「基盤ロボット技術活用型
オープンイノベーション促進プロジェクト」
（事後評価）分科会
資料５

「基盤ロボット技術活用型
オープンイノベーション促進プロジェクト」

事業原簿【公開】

担当部	独立行政法人新エネルギー・産業技術総合開発機構 技術開発推進部

—目次—

概　要

最終更新日	平成 23 年 9 月 30 日

プログラム（又は施策）名	経済産業省「ロボット・新機械イノベーションプログラム」 内閣府「社会還元加速プロジェクト」		
プロジェクト名	基盤ロボット技術活用型オープンイノベーション促進プロジェクト	プロジェクト番号	P08014
担当推進部/担当者	機械システム部　影山啓二（平成２１年４月～平成２３年２月） 機械システム技術開発部　佐藤治道（平成２０年１１月～平成２１年３月） 機械システム技術開発部　堀野正也（平成２０年１０月）		
Ｏ．事業の概要	我が国では、1980 年代以降、自動車や電機・電子産業等のユーザ産業の成長や人手不足を背景に、産業用ロボットの本格的な導入が進んだが、1990 年代以降、産業用ロボットの市場規模は緩やかな成長にとどまり、用途も特定の産業分野に限られていた。 少子高齢化に伴って社会環境が急速に変化しつつある中、国際的にもトップレベルの我が国のロボット技術(以下、「ＲＴ」と記す)を、生活環境などの製造業以外も含む様々な分野で活用することが期待されている。 しかしながら、ＲＴを利用するには各種部品の使いこなしや、制御ソフト開発などの難しさ・煩雑さがあり、これがＲＴ応用分野への参入障壁となっていた。本プロジェクトではこの障壁を取り除き、ＲＴ応用分野の拡大、および種々の企業の新規参入を促すことを目的とする。 具体的には、次の３点について研究開発を実施する。 (1)センサなどの既存部品をＲＴで利用しやすくする「橋渡し役」の小さな基板「基盤通信モジュール」およびパラメータ設定やシステム構築を行うための「開発ツール」。 (2)上記モジュールを用いて既存の要素部品をＲＴで利用できる形にした「ＲＴ要素部品」。 (3)「ＲＴ要素部品」を用いた、生活環境などを安全・快適にするアプリケーションシステムの開発と機能実証。		
Ⅰ．事業の位置付け・必要性について	少子高齢化と人口減少による労働力不足、産業の国際競争激化等の諸課題を抱える中で、我が国が製造業のものづくり力を維持・強化し、サービス業の労働生産性を向上させるためには、人を支援・補佐する、もしくは人の代替としてのロボット活用が欠かせない。人は様々な形態で産業に関わっており、これらのロボットにも様々なニーズが求められている。このため、ロボットの開発は短期間で、コストは劇的に低くできる点が重要であり、部品・通信の標準化、市販品の活用が必須となっている。 本プロジェクトでは、すでに市販されているセンサやモータを用いることで、安価にロボットの要素部品を開発し、またこれらの要素部品が通信するためのネットワークシステムも容易に構築できるための技術開発を行い、製造分野をはじめとする一部の分野に限られているＲＴ適応分野を拡大すること、および、ロボット分野への中小・ベンチャーや異業種を含む多様な企業や研究機関等の新規参入を促進することにより、ロボット産業の裾野拡大を図ることを目的とする。		

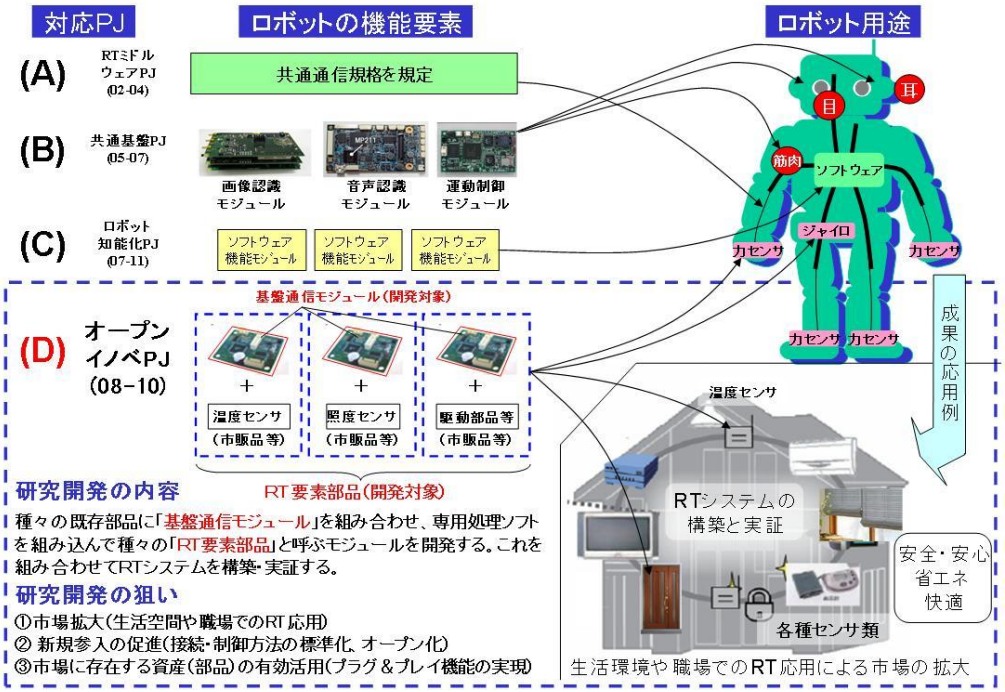
Ⅱ．研究開発マネジメントについて

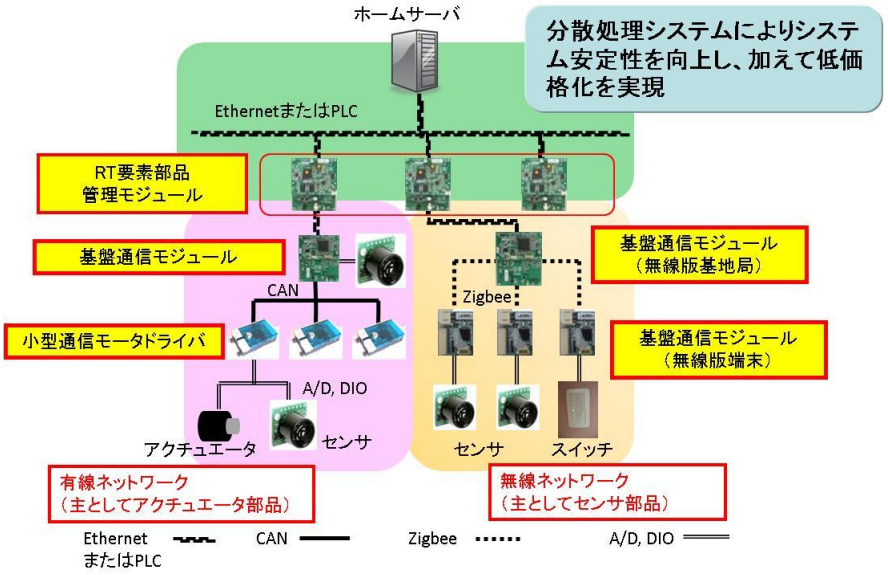
事業の目標	本プロジェクトでは、生活環境やロボットに使われる既存の要素部品を、共通の通信インタフェースとＲＴミドルウェアで動作させる「基盤通信モジュール」を開発する。次に、「基盤通信モジュール」を用いることにより既存の要素部品が容易にＲＴコンポーネント化でき、ＲＴシステム内で共通して利用できることを示すとともに、それを「ＲＴ要素部品」として広く提供する。さらに「ＲＴ要素部品」を用いた「ＲＴシステム」を開発し、実証試験を行い、同システムの有効性を検証することを目標とする。				
事業の計画内容	主な実施事項	H20fy	H21fy	H22fy	
	基盤通信モジュールおよび開発ツールの開発	→	→	→	
	基盤通信モジュールを用いた RT 要素部品の開発	→	→	→	
	RT 要素部品群による RT システムの開発・実証	→	→	→	

開発予算 （会計・勘定別に事業費の実績額を記載） （単位：百万円） 契約種類： ○をつける （委託（○）助成（　）　共同研究（負担率（　）	会計・勘定	H20fy	H21fy	H22fy	総額
	一般会計	49	135	99	283
	特別会計 （電源・需給の別）	0	0	0	0
	加速予算 （成果普及費を含む）	0	0	0	0
	総予算額	49	135	99	283

開発体制	経産省担当原課	製造産業局産業機械課
	プロジェクトリーダー	名城大学　理工学部機械システム工学科　教授　大道　武生
	委託先（＊委託先が管理法人の場合は参加企業数および参加企業名も記載）	(株)セック (株)ミサワホーム総合研究所 (株)テクノロジックアート ＴＨＫ(株) (株)アルゴシステム (大)大阪大学 (独)産業技術総合研究所
情勢変化への対応	（１)採択結果を受けての再公募の実施 　採択結果を検討した結果、研究開発内容の一部に開発力不足が懸念されたため、公募内容を該当開発項目に関し、追加公募を実施した。 （２)事業化をにらんだ研究開発内容の変更 　「RT 要素部品群による RT システムの開発・実証」では住宅を対象としたシステムを構築する計画であるが、米国のスマートグリッドにおいて PLC(電力線通信)技術が採用されることから、住宅内のネットワークとして PLC も含めた実証システムとするよう、研究開発内容を変更した。	
中間評価結果への対応	（中間評価は実施せず）	

評価に関する事項	事前評価	なし
	中間評価	なし
	事後評価	平成 23 年度　事後評価実施

Ⅲ．研究開発成果について （１／２）	（１）研究開発の概要 NEDO 技術開発機構では、ロボットの基本機能をモジュールとして部品化し再利用を促すことにより、毎度同様の開発をする必要なく高度なロボットを容易に構成可能とする技術を、一連の要素開発型プロジェクト群として推進してきた。図において、①～④はこれを可能にするプロジェクトを表しており、①において構成技術の基盤を、②～④においてロボットの機能部品を開発する。 本プロジェクトは図中の④にあたり、ＲＴ適用分野の拡大（生活空間や職場でのＲＴ応用）および新規参入の促進によるロボット産業の裾野拡大を目的として、共通の通信インタフェースとＲＴミドルウェアで動作させる基盤通信モジュール、既存の要素部品をＲＴコンポーネント化したＲＴ要素部品、それらを用いたＲＴシステムを開発するものである。  ＜事業全体の目標＞ 目標： 生活環境やロボットに使われる既存の要素部品を、共通の通信インタフェースとＲＴミドルウェアで動作させる「基盤通信モジュール」を開発する。 次に、「基盤通信モジュール」を用いることにより既存の要素部品が容易にＲＴコンポーネント化でき、ＲＴシステム内で共通して利用できることを示すとともに、それを「ＲＴ要素部品」として広く提供する。 さらに「ＲＴ要素部品」を用いた「ＲＴシステム」を開発し、実証試験を行い、同システムの有効性を検証することを目標とする。 ＜事業全体の成果＞ RTC-Lite フレームワークを基本とした RT ミドルウェア技術により、低価格な MPU で分散された RT 要素部品を安定に制御することを可能とする基盤通信モジュールを開発した。 基盤通信モジュールを利用し，RTC-Lite フレームワークの中で動作する RT 要素部品を開発した。 また，ユーザビリティを考慮し，プラグアンドプレイ機能を有する統合ミドルウェアを開発した．これらを利用した実証システムとして，住宅環境管理・支援 RT システムを構築し、評価を行った。

