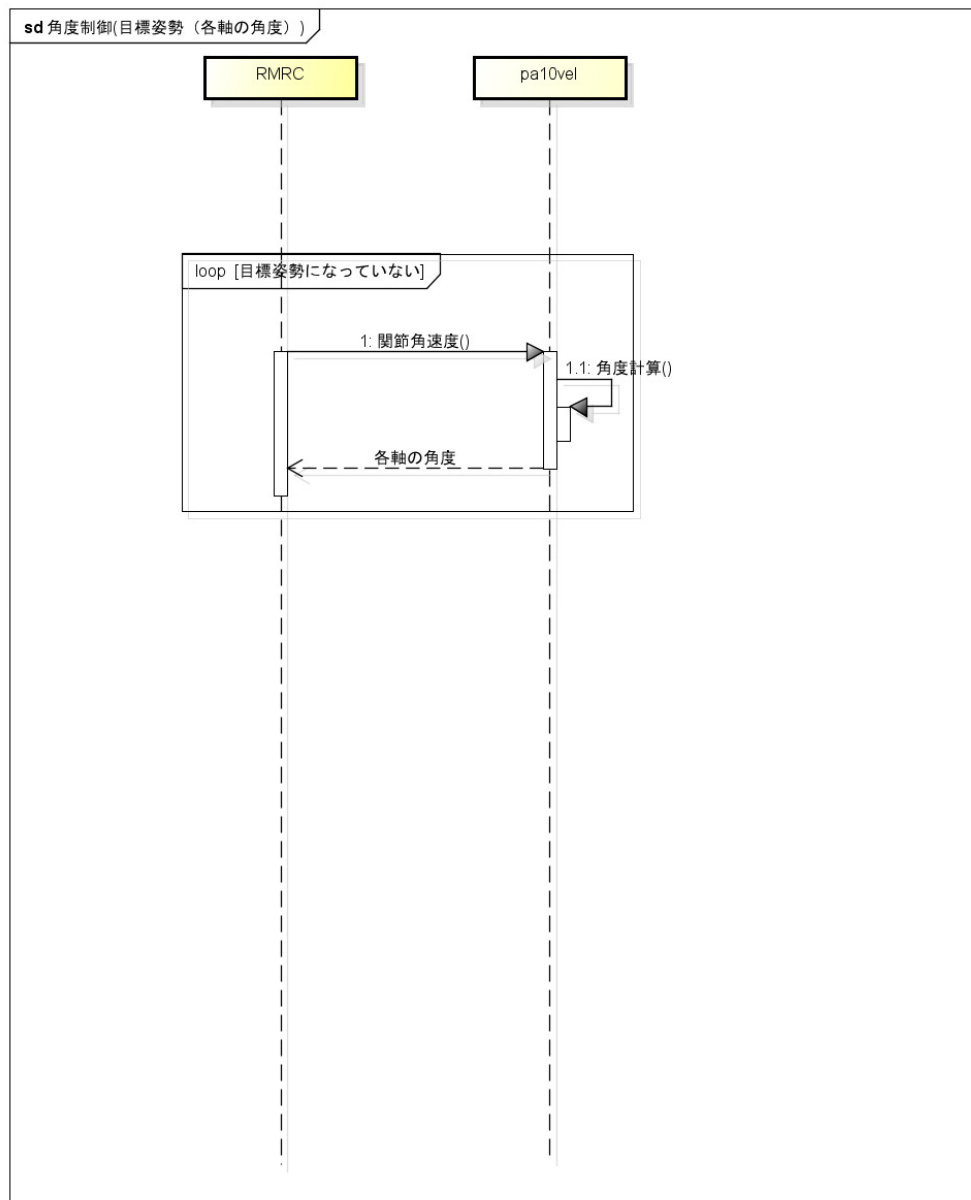


図: 角度制御シーケンス



4. モジュール仕様

4.1.1. RTC 仕様

4.1.1.1. PA10SystemController

1. 機能概要

PA10 システムの状態管理を行う。また、制御端末より指示をうけた場合には PA10 システムヘッドリンク給仕の指示を出す。

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	g++ 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	PCforSystemController/src/GeneralController/PA10SystemController
-------	--

3. ポート情報

サービスポート(prov)名	インターフェース名	説明
PA10SysCtrlrProv	SystemControl	PA10 システムの制御
サービスポート(cons)名	インターフェース名	説明
ControlTerminalCons	StateChangeFromPA10	制御端末へ PA10 の状態通知