Ⅲ．研究開発成果について （２／２）	＜研究開発項目ごとの目標＞ 研究開発項目① 以下の条件を満たす「基盤通信モジュール」及び「開発ツール」を開発する。 １）ＲＴミドルウェアを実装し、研究開発項目②で開発するＲＴ要素部品がＲＴシステムからOpenRTM 仕様に基づきＲＴコンポーネントとして利用可能とする。 ２）独自のネットワークを用いるのではなく、既存の標準化されたネットワークと接続可能とする。また、「基盤通信モジュール」間の通信は特別な理由がない限り、既存の標準化された方式を用いる。 ３）家庭や職場の環境内に構築するＲＴシステムで必要となる要素部品と接続可能なインタフェースを有する。このインタフェース仕様は、要素部品の使われ方を考慮して設定する。 研究開発項目② 以下の条件を満たす「ＲＴ要素部品」を開発する。 １）「基盤通信モジュール」、又は「共通基盤モジュール」と組み合わされている。これらは要素部品と一体化されていることが望ましいが、処理部として分離されても良い。 ２）ＲＴシステムから OpenRTM 仕様に基づきＲＴコンポーネントとして利用できる。 研究開発項目③ 以下の条件を満たすＲＴシステムを開発して有効性を検証するための実証を行い、実用レベルを達成する。実証に際しては、プロジェクト期間内に、必要に応じて第三者に対してデモンストレーションする。 成果： ネットワーク上に RT コンポーネントとして RT 要素部品を参加させることを可能とする基盤通信モジュールを開発し，また，それらの基盤通信モジュール上で動作する軽量版 RT ミドルウェア（RTC-Lite フレームワーク）およびそれを利用する開発ツールを現在の RT ミドルウェアをベースにカスタマイズした。 各 RT 要素部品すべてに高機能なモジュールを付加する必要はなく，低価格，省電力，小型化を実現することが可能となった。 窓の自動開閉システム用アクチュエータを開発した。また、度・湿度センサを初めとして、雨量センサ、人感センサ等のモジュールを開発、また、各種リモコンで操作可能な設備をホームサーバーから操作可能とする赤外線通信モジュールを開発した。加えて、既存に製品として販売されている設備を本開発されたネットワークプラットフォームに参加させるインターフェース部分を開発し、既存製品を容易に組み込む際の技術的かつ金銭的な評価を行った。 研究開発項目②で開発された各種 RT 要素部品を住宅システムとして統合し、住宅の機能としてのシステム全体の設計および開発した RT ミドルウェアや、各通信モジュールの動作機能検証およびシステム評価を行った。 

	投稿論文	「査読付き」５件、「その他」１５件
	特　　許	「出願済」２件
	その他の外部発表（プレス発表等）	平成 23 年 2 月 28 日：NEDO ロボットプロジェクト成果報告会
Ⅳ．実用化、事業化の見通しについて	開発された技術を実用化していくためには、技術的な課題と共に、住宅設備制御におけるオープン化の考え方を業界に浸透させていくことが重要である。 オープンな規格を普及させていくためには、民間企業独自の活動だけでは困難であり、各企業を中立の立場からまとめる第三者的な機関が必要となる。このような活動に対して、従来から RT ミドルウェアの標準化に向けた活動を進めている産業技術総合研究所として、ロボットだけでなく、幅広い分野において RT ミドルウェアを初めとしたオープン規格の普及活動を進めていく必要があると考えている。 　現在、産業技術総合研究所コンソーシアムとして、住宅・ビルといった建築関係のユーザー側である企業をまとめたコンソーシアムをプロジェクト当初から構築しており、本事業で開発された成果を関連業界に普及する活動を並行して進めていくと考えている。また、住宅やビルといった建物だけでなく、スマートグリッドのような都市といった社会システムへの応用は同じ技術で可能であることから、より関連企業の枠を広げ、オープン規格の有効性をアピールしていくことが重要である。一方、技術的に RT ミドルウェアが導入されたシステムにおいて、そのシステムの安全基準や安全評価手法、ならびに、安全を担保するツールの整備を進めていくことも重要である。 ・要素部品管理モジュールおよび基盤通信モジュール（株式会社アルゴシステム） ・RT ミドルウェア（株式会社セック） ・RT ミドルウェア開発支援ツール（株式会社テクノロジックアート） ・小型通信ドライバモジュールと小型リニアアクチュエータ（THK 株式会社） ・RT ミドルウェアの標準化ならびに安全性の検討（独立行政法人産業技術総合研究所） について実用化、事業化の見通しを検討した。	
Ⅴ．基本計画に関する事項	作成時期	平成 20 年 4 月　作成
	変更履歴	平成 20 年 7 月　改訂（イノベーションプログラム基本計画制定により改訂）

プロジェクト用語集

RT	ロボット技術（Robot Technology）の略
OpenRTM-aist	ロボットシステムをコンポーネント指向開発するためのソフト ウエアプラットフォームとして、産業技術総合研究所がベンダー提供している RT ミドルウェアの種類の一つ。
RTC-Lite	組込 MPU である PIC で動作する RT デバイスを RT ミドルウェア上に参加させる枠組みの一つ。
RTC	RT ミドルウェア上において、動作する分割コンポーネント（RTC：Robot Technology Component）
PLC	電力線通信（PLC：Power Line Communication）。通信規格として HD-PLC、HomePlug AV、UPA があるが、本事業ではオープン性が高い規格を利用することとしたため、HomePlug AV を採用。
CAN	ネットワーク規格の一つ。CAN：Controller Area Network。CAN は、当初車載系の LAN として開発されたものであるが、現在は幅広い分野で利用されている。
Zigbee	家電向けの短距離無線通信規格の一つ。低速で転送距離が短い代わりに、安価で消費電力が少ないという特徴を持つ。基礎部分の（電気的な）仕様は IEEE 802.15.4 として規格化されている。論理層以上の機器間の通信プロトコルについては Zigbee Alliance が仕様の策定を行っている。

Ⅰ. 事業の位置付け・必要性について

1. NEDO の関与の必要性・制度への適合性

1.1 NEDO が関与することの意義

我が国のロボット産業は、産業用ロボットの普及により製造業を中心に拡大発展し、今日、国際的にもトップレベルのロボット技術（以下、「RT」という。）を蓄積している。我が国の少子高齢化や安心・安全の問題が急速に進行しつつある中、RTを製造業以外も含む様々な分野で活用することが期待されている。具体的には、労働力不足や要介護者の増加などの課題を解決するとともに、犯罪、災害や医療等における将来への不安の軽減による安心で安全な社会を実現する手段として、RTを駆使して機能を実現するシステム（以下、「RTシステム」という。）を効率的に開発、実用化して、様々な分野で活用することが期待されている。例えば、家庭や職場の環境内でセンサやモータなど既存の部品をネットワーク接続してRTシステムを構築し、状況に応じた判断によって人の活動を支援できれば生活環境をより快適かつ安全なものにし、職場の生産性を向上させることができる。

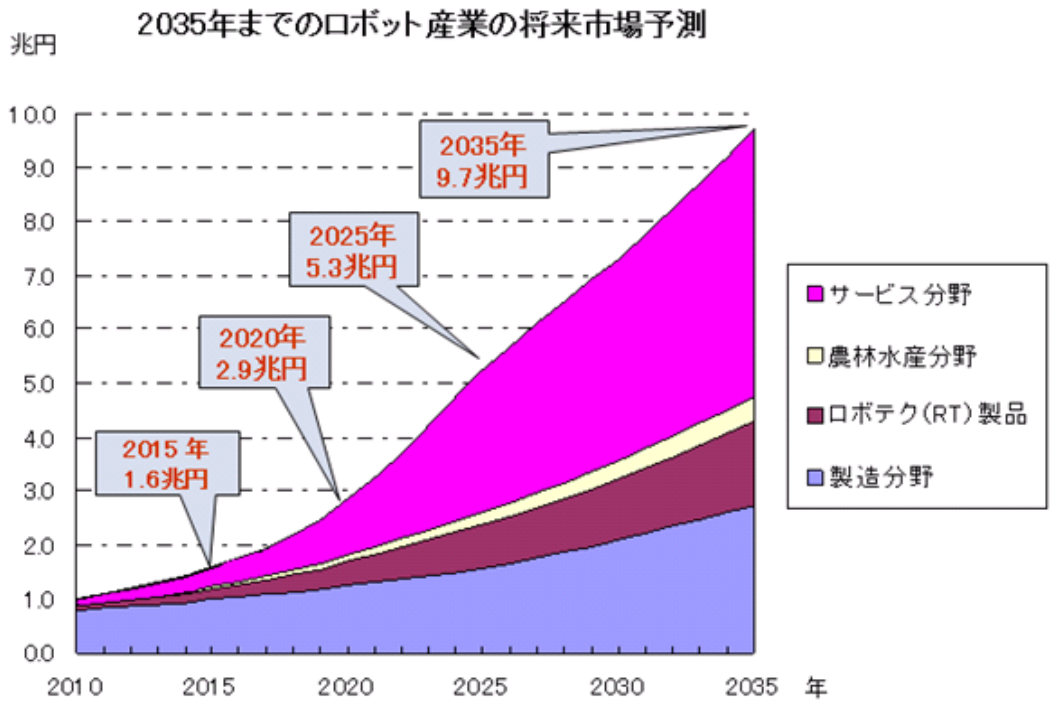
しかしながら、RTシステムを組み上げるには各種部品を集めて実装し、個々のシステムに合わせた制御ソフトを開発するという難しさ・煩雑さがあり、これがRT分野への新規参入の障壁となっていた。そこで独立行政法人新エネルギー・産業技術総合開発機構（以下、「NEDO技術開発機構」という。）ではこの障壁を解消することを目的として、RTミドルウェア並びに画像認識、音声認識及び運動制御の機能を有する共通基盤モジュールなどのRT開発基盤の整備を進めてきた。

これまでの開発成果を補完するものとして、生活環境やロボットで使用される各種要素部品を、RTシステムで利用しやすい共通の接続方式、制御方式のもとで利用可能な形で提供（RTコンポーネント化）するための基盤を開発し、またRTコンポーネント化された各種要素部品を用いることで既存の生活環境を簡単にRTシステム化し、さまざまな生活支援機能を提供することが可能であることを示す必要がある。

本事業は、少子高齢化への対応、新規産業創出による経済の活性化、安心・安全社会の実現等社会ニーズを満たすために必要なものであり、その成果は国民生活の質的向上等公共性が高く、新規産業創出や産業活性化が期待でき、産業政策的効果が高いことから、NEDO 技術開発機構の関与が必要である。

１．２ 実施の効果（費用対効果）

ロボット産業の市場は、2015 年には 1.6 兆円、2035 年には 9.7 兆円にまでに拡大すると予想される（図Ⅰ－1）が、基盤技術の普及と標準化・活用事例の創出がその予想を現実のものとするために必要である。プロジェクトの総実績額は約3億円であるが、今後の新たな市場創生・拡大を考えれば、費用に対する効果は妥当であると考える。



出典:平成 21 年度NEDO機械部調査報告書(委託先:三菱総研)

図Ⅰ-1　2035年までのロボット産業の将来市場予測

2.　事業の背景・目的・位置づけ

2.１　事業の背景

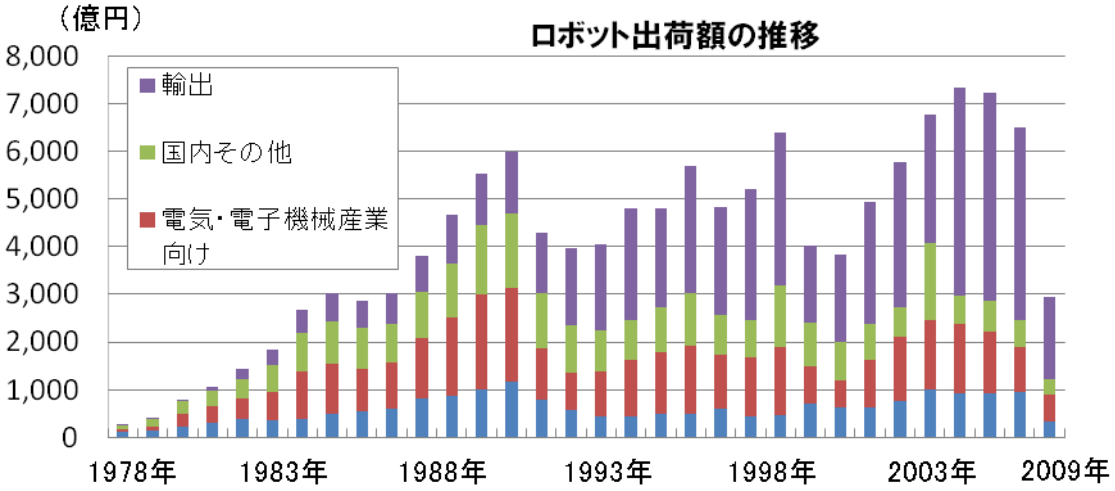
我が国では、1980 年代以降、自動車や電機・電子産業の成長や人手不足を背景に、産業用ロボットの本格的な導入ができた。現在、我が国は、国際的にもトップレベルのロボット技術を有するとともに、生産現場においても、全世界で稼働している産業用ロボットの約4割が日本で稼働している等、自他ともに認める「ロボット大国」といえる。ただし、1990 年代以降、産業用ロボットの市場規模は緩やかな成長にとどまり、用途も特定の産業分野に限られていた(図Ⅰ－2)。

しかし、ロボットを巡る状況は、着実に変わりつつある。製造業においては、ロボット・セルのように、さらに高度化した産業用ロボットが生産現場に投入されつつある。また、サービス業の分野においても、2005 年の愛知万博では、サービスロボットの実用化に向けた実証実験が行われるとともに、実際のビジネスにおいても、清掃ロボットや食事支援ロボット、災害復旧作業を行う遠隔操作型ロボット等の導入が進んでいる。このように、我が国のロボット産業・技術は、次の成長段階に踏みだしている。

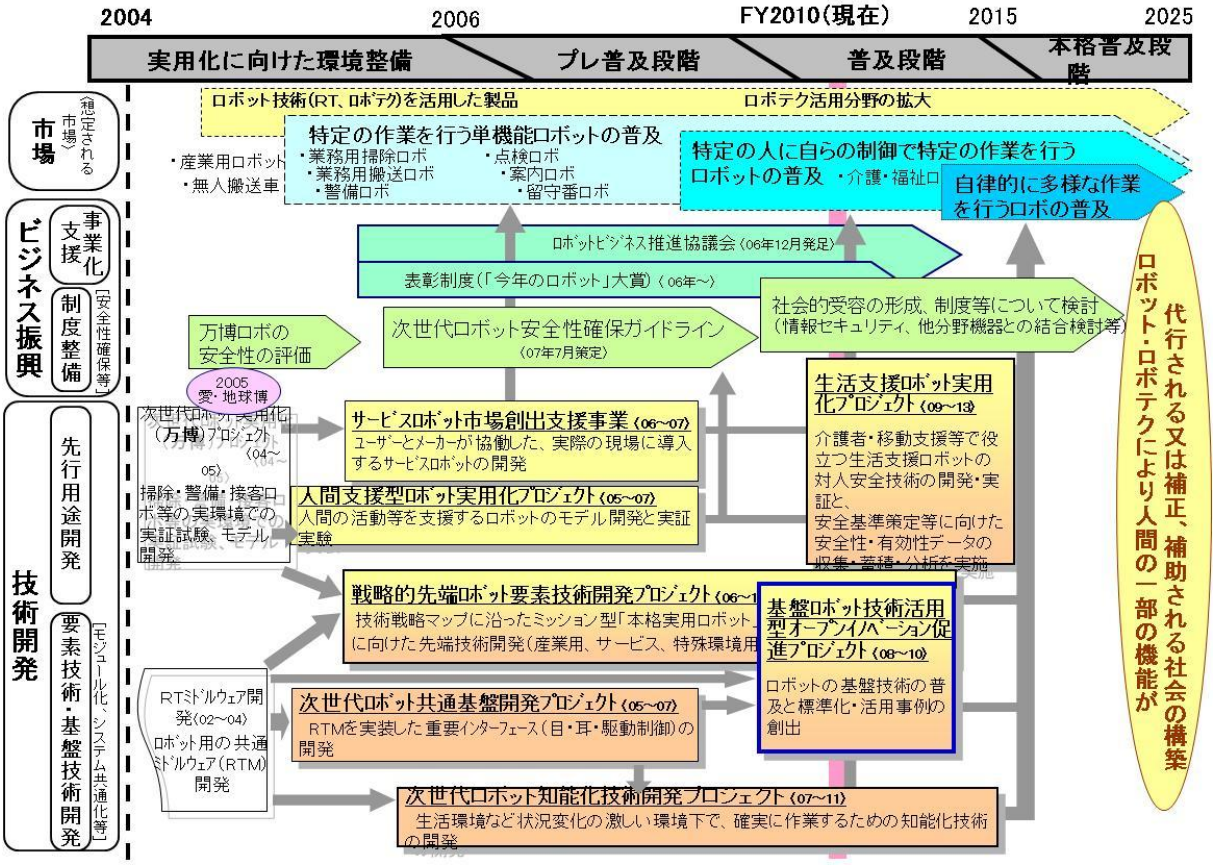
他方、我が国は、少子高齢化・人口減少、アジア諸国の台頭等を背景とした国際競争の激化や、地震や水害等大規模災害に対する不安といった社会的課題に直面している。我が国に蓄積された基盤的なロボット技術(RT)を活用・高度化することにより、これらの諸課題を解決することが期待されている。

NEDO技術開発機構ではこの障壁を解消することを目的として、RTミドルウェア並びに画像認識、音声認識及び運動制御の機能を有する共通基盤モジュールなどのRT開発基盤の整備を進めてきた。

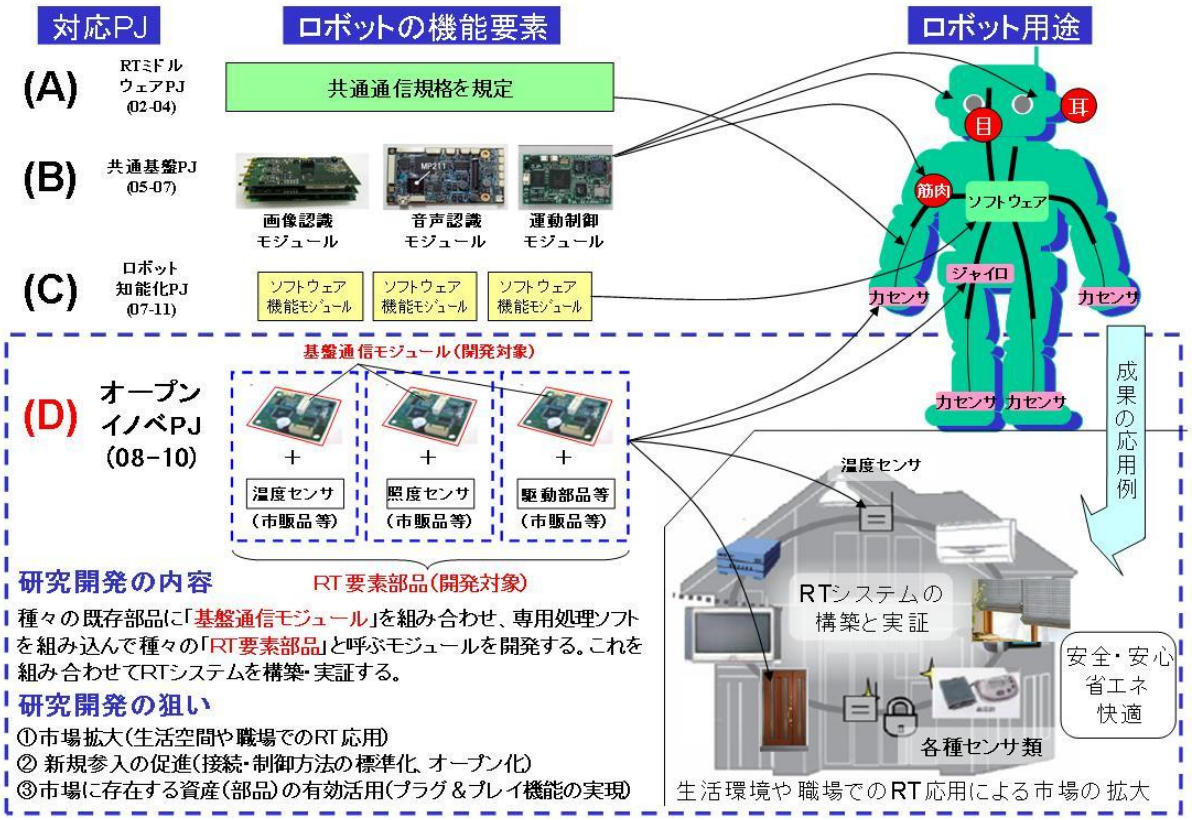
これまでの開発成果を補完するものとして、生活環境やロボットで使用される各種要素部品を、RTシステムで利用しやすい共通の接続方式、制御方式のもとで利用可能な形で提供(RTコンポーネント化)するための基盤を開発し、またRTコンポーネント化された各種要素部品を用いることで既存の生活環境を簡単にRTシステム化し、さまざまな生活支援機能を提供することが可能であることを示す必要がある。



図Ⅰ－2 ロボット出荷額の推移 （出展：ロボット工業会）



図Ⅰ－3 経済産業省「技術戦略マップ2010」より



図Ⅰ－4　本プロジェクトの概要

2．2　政策への適合性

「新産業創造戦略」(2005年6月経済産業省取りまとめ)では、先端的新産業分野として、「ロボット」を戦略7分野の一つとして掲げ、2010(平成22年)までの市場規模、その成長に向けたアクションプログラムを盛り込んでいる。当該アクションプログラムには、ユーザ(施設、地域)を巻き込んだ実証試験を中心としたモデル開発事業による先行用途開発、モデル事業と連携した重要な要素技術や共通インフラ技術の開発支援を行うこととしている。

「科学技術基本計画」(2006年3月閣議決定)では、ロボット・新機械技術は、特に重点的に研究開発を推進すべき分野(重点推進4分野)の一つである情報通信分野や、推進分野であるものづくり技術分野、社会基盤分野に位置づけられている。

「経済成長戦略大綱」(2006年7月財政・経済一体改革会議。2007年6月改訂版を経済財政諮問会議に報告)の中で、ロボット技術は産学官連携による世界をリードする新産業群の一つとして位置づけられ、次世代ロボットの市場の拡大に向けて、サービスロボット市場の整備、ロボットの認識技術の開発等必要な取組を継続することとしている。

「イノベーション25」(2007年6月閣議決定)では、ロボット・新機械技術は、生涯健康な社会生活や多様な人生を送れる社会の実現に向け、中長期的に取り組むべき課題として、新たな走行車等の普及促進のための環境整備、高度みまもり技術導入のためのルール作りなどの安全・安心な社会形成、また、ユビキタスネットワークや民生用ロボットの本格普及に向けた環境整備、低侵襲診断・治療技術の実現、安全・安心な社会のための将来デバイスの実現、さらに世界的課題解決に貢献する社会のための新しいものづくり技術など、今後の研究開発の進展等によって、その成果を社会に適用していく施策が必要であるとともに、随時見直しをし、その取組を加速・拡充していくことが必要とされている。

「経済財政改革の基本方針2008」（2008年6月閣議決定）では、経済成長戦略の3本の柱、革新的技術創造戦略のうち、革新的技術戦略のひとつとしてロボット技術が位置付けられた。

2．3　国のプログラムとの関連性

経済産業省が推進する「ロボット・新機械イノベーションプログラム」では、我が国の製造業を支えてきたロボット技術・機械技術を基盤とし、IT技術・知能化技術など先端的要素技術との融合を促進することにより、家庭、医療・福祉、災害対応など幅広い分野で活躍する次世代ロボットや新機械技術の開発・実用化を促進し、生産性の向上と人間生活の質の向上を実現するとともに、我が国経済社会の基盤である製造業の競争力の維持・強化を目指している。このロボット・新機械イノベーションプログラムの中で、「基盤ロボット技術活用型オープンイノベーション促進プロジェクト」は、これまでの研究開発プロジェクトの成果を活用し、生活環境やロボットで使用される各種要素部品をRT(Robot Technology)システムで利用しやすい共通の接続方式、制御方式の下で利用可能な形で提供（RTコンポーネント化）するための基盤を開発する。これにより既存の生活環境を簡単にRTシステム化し、それらを活用することにより様々な生活支援機能の提供、基盤ロボット技術の普及と標準化を推進する施策として位置づけられている。

また、内閣府が推進する「社会還元加速プロジェクト」では、高齢者・有病者・障害者への在宅での医療・介護などに資するロボット及びロボット技術（RT）の用途拡大、実用化促進及び社会的受容性の醸成を目指しており、その具体的な施策として本プロジェクトを位置付けている。

2．4　研究開発の目的

少子高齢化と人口減少による労働力不足、産業の国際競争激化等の諸課題を抱える中で、我が国が製造業のものづくり力を維持・強化し、サービス業の労働生産性を向上させるためには、人を支援・補佐する、もしくは人の代替としてのロボット活用が欠かせない。人は様々な形態で産業に関わっており、これらのロボットにも様々なニーズが求められている。このため、ロボットの開発は短期間で、コストは劇的に低くできる点が重要であり、部品・通信の標準化、市販品の活用が必須となっている。

本プロジェクトでは、すでに市販されているセンサやモータを用いることで、安価にロボットの要素部品を開発し、またこれらの要素部品が通信するためのネットワークシステムも容易に構築できるための技術開発を行い、製造分野をはじめとする一部の分野に限られているRT適応分野を拡大すること、および、ロボット分野への中小・ベンチャーや異業種を含む多様な企業や研究機関等の新規参入を促進することにより、ロボット産業の裾野拡大を図ることを目的とする。