4. サービスポート I/F 仕様

SystemControl

4.1 PA10GetStatus

関数名	PA10GetStatus			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	値			説明
	0			待機中
	1			起動中
	2			給仕中
	3			システム終了
	4			給仕完了
説明	PA10 の状態を取得する。			

4.2 PA10SetStatus

関数名	PA10SetStatus			
引数	名称	型	I/O	説明
	state	long	入力	PA10 の状態
戻り値	値			説明
	なし			-
説明	PA10 の状態を設定する。 PA10 の状態 0 : 待機中 1 : 起動中 2 : 給仕中 3 : 終了 4 : 給仕完了			

4.3 PA10End

関数名	PA10End			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	値			説明
	なし			-
説明	PA10 システムを終了させる。			

4.4 PA10DrinkServeReq

関数名	PA10DrinkServeReq			
引数	名称	型	I/O	説明
	drinknum	short	入力	給仕するドリンクの数
戻り値	値			説明
	なし			-
説明	<p>PA10 に引数指定された数のドリンクを給仕するよう要求し、給仕処理の状態を ControlTerminal へ通知する。現在はサービスポート内で PA10DrinkServe スクリプトを実行しているのみである。</p> <p>給仕処理の状態</p> <p>OK: 給仕の正常終了</p> <p>RH_LOCATION_ERROR: RH の到着位置が不適</p> <p>HOLDER_NOT_EMPTY: ドリンクホルダーが空でない</p> <p>LACK_OF_DRINK: ドリンクの在庫不足</p>			

4.5 PA10ReportError

関数名	PA10ReportError			
引数	名称	型	I/O	説明
	errnum	long	入力	エラー番号
戻り値	値			説明
	なし			-
説明	<p>エラーリスト</p> <p>0: 異常なし</p> <p>1: システム内 RTC が全て起動していません ※実機制御 RTC を除く</p> <p>2: 給仕シナリオ実行スレッド終了に失敗しました</p> <p>3: PA10 の状態が未定義の状態です</p> <p>4: 実機エラーが発生しました</p>			
説明	PA10 システムにおけるエラー設定をする。			

詳細に関しては参考資料2を参照

4.1.1.2. HandCtrl

1. 機能概要

シュンク・ジャパン社製電動式平行ハンド RH707 の制御を行う。

ハンドの制御は、ハンドを開く／ゆっくり開く／閉じる／ゆっくり閉じる／開口動作を停止させる の 5種類である。

2. 基本情報

2.1. 動作環境

動作 OS	Windows XP
開発言語	C++
コンパイラ	Microsoft Visual C++ 2008 Express Edition
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE(C++版)
依存パッケージ(OpenRTM)	OmniORB-4.1.4
依存パッケージ(その他)	コンテック社製 API 関数ライブラリ集 API-PAC(W32)*

※このライブラリはコンテック社製デジタル入出力ボードに付属するものである。

2.2. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト (1000Hz)

2.3. 格納場所

ソース式	device/HandCtrl
------	-----------------

3. ポート情報

サービスポート(Prov)

サービスポート名	インターフェース名	説明
RH707Prov	com_hand	ハンドの各種制御

4. サービスポート I/F 仕様

com_hand

4.1 InitHand

関数名	InitHand			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	ハンドの初期化(DIO オープン/クローズ、ハンドの開口)をする。			

4.2 OpenHand

関数名	OpenHand			
引数	名称	型	I/O	説明
	slow	short	入力	スロー指定 1=ゆっくり開く 0=通常の速さで開く
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	ハンドを開く。			

4.3 CloseHand

関数名	CloseHand			
引数	名称	型	I/O	説明
	slow	short	入力	スロー指定 1=ゆっくり閉じる 0=通常の速さで閉じる
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	ハンドを閉じる。			

4.4 GetHandCond

関数名	GetHandCond			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	short			ハンドの状態 1=ハンドの動作中 2=ハンドのクローズ完了 3=ハンドのオープン完了 4=ハンドのストップ実行中
説明	ハンド状態を取得する。			

4.5 StopHand

関数名	StopHand			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	ハンド開閉動作をストップする。			

詳細に関しては参考資料3を参照

4.1.1.3. pa10vel

1. 機能概要

本コンポーネントは三菱重工業製汎用ロボット PA10 の関節角速度制御を行う。

2. 基本情報

2.1. 動作環境

動作 OS	Windows XP
開発言語	C++
コンパイラ	Microsoft Visual C++ 2008 Express Edition
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE(C++版)
依存ライブラリ(OpenRTM)	OmniORB-4.1.4
依存ライブラリ(その他)	三菱重工業製 PA ライブラリ(商用)