Ⅱ. 研究開発マネジメントについて

1. 事業の目標

「基盤ロボット技術活用型オープンイノベーション促進プロジェクト」基本計画で以下のように定めている。
研究開発の目標（最終目標 平成22年度）:
本プロジェクトでは、生活環境やロボットに使われる既存の要素部品を、共通の通信インタフェースとRTミドルウェアで動作させる「基盤通信モジュール」を開発する。次に、「基盤通信モジュール」を用いることにより既存の要素部品が容易にRTコンポーネント化でき、RTシステム内で共通して利用できることを示すとともに、それを「RT要素部品」として広く提供する。さらに「RT要素部品」を用いた「RTシステム」を開発し、実証試験を行い、同システムの有効性を検証することを目標とする。
研究開発の目標の詳細については、研究開発計画（別紙）に記載のとおり。

2. 事業の計画内容

2．1 研究開発項目

上記目標を達成するために、次の3つの研究開発項目について研究開発を実施する。

①基盤通信モジュール及び開発ツールの開発

②基盤通信モジュールを用いたRT要素部品の開発

③RT要素部品群によるRTシステムの開発・実証

以下に研究開発項目ごとの研究開発目標を示す。

2．1．1 研究開発項目①「基盤通信モジュール及び開発ツールの開発」

2．1．1．1 研究開発の必要性

家庭や職場の環境内でRTシステムを構築するには、多数のセンサやモータなどの要素部品と制御装置を接続する必要があるが、現状では要素部品が共通化・標準化されていないため煩雑でコストが高くなり、実用化が困難な状況である。したがって、RTシステムを普及させるには、要素部品の共通化・標準化を図り、低コストかつ短期間でRTシステムを開発できるようにすることが重要である。とりわけ既存の環境に「RT要素部品」を導入することにより容易に生活環境の中でRTを利用できるようにすることはRTシステムの普及に大きく貢献する。これを実現するには、既存の要素部品に標準的なネットワークとの接続機能及びRTミドルウェアで動作する機能を与えてRTコンポーネント化する従来にないモジュールの開発が必要である。そこで本研究開発項目では、この機能を有する「基盤通信モジュール」を開発する。

また、開発する「基盤通信モジュール」を各種要素部品と組み合わせて「RT要素部品」とする際には、RT要素部品開発機関がネットワーク接続側及び要素部品接続側の各種パラメータ設定、信号処理プロセスのプログラミングなどを行う必要がある。これを容易にする「開発ツール」も併せて開発することにより、様々なタイプの「RT要素部品」の低コスト化及び開発期間の短縮が期待できる。

2．1．1．2 研究開発の具体的内容

既存のセンサ、モータなどの要素部品をネットワーク接続可能としRTコンポーネント化するための「基盤通信モジュール」及び「開発ツール」の開発を行う。

2．1．1．3 達成目標

（1）基本性能

以下の条件を満たす「基盤通信モジュール」及び「開発ツール」を開発する。

① RTミドルウェアを実装し、研究開発項目②で開発するRT要素部品がRTシステムから OpenRTM 仕様に

基づきRTコンポーネントとして利用可能とする。

② 独自のネットワークを用いるのではなく、既存の標準化されたネットワークと接続可能とする。また、「基盤通信モジュール」間の通信は特別な理由がない限り、既存の標準化された方式を用いる。

③ 家庭や職場の環境内に構築するRTシステムで必要となる要素部品と接続可能なインタフェースを有する。このインタフェース仕様は、要素部品の使われ方を考慮して設定する。

（2）その他の性能

① 提案に基づきプロジェクト着手の際には要素部品の消費電力目標を設定し、低消費電力化を目指す。

② 種々の要素部品との組み合わせを考慮して、できる限り小型化軽量とし、サイズ（基板面積）を名刺の 1/2 以下とする。ただし、開発状況によりサイズ目標は適宜見直すものとする。なお、コネクタと電源をこのサイズに納めることは必須条件ではなく、プロジェクト終了後、無理なく実用化可能とする。

（3）開発品の提供

① 開発した「基盤通信モジュール」をRT要素部品開発機関が利用できる形で提供する。

② (i)各種要素部品との接続、(ii)要素部品の制御・信号処理プログラム、(iii)ネットワークとの通信設定とその保持などを容易とする「開発ツール」を、RT要素部品開発機関が利用できる形で提供する。

③ 開発した「基盤通信モジュール」及びRTミドルウェアにて動作させる際に必要となる項目を記載した仕様書及び取扱説明書をRT要素部品開発機関が利用できる形で提供する。

（4）有効性の検証

① 開発した「基盤通信モジュール」及び「開発ツール」が開発仕様を満たし、有効に機能することをコンソーシアムメンバ間で協力して検証する。

② RTシステムの実証に際しては、RTシステム開発機関と協力して、開発品が有効に機能することを検証する。

（5）特記事項

RTコンポーネントは下記の OpenRTM 仕様書に準拠するものとする。

Robotic Technology Component (RTC)、1.0 adopted、OMG。

http://www.omg.org/technology/documents/domain_spec_catalog.htm

２．１．２　研究開発項目②「基盤通信モジュールを用いたＲＴ要素部品の開発」

２．１．２．１　研究開発の必要性

ＲＴを利用して生活環境をより快適かつ安全にしたり、職場の生産性を向上させるには、家庭や職場の環境の状態をモニタして、必要な処理を自動で実行できることが重要である。

具体的には、温度、湿度や照度などの物理量の値や推移をセンシングし、それに基づいて情報の発信、空調や照明等の機器のコントロール、アクチュエータによる扉やブラインドの開閉等の複合的な機能を実現することが求められる。これらを実現するには、センサやアクチュエータなどの要素部品をネットワークで接続し、ＲＴシステムとして動作させることが必要である。また、これらの機能を有するＲＴシステムを容易に開発する仕組みも必要である。しかしながら、高度な機能を有するＲＴシステムを容易に構築できる要素部品は一般に入手できる形では実用化されていない。そこで本研究開発項目では、高度な機能を容易に実現できるＲＴミドルウェアを用いて、ＲＴシステム内で共通して利用できる「ＲＴ要素部品」を開発する。これにより、用途に合わせたＲＴシステムを容易に構築できるようになる。

２．１．２．２　研究開発の具体的内容

研究開発項目①で開発された「基盤通信モジュール」と、既存のセンサ、モータなどの要素部品とを「開発ツール」を用いて接続し、ネットワーク接続及びシステム化を可能とする「ＲＴ要素部品」の開発を行う。「基盤通信モジュール」に代えて、「次世代ロボット共通基盤開発プロジェクト」にて開発された、「画像認識、音声認識、運動制御の機能を有する共通基盤モジュール」（別添２）から使用できるものを選定して利用しても良い。

２．１．２．３　達成目標

以下の条件を満たす「ＲＴ要素部品」を開発する。

（１）基本性能

① 「基盤通信モジュール」、又は「共通基盤モジュール」と組み合わされている。これらは要素部品と一体化されていることが望ましいが、処理部として分離されても良い。

② ＲＴシステムから OpenRTM 仕様に基づきＲＴコンポーネントとして利用できる。

（２）開発品の提供

開発したＲＴ要素部品及びＲＴミドルウェアにて動作させる際に必要となる項目を記載した仕様書及び取扱説明書を、ＲＴシステム開発機関が利用できる形で提供する。

（３）有効性の検証

① 開発した「ＲＴ要素部品」が開発仕様を満たし、有効に機能することをコンソーシアムメンバ間で協力して検証する。

② ＲＴシステムの実証に際しては、ＲＴシステム開発機関と協力して、開発品が有効に機能することを検証する。

（５）特記事項

ＲＴコンポーネントは下記の OpenRTM 仕様書に準拠するものとする。

Robotic Technology Component (RTC)、1.0 adopted、OMG。

http://www.omg.org/technology/documents/domain_spec_catalog.htm

２．１．３ 研究開発項目③「RT要素部品群によるRTシステムの開発・実証」

２．１．３．１ 研究開発の必要性

生活環境をより快適かつ安全にしたり、職場の生産性を向上させるには、実際に家庭や職場の環境の状態をモニタして、必要な処理を自動で実行できるRTシステムを構築して、その機能を検証することが重要である（イメージ図参照）。また、これらの機能を有するRTシステムを容易に開発する仕組みも必要である。しかしながら、高度な機能を有するRTシステムは一般に入手できる形では実用化されていない。

そのため、RTシステムの開発と、それを用いた実証システムの構築と、その有効性を検証することが必要である。

照度センサ温度センサ湿度センサ制御装置・・・基盤通信モジュール家庭および職場環境ネットワークモータモータ冷蔵庫ドア既存機器（電灯・エアコン等）

実証システム例のイメージ図

２．１．３．２ 研究開発の具体的内容

研究開発項目①及び②で開発されたRT要素部品群を用いてRTシステムを開発し、その実証システムを家庭や職場を模擬した環境内に構築して有効性の検証を行う。

2．1．3．3 達成目標

以下の条件を満たすRTシステムを開発して有効性を検証するための実証を行い、実用レベルを達成する。実証に際しては、プロジェクト期間内に、必要に応じて第三者に対してデモンストレーションする。

（1）基本性能

①RT要素部品を組み合わせたRTシステムとする。RTシステム化の際に必要となる制御装置も開発する。

②OpenRTM 仕様に基づくRTシステムとして開発する。

③システムインテグレータ（本プロジェクトではRTシステム開発機関がこれに相当）や、ある程度のスキルを持ったエンドユーザが容易にシステムの構築、変更ができるシステム構築・運用ツールを開発する。

④実証システムは実証目的に合わせて、要素部品又はRTシステムの入れ替えが可能である。

⑤実証システムは、要素部品としてセンサとアクチュエータとを含める構成とする。

（2）その他の性能

家庭や職場の環境内で構築されるRTシステムは、バッテリや省電力での駆動が想定されるので以下の条件を満たす。

①システム全体の省電力を意識した構成とする。

②家庭や職場の環境で想定される温度、湿度、照度などの諸条件のもとで正常に動作する。

（3）有効性の検証

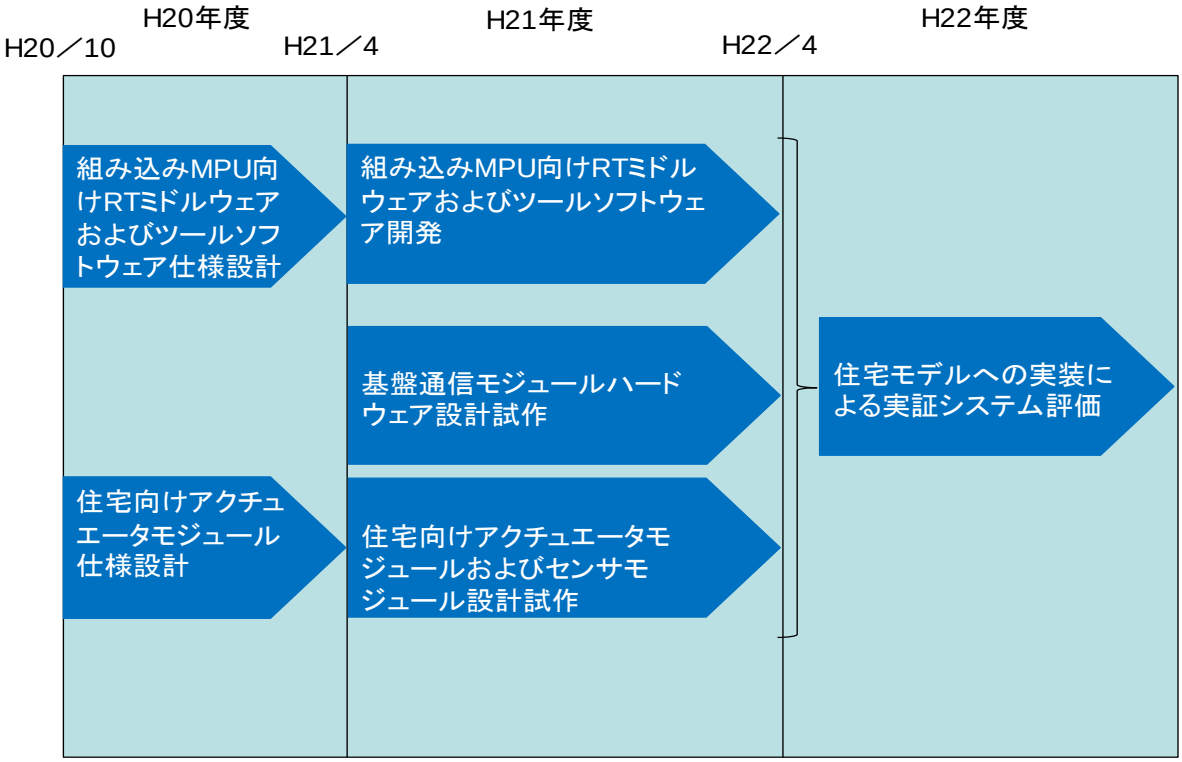
① 実証システムや小規模な個別デモシステムなどを用いて、推進委員会等によるRTシステム及び実証システムの有用性評価を適時実施する。

② 実証で得られた結果や知見を基盤通信モジュール開発機関及びRT要素部品開発機関にフィードバックし、必要に応じて改良を促す。

２．２　研究開発計画

研究開発スケジュールを図Ⅱ－１、予算を表Ⅱ－１に示す。

研究開発は、平成２０年度の下期から平成２２年度末まで実質２年半の期間で実施した。初年度は、技術課題の抽出およびソフトウェア、ハードウェアの仕様策定を実施、２年度目は、ソフトウェア開発およびハードウェア設計試作、３年度目は住宅モデルへの実装による実証システム評価を行った。



図Ⅱ－１　研究開発実施スケジュール

表Ⅱ－１　研究開発予算（実績）

単位：百万円

事業者＼年度	H20年度	H21年度	H22年度	合計
セック	3.9	17.4	21.9	43.2
ミサワホーム総合研究所	6.0	12.1	19.1	37.2
テクノロジックアート	18.0	20.0	9.2	47.2
ＴＨＫ	6.0	18.4	19.1	43.5
アルゴシステム	－	52.7	16.6	69.3
大阪大学	7.2	4.0	7.2	18.4
産業技術総合研究所	8.0	10.7	6.3	25.0
合計	49.1	135.3	99.4	283.8

２．３　研究開発の実施体制

本研究開発は、ＮＥＤＯ技術開発機構が実施者を公募により選定し、委託により実施した。本研究開発の推進にあたって、ＮＥＤＯが研究開発責任者（プロジェクトリーダー：ＰＬ）として名城大学理工学部 機械システム工学科　大道武生　教授に委嘱した。

図Ⅱ－２　実施体制

２．４　研究開発の運営管理

NEDO技術開発機構は、プロジェクトリーダー及び経済産業省と密接な関係を維持しつつ、研究開発状況の確認・推進及び必要な調査研究の推進、並びに意志決定・計画見直しなど本研究開発の目的及び目標に照らして適正な運営管理を実施した。

本事業の研究開発の進捗管理等を実施する目的として、推進会議・PL現地指導・進捗確認シートを使用し適切な運営を行った。

２．４．１　推進会議

推進会議は、NEDOが主催し、プロジェクトリーダー、NEDO部長以下の幹部職員、担当主査、実施者の代表と担当者が参加し、口頭発表による進捗確認会議である。実施者代表による概要進捗報告、成果や課題の発表を行い、情報共有とともにPLやNEDOによるコメントとともに進捗指導を行なった。各発表には質疑討論の時間を設定して、理解を深める会議とした。

推進会議はおよそ3カ月に一度開催し、開発の節目ごとの進捗管理を適正に行なった。プロジェクト期間中に定期推進会議は合計10回開催した。

定期の推進会議以外にも、技術課題を中心にした課題抽出などの目的で臨時会議を開催し、プロジェクトの成果が最大化するように運営を行なった。

2．4．2　進捗確認シート

2年度目（平成21年度）の第1四半期から3カ月ごとに、目標に対する研究開発の進捗資料を実施者（委託先）ごとに作成した。NEDOとPLが内容確認の上、コメントと評価を記入し実施者にフィードバックすることで進捗と予算の充当状況を管理した。この進捗管理シートはプロジェクト期間中に計8回作成した。

2．4．3　PLによる現地指導

研究開発の初期段階にPLが実施者の研究実施場所に訪問し、研究計画の指導および技術指導を行った。実施内容の整理とモチベーションの向上に効果があった。

３．情勢変化への対応

３．１　採択結果を受けての再公募の実施

採択結果を検討した結果、研究開発内容の一部に開発力不足が懸念されたため、公募内容を該当開発項目に関し、追加公募を実施した。

３．２　事業化をにらんだ研究開発内容の変更

「RT 要素部品群による RT システムの開発・実証」では住宅を対象としたシステムを構築する計画であるが、米国のスマートグリッドにおいて PLC(電力線通信)技術が採用されることから、住宅内のネットワークとして PLC も含めた実証システムとするよう、研究開発内容を変更した。

Ⅲ. 研究開発成果について

1.事業全体の成果

１．１ 事業概要

本開発事業ではロボット技術の応用アプリケーションとして住宅といった建築内におけるインテリジェントな環境埋め込み型ロボットシステムのビジネス展開を目指し，建築物内に分散させた RT（Robot Technology）要素部品を連係動作させることで，建物の品質管理および利便性が向上することを実証する．

RTC-Lite フレームワークを基本とした RT ミドルウェア技術により低価格な MPU を利用し，分散された RT 要素部品を安定に制御することを可能とするモジュール間の安定な制御機能を持った基盤通信モジュールを開発する．また，ユーザビリティを考慮し，プラグアンドプレイ機能を有する基盤通信モジュール用 RT ミドルウェアおよびその統合ミドルウェアを開発する．これらを利用した実証システムとして，ビジネスにつながる住宅環境管理・支援 RT システムを構築し，開発したシステム評価を実施する．

研究開発項目は次の３点である．

(a)「基盤通信モジュールおよび開発ツールの開発」

(b)「基盤通信モジュールを用いた RT 要素部品の開発」

(c)「RT 要素部品群による RT システムの開発・実証」

以上の 3 項目について、成果をまとめる。

1-1-a　基盤通信モジュールおよび開発ツールの開発

本研究開発項目ではネットワーク上に RT コンポーネントとして RT 要素部品を参加させることを可能とする基盤通信モジュールを開発し，また，それらの基盤通信モジュール上で動作する軽量版 RT ミドルウェア（RTC-Lite フレームワーク）およびそれを利用する開発ツールを現在の RT ミドルウェアをベースにカスタマイズすることを目標とする．

本研究開発でターゲットとする RT システムは，基盤通信モジュール，RT 要素部品管理モジュール，ホームコントローラ用 PC の 3 階層のモデルにより構成される．

基盤通信モジュールは，通信プロトコルとして Zigbee 無線，CAN、および PLC などで通信し，モジュール内の MPU としてコストの低い SH2 を利用している。また、RT ミドルウェアとして RTC-Lite フレームワークを備え，センサやアクチュエータなどの RT 要素部品を制御するための通信モジュールとして機能する．

RT 要素部品管理モジュールは，通信プロトコルとして PLC を利用し，MPU として SH4 や ARM9，RT ミドルウェアとして OpenRTM-aist を備えた高性能，低消費電力の比較的小型な CPU モジュールである．RT 要素部品管理モジュールは，複数の基盤通信モジュール群を管理する．

ホームコントローラ用 PC は，住宅内に設置した RT 要素部品の状態管理および住居者とのインターフェース機能を有する．本 PC は，RT 要素部品管理モジュールと PLC 経由で接続されており，接続されている RT 要素部品管理モジュール群を管理する．

以上のように，階層を分けることで，各 RT 要素部品すべてに高機能なモジュールを付加する必要はなく，低価格，省電力，小型化を実現することが可能となる．また，センサ出力の情報量が多い場合など，必要に応じて，RT 要素部品管理モジュールに直接 RT 要素部品を付加することも可能であり，ユーザーが付加する RT 要素部品の仕様によってモジュールを選ぶことを可能とする．また，RT 要素部品管理モジュールを複数設置することで，RT 要素部品の分散管理が可能となり，ホームコントローラ用 PC の負荷が低減されることで，システムの安定性の向上が期待できる．

本項目は更に以下の 4 つの開発項目で進められた。

（a-1） RT 要素部品管理モジュールおよび基盤通信モジュールの研究開発

（a-2） RTC-Lite フレームワークに基づく基盤通信モジュール RT ミドルウェアの研究開発

（a-3） RTC-Lite およびプラグアンドプレイに対応した RT ミドルウェア開発支援ツールの研究開発

（a-4） RT 要素部品管理モジュール用組み込み RT ミドルウェアの研究開発

以下にそれぞれの開発目標に対する研究開発成果および達成度をまとめる。

(a-1) RT 要素部品管理モジュールおよび基盤通信モジュールの研究開発

目標	研究開発成果	達成度
(1)RT 要素部品が容易に RT ミドルウエア上に参加できる低価格での分散型 RT 要素部品管理モジュール及び基盤通信モジュールの実動作ハードウエアの開発。 *具体的目標・コストの高い RT 要素部品用の基盤通信モジュール原価は 20,000 円を目指す。(出典:基本計画 p3) ・基盤通信モジュールのサイズを名刺半分にする。(出典:基本計画 p3)	(1)Ethernet 版及び PLC 版の RT 要素部品管理モジュールと基盤通信モジュールの開発と提供を行った。また、多様な通信プロトコルを許容するためのブリッジ機能を RT 要素部品管理モジュールに組込み,ホームネットワークシステムとしての拡張性を広げた。 ・要素部品、基盤通信、Zigbee 各 モジュールは 20,000 円以内実現。 ・基盤通信モジュールは名刺サイズ 2/3 を実現。尚、量産時に BGA 部品使用で名刺サイズの 1/2可能性あり。	(1)達成 サイズに関しては量産時に名刺半分サイズを実現

(a-2) RTC-Lite フレームワークに基づく基盤通信モジュール RT ミドルウェアの研究開発

目標	研究開発成果	達成度
(1)基盤通信モジュールで動作する RT ミドルウェアとして、OMG RTC 仕様に準拠した RTC-Lite フレームワークを実現する。この RTC-Lite フレームワークは、H8、PIC などの省資源なマイコンで動作可能なものとする。物理的な通信プロトコルとして、微弱無線、ZigBee、CAN を主要候補として、2 種以上の通信プロトコルをサポートする。(出典:基本計画 p5)	(1)RTC-Lite フレームワークに基づく、CAN 通信対応の miniRTCs および、Zigbee 通信に対応した microRTCs の 2 つの組込み用軽量 RT ミドルウェアを開発した。	(1)達成
(2)基盤通信モジュールの実証評価版は、平成 21 年度中に RT 要素部品開発機関ならびに、RT システム開発機関に無償で提供し、RT 要素部品の開発や RT システムの開発に供する。この際、RT ミドルウェアだけではなく、仕様書および取扱説明書もあわせて提供する。(出典:基本計画 p5)	(2)本プロジェクトのコンソーシアムメンバの他、コンソーシアム外メンバ(滝田技研株式会社、株式会社オカテック、株式会社リバスト、旭光電機株式会社)に miniRTCs および、microRTCs のソースコードならびに、インタフェース仕様書、利用マニュアルを提供した。	(2)達成
(3)実証評価の結果をフィードバックし、基盤通信モジュールの最終版をプロジェクト終了時までにリリースする。(出典:基本計画 p5)	(3)実証評価で発生した課題・問題を解決した最終版をコンソーシアムメンバにリリースした。最終的な RT ミドルウェアのインタフェース仕様は成果報告書として一般に公開する。	(3)達成見込み(2011 年 8 月)