2.2. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト(1000Hz)

2.3. 格納場所

ソース一式	device/pa10vel
-------	----------------

3. ポート情報

データポート(Inport)

名称	型	値の範囲	説明
vel	TimedDoubleSeq	[※1]	各軸（7 軸）の動作角速度 (単位: rad/sec)

[※1] PA10 の制御パラメータ情報より速度リミットを取得し、判断を行っている。リミット値のデフォルトの設定はアーム本体仕様を参照のこと。

データポート(Outport)

名称	型	値の範囲	説明
th	TimedDoubleSeq	[※2]	現在の各軸（7 軸）角度 (単位: rad)

[※2] PA10 の制御パラメータ情報より各軸の角度情報を取得し、出力パラメータとしている。リミット値のデフォルトの設定はアーム本体仕様を参照のこと。

サービスポート(Prov)

サービスポート名	インターフェース名	説明
PA10Prov	com_pa10	PA10 の各種動作制御

4. サービスポート I/F 仕様

4.1 move_here_pa10

関数名	move_here_pa10			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	long			O:正常終了 その他:異常終了
説明	PA10 実機の現在の各軸角度を取得後、その角度へ各軸角度制御を行う。これにより PA10 実機の各軸角度とコンポーネントが保持する現在角度との同期をとることが可能である。			
備考	本 API は reset_pa10 で呼び出される			

4.2 reset_pa10

関数名	reset_pa10			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	long			O:正常終了 その他:異常終了
説明	PA ライブラリの初期化及びアームのリセット処理を行う。			
備考	本 API は init_pa10 で呼び出される。			

4.3 init_pa10

関数名	init_pa10			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	long			O:正常終了 その他:異常終了
説明	待機姿勢として指定された角度へ各軸制御を行う。			

4.4 on_pa10

関数名	on_pa10			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	long			O:正常終了 その他:異常終了
説明	PA10 の各軸速度制御の開始処理を行う。			

4.5 off_pa10

関数名	off_pa10			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	long			0:正常終了 その他:異常終了
説明	PA10 の各軸速度制御の一時停止処理を行う。			

4.6 end_pa10

関数名	end_pa10			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	long			0:正常終了 その他:異常終了
説明	PA ライブラリの終了及びアームのクローズ処理を行う。			

4.7 set_flg

関数名	set_flg			
引数	名称	型	I/O	説明
	flg	long	入力	割り込みフラグ 1: ON 各軸角速度制御を中断する 2: OFF 各軸角速度制御を再開する
戻り値	型			説明
	なし			-
説明	各軸角速度制御への割り込みフラグを設定する。			
備考	本コーポメントがアクティブ状態であるときは、必ず本フラグの設定をおこなうこと。実装例は下記の通りである。			

詳細に関しては参考資料4を参照

4.1.1.4. ACT_PA10_RH

1. 機能概要

本コンポーネントは NEDO 知能化プロジェクト作業の作業サブ WG において定められている中レベル ACT RTC の共通インターフェースを提供する。

2. 基本情報

2.1. 動作環境

動作 OS	Ubuntu10.04
開発言語	C++
コンパイラ	g++4.4.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE(C++版)
依存パッケージ(OpenRTM)	OmniORB-4.1.2-1
依存パッケージ(その他)	行列演算ライブラリ

2.2. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト(1000Hz)

2.3. 格納場所

ソース式	PCforSystemControl/src/MotionController/ACT_PA10_RH
------	---

3. ポート情報

サービスポート(Prov)

サービスポート名	インターフェース名	説明
CommonProv	ManipulatorCommonInterface_Common	共通コマンド用サービスポート
MiddleProv	ManipulatorCommonInterface_Middle	モーションコマンド用サービスポート
ACTProv	com_ACT	目標手先位置・姿勢データを作成する。

サービスポート(Cons)

サービスポート名	インターフェース名	説明
HandCons	com_hand	ハンドの各種制御
RMRCCons	com_frm_ctrl	手先位置・姿勢の軌跡制御
RMRCCons	com_fk	順運動学計算
RMRCCons	com_move	算出した目標手先位置・姿勢を指定

4. サービスポート I/F 仕様

ManipulatorCommonInterface_Middle

4.1. closeGripper

関数名	closeGripper			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	グリッパを閉じる。 閉口しきっていない場合も一定時間後に動作を完了する。			