(a-3) RTC-Lite およびプラグアンドプレイに対応した RT ミドルウェア開発支援ツールの研究開発

目標	研究開発成果	達成度
①RT 要素部品(RT コンポーネント)化支援ツールの開発(出典:基本計画 p8)		
(1)組み込み MPU 上で動作する組み込み RT コンポーネントの仕様から雛形コードを自動生成するツールの設計, 開発	(1)本ツールでは, 本プロジェクトにて開発を行った各種基盤通信モジュールおよび汎用的な組み込み MPU 上で動作する RT コンポーネントの雛形コードを生成可能である. 本ツールにて生成可能な RT コンポーネントを以下に示す. ・高速制御用(CAN 用)RTC-Lite(miniRTC) ・低速制御用(Zigbee 用)RTC-Lite(microRTC) ・PIC/dsPIC 用 RTC-Lite	(1)達成
(2)通常の RT コンポーネントを作成する場合と類似の操作で, 組み込み RT コンポーネントの雛形コードを生成する機能.	(2)本ツールは, 既存ツールとの操作性の統一, 連携の容易化を図るため, 独立行政法人 産業技術総合研究所が開発を行っている RTCBuilder を拡張する形式で Eclipse プラグインとして実現している.	(2)達成
(3) 生成対象コンポーネントの仕様を,「RT コンポーネント仕様記述方式」を拡張した形式として保存/読み込みする機能.	(3) 本ツールは各組み込み MPU に固有な拡張情報を,「RT コンポーネント仕様記述方式(RTCProfile)」の拡張領域に保持する形式とし, 生成したコンポーネント情報の保存／読み込みを行えるようにした.	(3)達成
(4)プラグイン方式を採用し, 用途に応じて各種機能追加, カスタマイズを容易に実行できる構成.	(4)各 RT コンポーネント向けの雛形コード生成ツールは, 別々の独立したプラグインとして実現しているため, 利用者が使用したい RT コンポーネント用のツールのみを選択, インストールする事が可能となっている. また, 必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで, 利用者自身が使用する開発環境全体をカスタマイズする事ができるようになっている.	(4)達成
(5)デバイス分類毎に共通な制御・信号処理の部分を生成する機能.	(5)PIC/dsPIC 用 RTC-Lite 向けツールについては, コンポーネント・パターンの仕組みを採用し, 使用頻度が高いと考えられる「デジタル入力」「デジタル出力」「アナログ入力」「PWM 制御」の種類を選択するだけで, より詳細なコードを自動生成することができる. また, 他の RT コンポーネントについては, ECHONET(Energy Conservation and Homecare Network)にて規定されている各種標準コマンドを送受信するためのインターフェース定義に対応した雛形コード生成機能を実現するとともに, ユーザーが設定画面にてカスタマイズできるようになっている.	(5)達成

②RT 要素部品(RT コンポーネント)コンフィギュレーション支援ツールの開発（出典：基本計画 p8）		
(1) 組み込み RT コンポーネントのコンフィギュレーション情報の設定支援を行う機能	(1) 本ツールでは，RT コンポーネント(組み込み RT コンポーネント含む)にて必要となる各種コンフィギュレーション情報の入力項目の名称や単位変換などを行い，設備機器メーカー視点からの各種項目設定を実現した．また，各種パラメータに対して設定された制限値を用いることで，利用者が誤って不正な値を設定することを防止する機能を実現した．	(1)達成
(2) プラグイン方式を採用し，必要に応じて各種機能追加，カスタマイズを容易に実行できる構成．	(2) 本ツールは，既存ツールとの操作性の統一，連携の容易化を図るため，独立行政法人産業技術総合研究所が開発を行っている RTSystemEditor を拡張する形式で Eclipse プラグインとして実現している．このため，必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで，利用者自身が使用する開発環境全体をカスタマイズする事ができるようになっている．	(2)達成
(3) 類似デバイスの設定情報の再利用などを容易に行える機能．	(3) 本ツールでは，各種設定に用いる情報をチューニング定義ファイルとして，外部ファイルに保持する形式を採用している．そして，必要に応じてこの情報のみのインポート/エクスポートが行えるようになっている．このため，類似の RT コンポーネントを使用してシステムを構築する場合は，チューニング定義ファイルを再利用することで，各種詳細設定を行う負担を削減でき，開発効率を向上させることができる．	(3)達成

③RT システム構築支援ツールの開発（出典：基本計画 p8）		
(1)通常の RT コンポーネントのみを用いて RT システムを構築する場合と類似の操作で，組み込み RT コンポーネントを含んだ RT システムを構築する機能.	(1)今回開発したシステムでは，システムの可用性・信頼性を高めるため，コントローラが複数箇所に分散配置されている．また，高速制御用 RTC-Lite(miniRTC)，低速制御用 RTC-Lite(microRTC)は，実際の制御/センサ用基盤の上で各コンポーネントが動作しているが，これらのデバイスを設備機器アプリケーション単位で管理するためのモジュール(RTC-Lite Manager)が，基盤通信モジュール上で動作している．そこで，これらの仕組みを隠蔽し，通常の RT コンポーネントと同様に各種デバイス向けコンポーネントを操作するためのツールを開発した．また，各デバイス単位での操作，情報取得を実現するとともに，デバイス間のポートの接続/切断を実行する機能を実現した.	(1)達成
(2)システムインテグレータやある程度のスキルを持ったエンドユーザーが，容易に RT システムを構築することができる RT システム構築支援ツールの研究開発	(2)今回開発を行ったシステムでは，要素部品管理モジュール上の RTCHub が，各種機器単位で管理，制御するため，RTC-Lite Manager 毎に疑似複合コンポーネントを作る形で動作している．そこで， RTCHub を構築する際に必要となるプロファイルテーブル情報を生成するツールを開発した．この機能を利用することで，システムインテグレーターやある程度のスキルを持ったエンドユーザーが，通常の RT コンポーネントのみを用いて RT システムを構築する場合と類似の操作で，組み込み RT コンポーネントを含んだ RT システムを容易に構築することが可能となる. また，オフライン状態で構築した RT システムの情報から，実際のシステムを起動するためのツールを開発した.	(2)達成
(3)構築した RT システムの情報を「RT システム仕様記述方式」を拡張した形式で保存/読み込みする機能.	(3)RTCHub の仕組みを隠蔽するためのツールおよびプラグアンドプレイ機能に必要となる情報を設定するツールを用いて設定を行った内容を，「RT システム仕様記述方式(RTSProfile)」を利用して保存/読み込みができる機能を実現した.	(3)達成
(4)プラグイン方式を採用し，用途に応じて各種機能追加，カスタマイズを容易に実行できる構成.	(4)本ツール群は，既存ツールとの操作性の統一，連携の容易化を図るため，独立行政法人産業技術総合研究所が開発を行っている RTSystemEditor を拡張する形式で Eclipse プラグインとして実現している．このため，必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで，利用者自身が使用する開発環境全体をカスタマイズする事ができるようになっている.	(4)達成

(5)プラグアンドプレイ機能を実現するために必要な各種情報を設定可能であるとともに，構築した RT システムの RT システム仕様記述情報ファイルの内容から，各 RT コンポーネントを設定された手順で起動/終了可能なツールの研究開発.	(5)プラグアンドプレイ設定ツールでは，プラグアンドプレイ機能を実現するために必要となる各種情報を RT システム構築時に設定するとともに，設定したプラグアンドプレイ情報を用いて，実際の各 RT コンポーネントを制御する機能を実現している. プラグアンドプレイ機能を実現するために，対象システムを構成する各コンポーネントのアクション実行順序，アクション実行条件を設定することができるようになっている. また，設定したプラグアンドプレイ情報に基づいて，指定された実行順序でアクションを実行するとともに，制約条件が成立していない場合のアクションの実行待機などを制御する機能も実現している.	(5)達成
④RT システム開発支援ツールの開発（出典：基本計画 p8）		
(1) よりエンドユーザーに近い立場の人が，RT コンポーネントの仕様から，できるだけノンプログラミングで RT システムとして提供すべきサービスを構築する機能.	(1) 本ツールは，対象となる RT システム内に配置された RT コンポーネントの仕様情報(RTCProfile)から，システム全体の振る舞いを定義する機能を実現している. また，作成した振る舞いに従って，システム全体を制御する機能も実現している. 更に，システムの振る舞い定義は，グラフィカルなエディタを用いて，プログラミングレスで行うことが可能である.	(1)達成
(2) 定義済みのサービス情報を再利用し，効率的にシステム構築を行うための機能.	(2) 既存の状態遷移定義を再利用したり，粒度が細かくより詳細な状態遷移定義から，より広い視点での状態遷移を定義するために，複数の状態遷移をマージする機能も実現している. この機能を利用することで，ユーザーは類似のサービスを構築する際に，過去の資産を再利用することができ，開発効率を向上させることができる. また，定義した振る舞い，サービスを実現するために必要となる各 RT コンポーネント間の接続情報を RT システム仕様記述方式（RTSProfile）の形式で自動生成することができるため，対象システムを容易に構築することが可能となる.	(2)達成

(a-4)　RT 要素部品管理モジュール用組み込み RT ミドルウェアの研究開発

目標	研究開発成果	達成度
(1)OpenRTM-aist の組込み MPU 向け RT ミドルウェアの開発(出典:基本計画 p9)	(1)TOPPERS 版 OpenRTM-aist として SH2 および ARM といった組込み MPU ボードに実装し動作確認を行った。	(1)達成
(2)住宅システムにおけるシステム設計ならびに、それによって構築されたシステムのスループット等の評価(出典:基本計画 p9)	(2)ホームコントローラの集中制御ではなく、その下部に位置する要素部品管理モジュールが各設備に設置する基盤通信モジュールを管理し、その基盤通信モジュールに RT コンポーネントが動作することで、分散処理系を構成し、住宅システムの安定性を向上させ、動作検証を行った。詳細は c-1-6-4-c 節に記載。	(2)達成
(3)住宅用実証 RT システムの構築並びに住宅システムとしての評価(出典:基本計画 p9)	(3)産業技術総合研究所の実験室内ある住宅モデル、およびミサワホーム展示場のモデルハウス内に開発した RT システムを導入し、システム全体として①RT システムとしての動作確認、②RT 要素部品のプラグアンドプレイ機能確認、③PLCにおける基盤通信モジュール間のネットワーク動作、④住宅システム内の RT 要素部品の安定動作、⑤通信モジュールの待機消費電力の評価、以上の 5 つの評価を行った。 詳細は c-1-6-4-d 節に記載	(3)達成

1-1-b 基盤通信モジュールを用いた RT 要素部品の開発

本研究開発課題においては、1-1-a 節において開発された要素部品管理モジュールおよび基盤通信モジュールを介して動作する住宅内に設置する設備に係るハードウェアおよびソフトウェアの開発を行った。

以下、3つの課題に分けて進められた。

(b-1) 小型通信ドライバモジュールと小型リニアアクチュエータによるRT要素部品の開発
(b-2) 環境情報計測用センサ要素部品の開発
(b-3) 分散する RT 要素部品の RT コンポーネント開発とコンポーネント作成ツールの評価

以下にそれぞれの開発目標に対する研究開発成果および達成度をまとめる。

(b-1) 小型通信ドライバモジュールと小型リニアアクチュエータによるRT要素部品の開発

目標	研究開発成果	達成度
①RTC－Lite フレームワークに準拠した小型通信ドライバモジュールの開発		
(1)小型の CAN による通信機能を有した小型モータ向けモータコントローラドライバのラインナップを，各モータタイプにおいて開発(出典:基本計画 p10)	(1)DC モータ、DC ブラシレスモータ、ステッピングモータに対応したドライバを開発した。	(1)達成
(2)上記各種ドライバのインタフェースは共通機能とする。同一 CPU、A/D 4ch、I/O 4ch(エンコーダ入力対応)、CAN 通信機能、電源電圧(8～24V)、単一電源 回路消費電流 40mA 程度、モーション登録機能、基板	(2)I/O、A/D、CAN の機能を確認、目標としたサイズより 2mm、重量で 1g 超過したが機能的問題はない。	(2)達成

サイズ 36x23x7mm 重量 10g以下 を目標(出典:基本計画 p10) (3)小型モータコントローラドライバのプロトコルを変換・中継する、RT ミドルウェアとの共通インターフェースを持つ RTC-Lite フレームワークにあわせた、小型通信ドライバモジュールを開発する。(出典:基本計画 p11)	(3) 小型モータ向けモータコントローラドライバに RTC を適用し、他の RTC との接続を確認した。	(3)達成
②小型通信ドライバモジュールのパラメータエディタ RTC の開発 (1) 各小型基盤通信モジュールへの、モータパラメータや一連の動作を登録するためのパラメータエディタRTCの開発を行う(出典:基本計画 p11)	(1) 各種パラメータおよび動作の設定が可能とするパラメータエディタ RTC を開発した。	(1)達成
③RT 要素部品として小型リニアアクチュエータの開発 (1)小型・安価・簡易でかつ安全装置が付加されたリニアアクチュエータの開発を行う(出典:基本計画 p12)	(1)軸長 50mm、ストローク 30mm、軸径 16mm、瞬時最大推力 50N、リミットセンサ具備、量産時見込み価格 19、800 円。小型通信ドライバモジュールから簡単に制御可能	(1)達成
④RT 要素部品のラインナップ (1)RT システムとして使用の可能性の高いモータを選び、それぞれのモータに合わせた小型通信ドライバモジュールを用意し、パラメータエディタを用いて簡単に設定できるようにする(出典:基本計画 p12)	(1) THK 製アクチュエータを RT 要素部品としてラインナップした。「次世代ロボット向けエンドエフェクタ構成要素 SEED」	(1)達成
⑤窓サッシのインテリジェント化 (1)開発された RT 要素部品を利用し、ミサワホーム総合研究所と共に、インテリジェントウインドウシステムを設計・試作する(出典:基本計画 p12)	(1)小型通信ドライバモジュールおよび小型リニアアクチュエータを適用し、インテリジェントウインドウシステムを構築した。	(1)達成
(2)実証 RT システムとしてネットワークと結合しシステム化することで、全体システムからみた RT 要素部品の評価を行う(出典:基本計画 p12)	(2)基盤通信モジュールを介してネットワーク内での動作を確認。	(2)達成
(3)窓の開閉を補助するアシスト機能およびコントローラより施錠を可能とする自動施錠機能を有する RT 要素部品を開発する。(出典:基本計画 p12)	(3)アシスト機能部品、自動施錠部品を開発し、インテリジェントウインドウシステムに組込み、動作を確認。	(3)達成

(b-2) 環境情報計測用センサ要素部品の開発

目標	研究開発成果	達成度
(1) 既存のセンサに基盤通信モジュールを付加することでRTミドルウエア上に参加可能な温湿度、人感、照度などの各センサの RT 要素部品化したハードウエアの開発を行う。 安価な RT 要素部品化に使用する Zigbee エンドポイントの原価を 2,000 円以下にする。（出典:基本計画 p13）	(1) 温湿度センサ、人感センサ、照度センサなどの RT 要素部品化を図り、RTミドルウエア上に参加可能にするための Zigbee エンドポイントのモジュール開発を行い、関係機関に配布した。 ・センサーなど安価な RT 要素部品化に使用の Zigbee エンドポイントの原価は 2,000 円以下を実現した。 ・Zigbee エンドポイントは名刺サイズの 1/8 を実現した。 ・低消費電力化については消費電流を 980μ A を実現して電池駆動を可能にした。	(1)達成

(b-3) 分散する RT 要素部品の RT コンポーネント開発とコンポーネント作成ツールの評価

目標	研究開発成果	達成度
①デバイスベンダより提供されるデバイスに応じた RT コンポーネント開発		
(1) デバイスの推奨する制御方法とRTミドルウェアフレームワークとの間での整合性を取りながら，安定して動作するためのコンポーネント設計，インタフェース設計，各デバイスベンダへのコンポーネント供給を行う.（出典:基本計画 p13）	(1) オカテック社のモータドライバを対象とし，モータドライバのコントローラへの組み込み RTM の実装，および実装を通じたフィードバックを行うとともに，組み込み RTM 対応製品開発に関わるコスト試算を行った.	(1)達成
②RTC-Lite および開発ツールの検証およびその評価		
(1) RT 要素部品の実装を通じて，RTC-Lite の評価を行い，問題点を開発機関にフィードバックする. および RT 要素部品の実装時にテクノロジックアートより提供される開発支援ツールを用い，その評価および問題点のフィードバックを行う.（出典:基本計画 p13）	ミサワホーム住宅展示場システム，および産総研実証スペースにおけるシステムの実装を通じて，逐次問題点をフィードバックし，組み込み RT ミドルウェア開発の開発者視点からの利用事例の提供を行った.	(1)達成
③RTC-Lite を利用しない直接的な RT ミドルウェアネットワークの開発（出典:基本計画 p14）		
(1)CAN ネットワークも RT ミドルウェアに参加できるようなブリッジ機能を RT ミドルウェアに追加する.	(1)旭光電機社製エリア型人感センサにおいて，ブリッジユニットによる RT ミドルウェアへの対応を行った.	(1)達成

1-1-c RT 要素部品群による RT システムの開発・実証

本研究開発課題においては、1-1-b 節において開発された各種 RT 要素部品を住宅システムとして統合し、住宅の機能としてのシステム全体の設計および開発した RT ミドルウェアや、要素部品管理モジュールおよび基盤通信モジュール等の動作機能検証およびシステム評価を行った。住宅の機能としては、窓の自動開閉の連携動作によるインテリジェント空調システムと、窓の容易な開閉を支援するパワーアシスト機能および、外出時の施錠との連携動作による自動施錠といったセキュリティに係る機能を構築した。また、本住宅システムは、それぞれの RT 要素部品が各種機能に対して連携して動作するシステムであり、得に物理的な動作を伴っている。そのため、システム構成が複雑であり、一体型のシステムと異なり安全性の評価が難しい。これに対して、RT システムとしての安全性の検討手順を構築した。

具体的に、以下の3つの課題に分けて進められた。

(c-1) 住宅環境における RT 実証システムの研究開発
(c-2) プラグアンドプレイ機能を実現する統合ミドルウェアの開発
(c-3) 安全性の検討

(c-1) 住宅環境における RT 実証システムの研究開発

目標	研究開発成果	達成度
1) 分散型 RT 要素を利用した住宅用ホームオートメーションのビジネスモデルを検討. (出典:基本計画 p15)	(1)従来の住宅のビジネスモデルに基づき、RT が導入した場合のシステムの優位性を従来ある自動化の既製品と比較してまとめた。	(1)達成
(2)RT システムを住環境に応用する場合にハードウェアに求められる要件,そして RT の専門家以外の技術者或いは居住者によるアプリケーション開発を実現する為に開発環境に求められる要件をまとめる. (出典:基本計画 p15)	(2) RT システムに要求される仕様を、ユーザー企業となる住宅メーカーからの見地からまとめた。	(2) 達成
(3) RT システムを構築する際の仕様 (住宅用設備開発フレームワーク等) を固める. (出典:基本計画 p16)	(3)住宅メーカーが顧客から要求されているニーズをまとめ、その中で RT システムで実現可能な機能を抽出した。	(3)達成
(4) 実証システムとして提示している「住宅用インテリジェント空調システム」「インテリジェント・ウィンドウ (窓) システム」のシステム設計仕様を固める。(出典:基本計画 p16)	(4)実証システムとして構築する「住宅用インテリジェント空調システム」「インテリジェント・ウィンドウ (窓) システム」について連携に必要な設備機器 (RT 要素部品) をまとめ、システム設計を行った。	(4)達成
(5) RT 要素部品および開発環境を利用し,実証システムを構築し評価を行う。(出典:基本計画 p16)	(5) 実証システムとして産業技術総合研究所実験室内のモデルルームおよびミサワホーム住宅展示場モデルハウスに導入し、住宅システム全体としての評価を行った。	(5)達成

(c-2) プラグアンドプレイ機能を実現する統合ミドルウェアの開発

目標	研究開発成果	達成度
(1) プラグアンドプレイ機能や RT 要素部品のステータス管理機能などを有する汎用 PC もしくは組み込み MPU を利用したホームコントローラ上で動作する基盤通信モジュールの統合ミドルウェアを実現する。また、基盤通信モジュール上の RTC-Lite フレームワークが、OpenRMT-aist と相互運用可能なように、SH4 / ARM9 200MHz 相当の RT 要素部品管理モジュール上に ProxyRTC 機能を実現する。統合ミドルウェアの動作環境として、Linux および Windows の 2 種類の OS をサポートする。(出典:基本計画 p17)	(1)プラグアンドプレイや RTC のステータス管理機能を持つ統合ミドルウェアを開発した。	(1)達成
(2) 統合ミドルウェアおよび ProxyRTC の実証評価版は、平成 21 年度中に RT システム開発機関に無償で提供し、RT システムの開発に供する。この際、統合ミドルウェアと ProxyRTC だけではなく、仕様書および取扱説明書もあわせて提供する。(出典:基本計画 p17)	(2) 統合ミドルウェアのソフトウェア、ドキュメントをプロジェクト実施者に提供し評価した。	(2)達成
(3) 実証評価の結果をフィードバックし、統合ミドルウェアと ProxyRTC の最終版をプロジェクト終了時までにリリースする。(出典:基本計画 p17)	(3) フィードバックした最終版をリリースした。	(3)達成
(4)22 年度に構築する実証 RT システムについては、株式会社ミサワホーム総合研究所で有する顧客からの要求仕様を元に、株式会社セックが必要な要素部品を構成するシステムの仕様書を作成し、事業主体として各委託機関からの RT 要素部品や関連ソフトウェアをとりまとめて構築する。(出典:基本計画 p17)	(4)産総研およびミサワホーム展示場に実証フィールドを構築し、実証 RT システムを評価した。	(4)達成
(5)住宅価格に対して RT システムの価格が 5%以下、例えば 2400 万円(床面積 40 坪、坪単価 60 万円)の住宅の場合、RT システムの価格を 120 万円以下とする事を目標とする。(出典:基本計画 p17)	(5) 本目標は実現可能と判断する。ただし、導入する RT 要素部品や RT システムによる。	(5)達成

(c-3) 安全性の検討

目標	研究開発成果	達成度
1) 本実証 RT システムに対しての安全性の検討手順を構築する. (出典:基本計画 p17)	(1)SysML を利用し、システムを見える化し、構造図に基づくリスク評価手順を示した。	(1)達成
(2) プロジェクト終了後, 開発成果の一つとして, 様式を含めた安全性チェック手順を公開する. (出典:基本計画 p17)	(2) 報告書で公開を行う。	(2)達成見込み(2011 年 8 月)

2.研究開発項目毎の成果

(a). 「基盤通信モジュールおよび開発ツールの開発」

(a-1) RT 要素部品管理モジュールおよび基盤通信モジュールの研究開発

（委託先：株式会社アルゴシステム）

a-1-1 概要

基盤ロボット技術活用型オープンイノベーション促進プロジェクトのハードウエア開発として「RTC-Lite フレームワークに基づく基盤通信モジュール RT ミドルウエアの研究開発」と「RT 要素部品群によるRTシステムの開発・実証」が実現できること及び、従来 RT ミドルウエアの動作として必要であった PC に変わるものとして、RT 要素部品が容易に RT ミドルウエア上に参加できる低価格での分散型 RT 要素部品管理モジュール及び基盤通信モジュールの「実動作ハードウエア開発」を行った。

平成21年度の RTC-Lite フレームワークに基づく基盤通信モジュール・RTミドルウエアの研究開発では、産業技術総合研究所で開発した RTC-Lite フレームワークをベースとしてOMGのRTC仕様に準拠し、OpenRTM-aist と相互運用可能なRT要素部品管理モジュール、基盤通信モジュール、Zigbee ステーション、Zigbee センサモジュールの Ethernet タイプの開発及び試作品の製作を行った。

平成22年度は住宅内で利用する可能性の高い通信プロトコルの高速 PLC 版のRT要素部品管理モジュール、基盤通信モジュール、Zigbee ステーション、Zigbee センサモジュールのハードウエア開発を行った。また、RT 要素部品管理モジュールと RT 要素部品間の多様な通信プロトコルをも許容するためのブリッジ機能を RT 要素部品管理モジュールに組み込むと共に、RT 要素部品とのインターフェースの融通性向上のため、CAN 版基盤通信モジュールと Zigbee 版基盤通信モジュール（Zigbee ステーション）を開発した。これらにより、ホームネットワークとしての拡張性を広げることが出来た。これら、開発した各モジュールを住宅用実証 RT システムに提供してモジュールの総合評価を行った。

本件での目標、並びに達成度は表 a-1-1 の通りである。また、開発したモジュールの階層と構成を図 a-1-1 に示す。

表 a-1-1　研究項目に対する目標並びに達成度

目標	研究開発成果	達成度
RT 要素部品管理モジュールおよび基盤通信モジュールの研究開発 （1）RT 要素部品が容易に RT ミドルウエア上に参加できる低価格での分散型 RT 要素部品管理モジュール及び基盤通信モジュールの実動作ハードウエアの開発。 *具体的目標 ・コストの高い RT 要素部品用の基盤通信モジュール原価は 20,000 円を目指す。 ・基盤通信モジュールのサイズを名刺半分に する。	（1）Ethernet 版及び PLC 版の RT 要素部品管理モジュールと基盤通信モジュールの開発と提供を行った。また、多様な通信プロトコルを許容するためのブリッジ機能を RT 要素管理モジュールに組込み,ホームネットワークシステムとしての拡張性を広げた。 ・要素部品、基盤通信、Zigbee 各 モジュールは 20,000 円以内実現。 ・基盤通信モジュールは名刺サイズ 2/3 を実現。尚、量産時に BGA 部品使用で名刺サイズの1/2可能性あり。	（1）達成 サイズに関しては量産時に名刺半分サイズを実現