4.2. openGripper

関数名	openGripper			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	グリッパを開く。			

4.3. getBaseOffset

関数名	getBaseOffset			
引数	名称	型	I/O	説明
	offset	RTC::HgMatrix	出力	関連文書 2 の 5.3.1 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	マニピュレータの設置位置を取得する。			

4.4. getFeedbackPosCartesian

関数名	getFeedbackPosCartesian			
引数	名称	型	I/O	説明
	pos	RTC::CarPosWithElbow	出力	関連文書 2 の 5.3.2 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	直交座標系の位置フィードバック情報を取得する。			

4.5. getSoftLimitCartesian

関数名	getSoftLimitCartesian			
引数	名称	型	I/O	説明
	xLimit	RTC::LimitValue	出力	関連文書1の 3.1.3 参照
	yLimit	RTC::LimitValue	出力	関連文書1の 3.1.3 参照
	zLimit	RTC::LimitValue	出力	関連文書1の 3.1.3 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	直交座標系のソフトリミット値を取得する。			

4.6. moveLinearCartesianAbs

関数名	moveLinearCartesianAbs			
引数	名称	型	I/O	説明
	carPoint	const RTC::CarPosWithElbow&	入力	関連文書 2 の 5.3.2 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	直交座標の直線補間(絶対指令)をする。			

4.7. movePTPCartesianAbs

関数名	movePTPCartesianAbs			
引数	名称	型	I/O	説明
	jointPoints	const RTC::JointPos&	入力	関連文書1の 3.1.2 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	関節座標の直線補間(直交・絶対指令)をする。			

4.8. setControlPointOffset

関数名	setControlPointOffset			
引数	名称	型	I/O	説明
	offset	const RTC::HgMatrix&	入力	関連文書 2 の 5.3.1 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	制御点のフランジ面からのオフセット量を設定する。			

4.9. setMaxSpeedJoint

関数名	setMaxSpeedJoint			
引数	名称	型	I/O	説明
	speed	const RTC::DoubleSeq&	入力	関連文書1の 3.1.1 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	関節動作時の最大動作速度を設定する。			

4.10. setSoftLimitCartesian

関数名	setSoftLimitCartesian			
引数	名称	型	I/O	説明
	xLimit	const RTC::LimitValue&	入力	関連文書1の 3.1.3 参照
	yLimit	const RTC::LimitValue&	入力	関連文書1の 3.1.3 参照
	zLimit	const RTC::LimitValue&	入力	関連文書1の 3.1.3 参照
戻り値	型			説明
	RTC::RETURN_ID			関連文書1の 3.1.4 参照
説明	直交座標系のソフトリミット値を設定する。			

com_ACT

4.11. createCarPosWithElbow

関数名	createCarPosWithElbow			
引数	名称	型	I/O	説明
	xyz	Vector	入力	目標手先位置 [mm]
	abc	Vector	入力	目標手先回転角度 [degree]
戻り値	型			説明
	RTC::CarPosWithElbow			目標手先位置・姿勢を有した CarPosWithElbow 型のデータ
説明	CarPosWithElbow 型のデータを作成する。			

詳細に関しては参考資料5を参照

4.1.1.5. RMRC

1. 機能概要

ロボットアームの分解運動速度制御を実現する。

本知能モジュールでは、目標とする手先位置・姿勢の指定、動作速度の上限指定、手先に取り付けるツール長とその姿勢の設定を行う機能がサービスとして提供されている。

※尚、可動範囲は PA10 架台取り付け中心部分から半径 850mm の球面内とする。

2. 基本情報

2.1. 動作環境

動作 OS	Ubuntu10.04
開発言語	C++
コンパイラ	g++4.4.3-1
RT ミドルウェア／バージョン	OpenRTM-aist-1.0.0-RELEASE(C++版)
依存パッケージ(OpenRTM)	OmniORB-4.1.2-1
依存パッケージ(その他)	行列演算ライブラリ(自作)
	順運動学計算モジュール

2.2. 動作条件

実行型	PeriodicExecutionContext
実行周期	デフォルト(1000Hz)

2.3. 格納場所

ソース一式	PCforSystemControl/src/MotionController/RMRC
-------	--

3. 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(軸数)	現在の各軸角度情報

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

サービスポート(Prov)

サービスポート名	インターフェース名	説明
RMRCProv	com_frm_ctrl	手先位置・姿勢の軌跡制御
RMRCProv	com_fk	順運動学計算
RMRCProv	com_move	算出した目標手先位置・姿勢を指定

4. コンフィグレーション情報

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)
min_joint_vel	vector<double>	[0.0,0.0,0.0,0.0,0.0,0.0,0.0]	各関節の最小動作角速度 [rad/sec] データフォーマット: [第1軸, 第2軸, … 第7軸]

コンフィグレーションセット

コンフィグレーションセット名	名称	値
PA10	n_size	7
	m_max	100
	min_joint_vel	[0.0,0.0,0.0,0.0,0.0,0.0,0.0]
RefHardArm	n_size	6
	m_max	100
	min_joint_vel	[0.006,0.005,0.012,0.014,0.031,0.001]

5. サービスポート I/F 仕様

com_frm

5.1. set_param

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 [mm/sec]
	k_pr	double	入力	-
戻り値	値			説明
	なし			-
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。			

com_fk

5.2. set_tool

関数名	set_tool			
引数	名称	型	I/O	説明
	frm	const Frame&	入力	ツールの位置[mm]・姿勢
戻り値	型			説明
	short			0:正常終了 その他:異常終了
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。			

5.3. get_cur_frm

関数名	get_cur_frm			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	Frame			現在の手先の位置[mm]・姿勢
説明	現在の手先の位置・姿勢を取得する。			

5.4. get_joint_num

関数名	get_joint_num			
引数	名称	型	I/O	説明
	なし	-	-	-
戻り値	型			説明
	short			軸数
説明	軸数を取得する。			

com_move

5.5. move_to

関数名	move_to			
引数	名称	型	I/O	説明
	frm	const Frame&	入力	手先の目標位置[mm]と姿勢
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	分解運動速度制御のトリガーであり、目標手先位置・姿勢を指定する。			

5.6. set_max_joint_vel

関数名	set_max_joint_vel			
引数	名称	型	I/O	説明
	max_vel	const DoubleSeq&	入力	各軸最大動作角速度[rad/sec]
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	最大動作角速度を設定する。			

5.7. move_joint

関数名	move_joint			
引数	名称	型	I/O	説明
	ref_angles	const DoubleSeq&	入力	各軸の目標角度 [rad]
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	角度制御により各軸を目標角度へ移行させる。			

5.8. get_softlimit

関数名	get_softlimit			
引数	名称	型	I/O	説明
	x_max	double	出力	X軸のソフト上限値 [mm]
	y_max	double	出力	Y軸のソフト上限値 [mm]
	z_max	double	出力	Z軸のソフト上限値 [mm]
	x_min	double	出力	X軸のソフト下限値 [mm]
	y_min	double	出力	Y軸のソフト下限値 [mm]
	z_min	double	出力	Z軸のソフト下限値 [mm]
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	直行座標系におけるソフトリミット(可動範囲)を取得する。			

5.9. set_softlimit

関数名	set_softlimit			
引数	名称	型	I/O	説明
	x_max	double	入力	X軸のソフト上限値 [mm]
	y_max	double	入力	Y軸のソフト上限値 [mm]
	z_max	double	入力	Z軸のソフト上限値 [mm]
	x_min	double	入力	X軸のソフト下限値 [mm]
	y_min	double	入力	Y軸のソフト下限値 [mm]
	z_min	double	入力	Z軸のソフト下限値 [mm]
戻り値	型			説明
	short			0:正常終了 その他:異常終了
説明	直行座標系におけるソフトリミット(可動範囲)を設定する。			

6. 依存モジュール情報

6.1 行列演算ライブラリ

モジュール名	行列演算ライブラリ
ファイル/ヘッダ名	geo_std.o
機能概要	行列演算に関する関数群を定義している。

6.2 順運動学計算モジュール

モジュール名	順運動学計算モジュール
ファイル/ヘッダ名	fk.cpp / fk.h
機能概要	本ファイル内でロボットアームの関節数、リンク長を設定し、その情報に基づきアームの順運動学計算を行う。

現在は本モジュールにおいてロボットアーム固有のパラメータ設定や、順運動学計算の記述がされている。

本モジュールはロボットアーム毎に各自でファイルを用意するものとする。制御するロボットアームを変更する際は、適宜対応するファイルに差し替え、リコンパイルをする。

詳細に関しては参考資料5を参照

4.1.2. RTS 仕様

4.1.2.1. OpenVGR

1. 機能概要

作業対象である飲み物(190mm 缶)やドリンクホルダを認識する。

・飲み物認識結果データは、認識確度が 0.30 以上のものを採用する。

・ドリンクホルダ認識結果データは、認識確度が 0.20 以上のものを採用する。

2. 基本情報

2.1. 動作環境

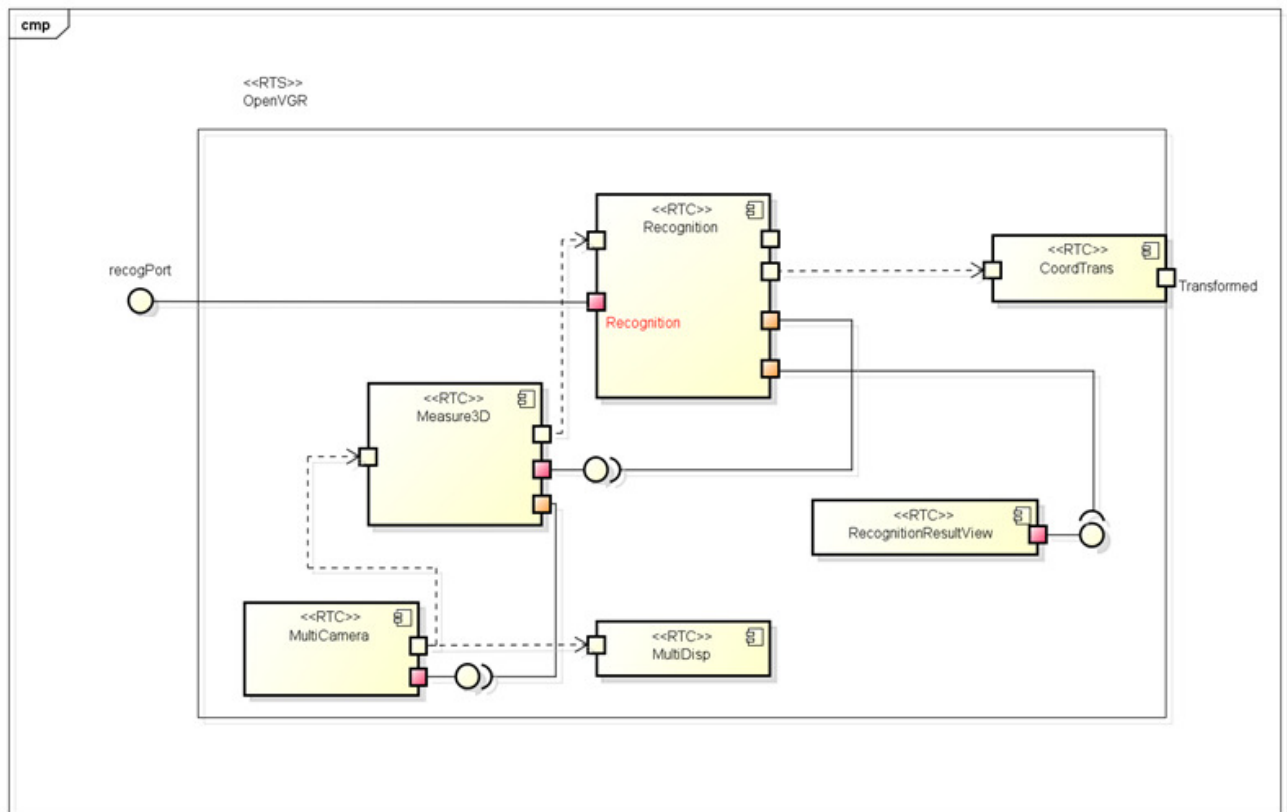
種別	RTS
ベンダ名	産業技術総合研究所
バージョン	1.0.0
動作 OS	Ubuntu10.04LTS
RT ミドルウェア	OpenRTM-aist-1.0.0-RELEASE
開発言語	C,C++
コンパイラ	g++ 4.4.3
依存ライブラリ	OpenCV 2.0 libdc1394-22

2.2. 格納場所

ソース一式	PCforOpenVGR/src
-------	------------------

2.3. OpenVGR 構成

OpenVGR は下図のような構成になっている。



3. ポート情報

PTS 外部に対するポートの情報は以下のようにになっている。

データポート(Outport)

RTC 名	ポート名	型	データ長	説明
CoordTrans	Transformed	TimedDoubleSeq	不定	認識結果データ

サービスポート(prov)

RTC 名	サービス名	インターフェース名	説明
Recognition	RecognitionService	recogPort	認識トリガ入力

4. サービスポート I/F 仕様

recogPort

4.1. setModelID

関数名	setModelID			
引数	名称	型	I/O	説明
	ModelID	Long	入力	認識対象モデル ID
戻り値	値			説明
	なし			-
説明	ID によって指定した物体の 3 次元位置・姿勢を指定し recogResult データポートから共通形式で出力する。			

詳細に関しては参考資料6を参照

4.1.3. スクリプト仕様

4.1.3.1. キャリブレーション用 PA10 動作スクリプト(VGRcalib)

概要: 作業対象認識のためのカメラ座標キャリブレーションに必要な PA10 の動作を行う。

関数仕様:

関数名	引数	機能	使用例
calib	モード 1: 飲み物認識用 2: ドリンクホルダ認識用 3: 終了	キャリブレーション 用 PA10 動作開始	>>>calib(1)

4.1.3.2. PA10 給仕シナリオスクリプト(PA10DrinkServe)

概要: 制御端末からの給仕指示を受け、PA10 システムによる給仕シナリオ動作を行う。

関数仕様:

関数名	引数	機能	使用例
start_handle	Rtm環境のインスタンス	rtc_handleによる RTCのハンドリング	>>>start_handle(env)
activate	なし	PA10RMRC、 pa10vel、HandCtrl、 VVV RTCの活性化	>>>activate()
deactivate_all	なし	PA10RMRC、 pa10vel、HandCtrl、 VVV RTCの非活性化	>>>deactivate_all()
end_all	Rtm環境のインスタンス	各モジュールの終了 (rtc_handleを終了 するのみで、RTC自 体は終了しない)	>>>end_all(env)
recog_holder	座標格納用のリスト	RHのドリンクホルダ を認識し、挿し口 (穴)の座標を渡さ れたリストに格納	>>>hole_list = [] >>>recog_holder(hole_list)
set_drinks	挿し口座標リスト	テーブル上の飲み物 をを認識し、ドリン クホルダにセットす る	>>>set_drinks(hole_list)

4.1.3.3. PA10 制御スクリプト(pa10act_ctrl、pa10sim_ctrl)

- pa10act_ctrl 実機制御用
- pa10sim_ctrl シミュレータ(VPython)上制御用

概要: PA10 制御関連 RTC の自動接続、活性化／非活性化を行い、PA10 制御に関する I/F を提供する。

関数仕様: pa10act_ctrl、pa10sim_ctrl 共通

関数名	引数	機能	使用例
assign_variables	Rtm環境	RtcHandleの変数への割り付け	>>>env = RtmEnv(sys.argv,[PA10_NS]) >>>for ns in env.name_space : >>>env.name_space[ns].list_obj() >>>assign_variables(env)
start_handle	Rtm環境のインスタンス	rtc_handleによるRTCのハンドリング	>>>start_handle(env)
make_connectors	Rtm環境のインスタンス	ポートのコネクタ作成 (接続対象は関数内で直に指定)	>>>make_connectors(env)
disconnect_all	Rtm環境のインスタンス	接続対象ポートの接続解除	>>>disconnect_all(env)
connect_list	Rtm環境のインスタンス、接続対象のリスト	接続対象ポートの接続	>>>connect_list(env,[0,1,2]) ※接続対象リストの0番、1番、2番を指定
activate_seq	RTC名とsleep時間を含むリスト	指定したsleep間隔でのRTCの活性化	>>>activate_seq([pa10vel, RMRC, 3])
deactivate	なし	ACT_PA10_RH、RMRC、pa10vel RTCの非活性化	>>>deactivate()
end	Rtm環境のインスタンス	PA10制御の終了 (rtc_handleを終了するのみで、RTC自体は終了しない)	>>>end(env)
move_joint	目標関節角度	関節角度制御	>>>move_joint(pa10vel, j_ready) または
mode_rmrc	Rtm環境のインスタンス	関節角速度制御モードへ移行	>>>mode_rmrc(env)
get_pos	なし	現在手先位置・姿勢を取得	>>>get_pos()
set_tool	ツールオフセット	ツール長(把持点)の設定	>>>tool = [0.0, 0.0, 150.0] >>>set_tool(tool)
go_to	目標値	指定した目標値へ移行	>>>point1 = [600.0, 0.0, 500.0, 0.0, 180.0, 45.0] # 単位[mm], [degree] >>>go_to(point1)
demo	目標値のリスト	目標値リストへの連続移行	>>>demo([point1,point2,point3])
can_move	目標値	目標値が到達可能な値かどうかの判定	>>>can_move(point1)

4.1.3.4. ハンド制御スクリプト(hand_ctrl)

概要: HandCtrl RTC の活性化／非活性化を行い、実機 RH707 制御に関する I/F を提供する。

関数仕様:

関数名	引数	機能	使用例
assign_variables	Rtm環境	RtcHandleの変数への割り付け	<pre>>>>env = RtmEnv(sys.argv,[DEVICE_NS]) >>>for ns in env.name_space : >>>env.name_space[ns].list_obj() >>>assign_variables(env)</pre>
start_handle	Rtm環境のインスタンス	rtc_handlelによるRTCのハンドリング	<pre>>>>start_handle(env)</pre>
activate	なし	HandCtrl RTCの活性化	<pre>>>>activate()</pre>
deactivate	なし	HandCtrl RTCの非活性化	<pre>>>>deactivate()</pre>
reset	なし	HandCtrl RTCのリセット	<pre>>>>reset()</pre>
hand_open	0: (通常スピード) or 1: (ゆっくり)	ハンドの開口	<pre>>>>hand_open(0)</pre>
hand_close	0: (通常スピード) or 1: (ゆっくり)	ハンドの閉口	<pre>>>>hand_close(0)</pre>
hand_stop	なし	ハンド動作の中断	<pre>>>>hand_stop()</pre>
hand_cond	なし	ハンドの状態を取得	<pre>>>>hand_cond()</pre>

4.1.3.5. 飲み物認識スクリプト(VGRdrink_ctrl)

概要: OpenVGR 関連 RTC の自動接続、活性化／非活性化を行い、OpenVGR 制御に関する I/F を提供する。

関数仕様:

関数名	引数	機能	使用例
assign_variables	Rtm環境	RtcHandleの変数への割り付け	<pre>>>>env = RtmEnv(sys.argv,[DRINK_NS]) >>>for ns in env.name_space : >>>env.name_space[ns].list_obj() >>>assign_variables(env)</pre>
start_handle	Rtm環境のインスタンス	rtc_handleによるRTCのハンドリング	<pre>>>>start_handle(env)</pre>
activate	なし	OpenVGR RTCの活性化	<pre>>>>activate()</pre>
deactivate	なし	OpenVGR RTCの非活性化	<pre>>>>deactivate()</pre>
end	Rtm環境のインスタンス	OpenVGR制御の終了 (rtc_handleを終了するのみで、RTC自体は終了しない)	<pre>>>>end(env)</pre>
start_recog	モデルID	指定されたIDのモデルの認識開始	<pre>>>>start_recog(1) 1:190mm缶</pre>
lookat	モデル名	指定されたモデルを認識する	<pre>>>>lookat("can") 190mm缶を認識</pre>
chk_drink	認識データ	飲み物の認識結果として妥当かどうかの判定	<pre>>>>recognition = env.name_space[DRINK_NS].rtc_handles[' OpenVGRRecognition0.rtc '] >>>start_recog(model_data[" can"]) >>>data = recognition.outputs[' recogResult '].read().data >>>chk_drink(data)</pre>

4.1.3.6. ホルダ認識スクリプト(VGRholder_ctrl)

概要: OpenVGR 関連 RTC の自動接続、活性化／非活性化を行い、OpenVGR 制御に関する I/F を提供する。

関数仕様:

関数名	引数	機能	使用例
assign_variables	Rtm環境	RtcHandleの変数への割り付け	<pre>>>>env = RtmEnv(sys.argv,[RH_NS]) >>>for ns in env.name_space : >>>env.name_space[ns].list_obj() >>>assign_variables(env)</pre>
start_handle	Rtm環境のインスタンス	rtc_handleによるRTCのハンドリング	<pre>>>>start_handle(env)</pre>
activate	なし	OpenVGR RTCの活性化	<pre>>>>activate()</pre>
deactivate	なし	OpenVGR RTCの非活性化	<pre>>>>deactivate()</pre>
end	Rtm環境のインスタンス	OpenVGR制御の終了（rtc_handleを終了するのみで、RTC自体は終了しない）	<pre>>>>end(env)</pre>
start_recog	モデルID	指定されたIDのモデルの認識開始	<pre>>>>start_recog(15) 15:ドリンクホルダ</pre>
lookat	モデル名	指定されたモデルを認識する	<pre>>>>lookat(" holder")ドリンクホルダを認識</pre>
chk_holder	認識データ	ドリンクホルダの認識結果として妥当かどうかの判定	<pre>>>>recognition = env.name_space[RH_NS].rtc_handles[' OpenVGRRecognition0.rtc '] >>>start_recog(model_data[" holder"]) >>>data = recognition.outputs[' recogResult '].read().data >>>chk_holder(data)</pre>

5. その他

5.1. その他の要件

特になし。

5.2. 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