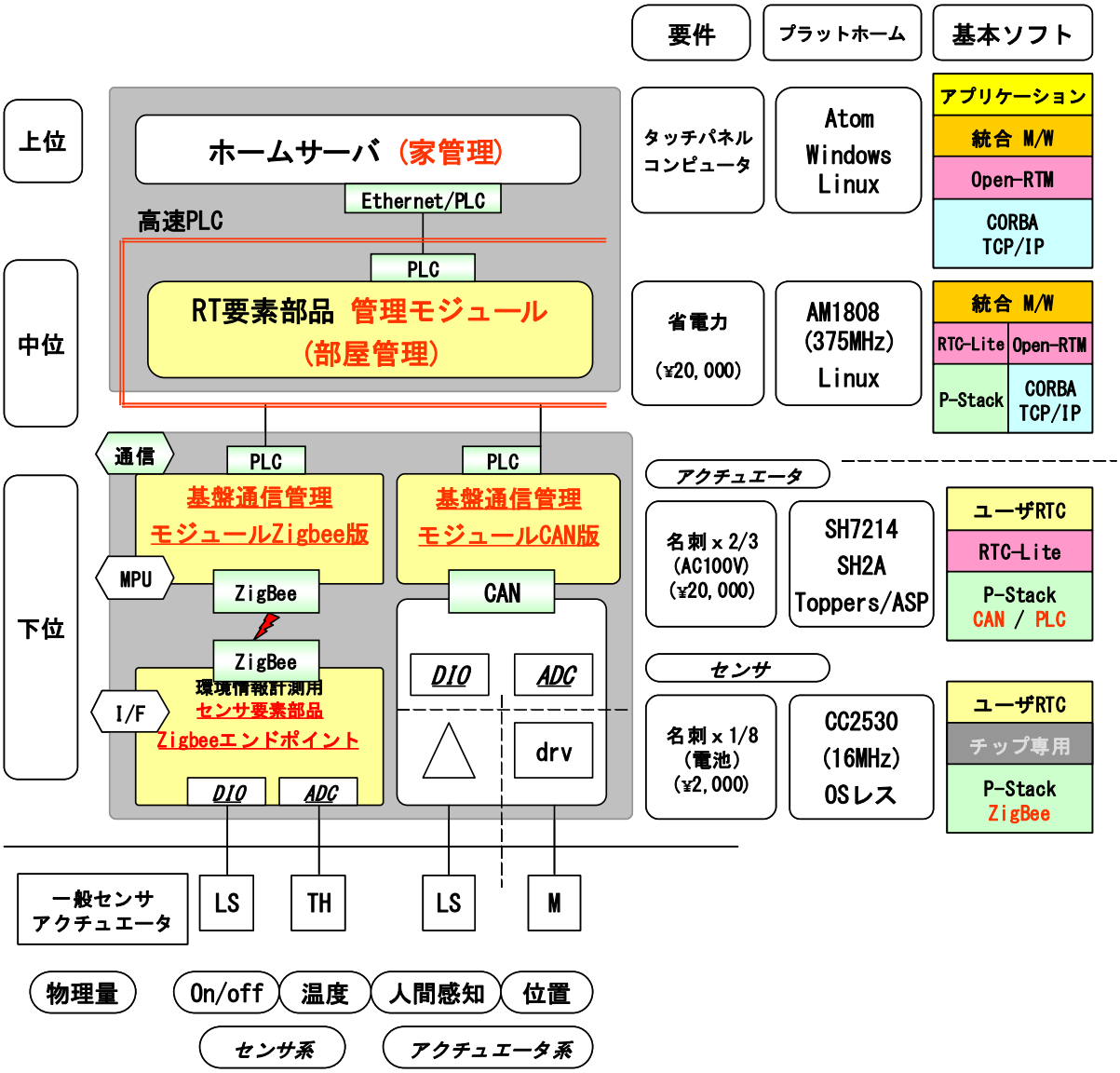


図 a-1-1　モジュールの階層と構成

a-1-2 RT 要素部品管理モジュールの開発

1)概要と特徴

- 本製品は CPU にAM1808 375MHz ARMコアを搭載。
- PLC制御にアセロス製HomePlug AV ChipSetを搭載。
- 保存領域としてNAND FLASH 2GByteを搭載。
- 基板はCPU部と電源/PLC部に分割。

本製品の特長を以下に示す。

① CPUはTI製ARM CPUであるAM1808(375MHz)を搭載
② DDR2 256MByte搭載
③ NAND FLASH 2GByte搭載(SATA接続)
④ 高速PLC用に、アセロス製 INT6400／INT1400を搭載
⑤ RTC搭載(二次電池によるバッテリバックアップ機能付き)
⑥ シリアルインタフェース 1chを搭載(コンソール用)

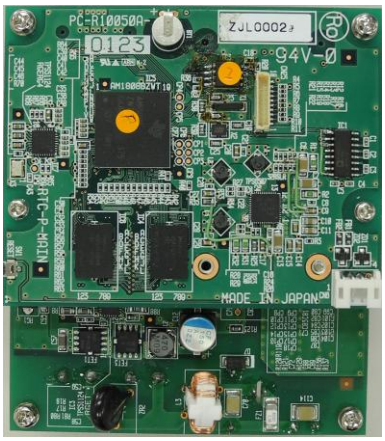


図 a-1-2 RT 要素部品管理モジュール

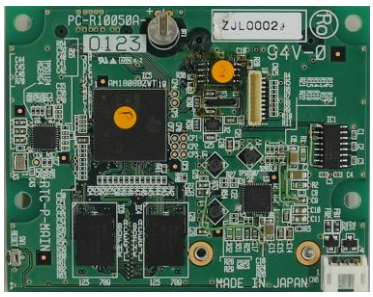


図 a-1-3 CPU部

図 a-1-4 電源/PLC 部

2)仕様

・ 電気仕様

表 a-1-2　電気仕様

項　目		仕　様
電 源	定格電圧	AC100V
	電圧許容範囲	AC85～132V
	許容瞬時停電時間	1ms 以下
	内部消費電力	5W 以下
	ｽﾃｰﾀｽ LED（POWER）	ﾌﾞﾙｰ

・ 環境仕様及び質量

表 a-1-3　環境仕様及び質量

項　目		仕　様
物理的環境	使用周囲温度	0～50℃(取付け角度による制限有り)
	保存周囲温度	-20～70℃
	使用周囲湿度	30～90%RH(結露無きこと)
	保存周囲湿度	30～90%RH(結露無きこと)
	使用雰囲気	腐食性ｶﾞｽ無きこと

・ 機能仕様

表 a-1-4　機能仕様

項　目	仕　様
CPU	Texas Instruments 社製 CPU AM1808 375MHz
RAM	DDR2　256Mbyte
ROM	NAND FLASHROM 2Gbyte(m-SATA ﾓｼﾞｭｰﾙ)
PLC	ｱｾﾛｽ社製　INT6400/INT1400　ChipSET 200 Mbps PHY rate
RTC	CPU 内蔵の Real Time Clock を使用 (ﾘﾁｳﾑ 2 次電池は 8 時間以上充電が必要)
RS-232C	1ch(Max38400bps)　4 ﾋﾟﾝ　ｺﾈｸﾀ制御信号なし

3)各部の名称

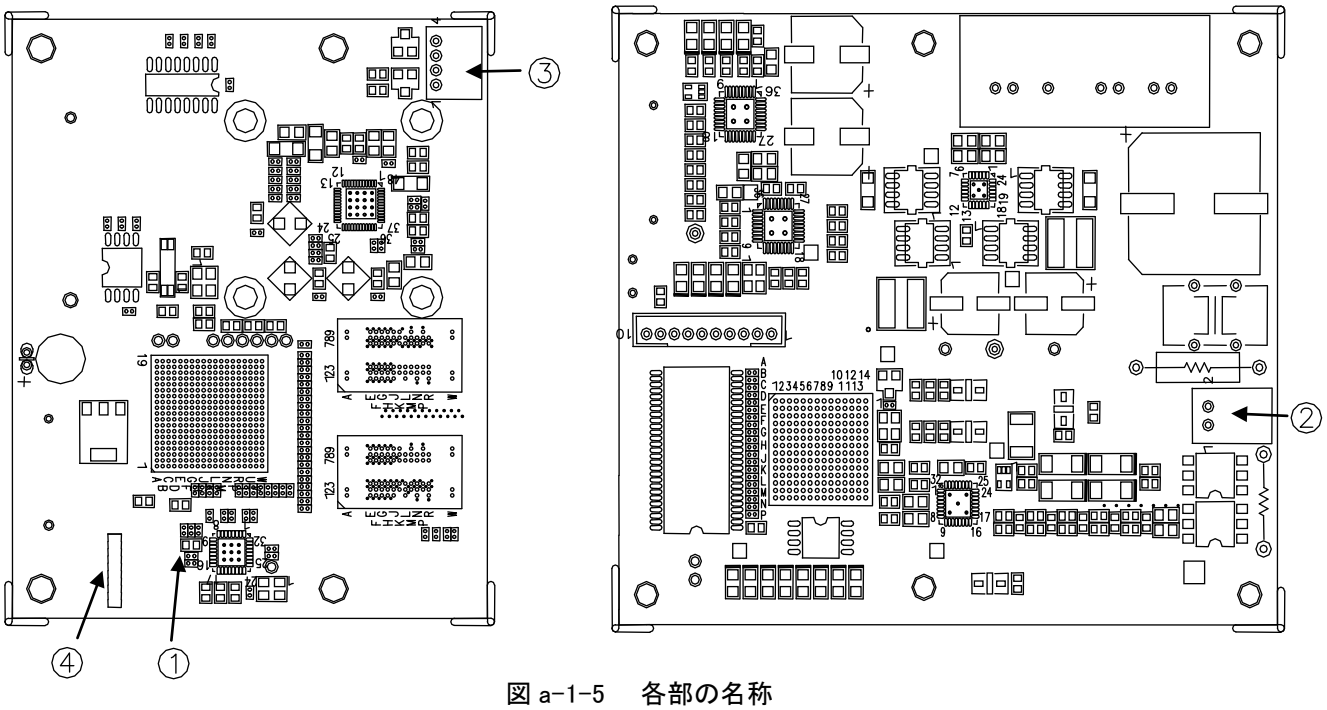


図 a-1-5　各部の名称

表 a-1-5　名称の説明

<table>
<tr><th>No.</th><th>名　称</th><th>内　容</th></tr>
<tr><td>①</td><td>POWER LED</td><td>電源投入後しばらくすると LED が点灯します</td></tr>
<tr><td>②</td><td>AC100V 電源ｺﾈｸﾀ</td><td>
<table>
<tr><td>1</td><td>AC(L)</td></tr>
<tr><td>2</td><td>AC(N)</td></tr>
</table>
適合ｺﾈｸﾀ　　:　XHP-2　　（JST 製）
適合電線ｻｲｽﾞ　:AWG#30～AWG#22
</td></tr>
<tr><td>③</td><td>ｺﾝｿｰﾙｺﾈｸﾀ</td><td>ｺﾝｿｰﾙ用ｺﾈｸﾀです
<table>
<tr><td>1</td><td>TXD</td></tr>
<tr><td>2</td><td>RX</td></tr>
<tr><td>3</td><td>NC</td></tr>
<tr><td>4</td><td>GND</td></tr>
</table>
38400bps,ﾃﾞｰﾀ長 8bit,ﾊﾟﾘﾃｨなし,ｽﾄｯﾌﾟ 1 の設定でﾀｰﾐﾅﾙと接続します

適合ｺﾈｸﾀ　　:PHR-4 (JST 製)
適合電線ｻｲｽﾞ　:AWG#30～AWG#22
</td></tr>
<tr><td>④</td><td>JTAG ｺﾈｸﾀ</td><td>ﾃﾞﾊﾞｯｸﾞｺﾈｸﾀ
<table>
<tr><td>1</td><td>3.3V</td></tr>
<tr><td>2</td><td>TRST</td></tr>
<tr><td>3</td><td>TDI</td></tr>
<tr><td>4</td><td>TMS</td></tr>
<tr><td>5</td><td>TCK</td></tr>
<tr><td>6</td><td>RTCK</td></tr>
<tr><td>7</td><td>TDO</td></tr>
<tr><td>8</td><td>nRST</td></tr>
<tr><td>9</td><td>EMU0</td></tr>
<tr><td>10</td><td>GND</td></tr>
</table>
適合ｺﾈｸﾀ　　:SM10B-SRSS-TB(JST 製)
適合電線ｻｲｽﾞ　:AWG#30～AWG#26
</td></tr>
</table>

4)ソフトウエア構成

● **要素部品管理モジュール PLC版 ソフトウェア構成図**

要素部品管理モジュール PLC版のソフトウェア構成を下図に示します。

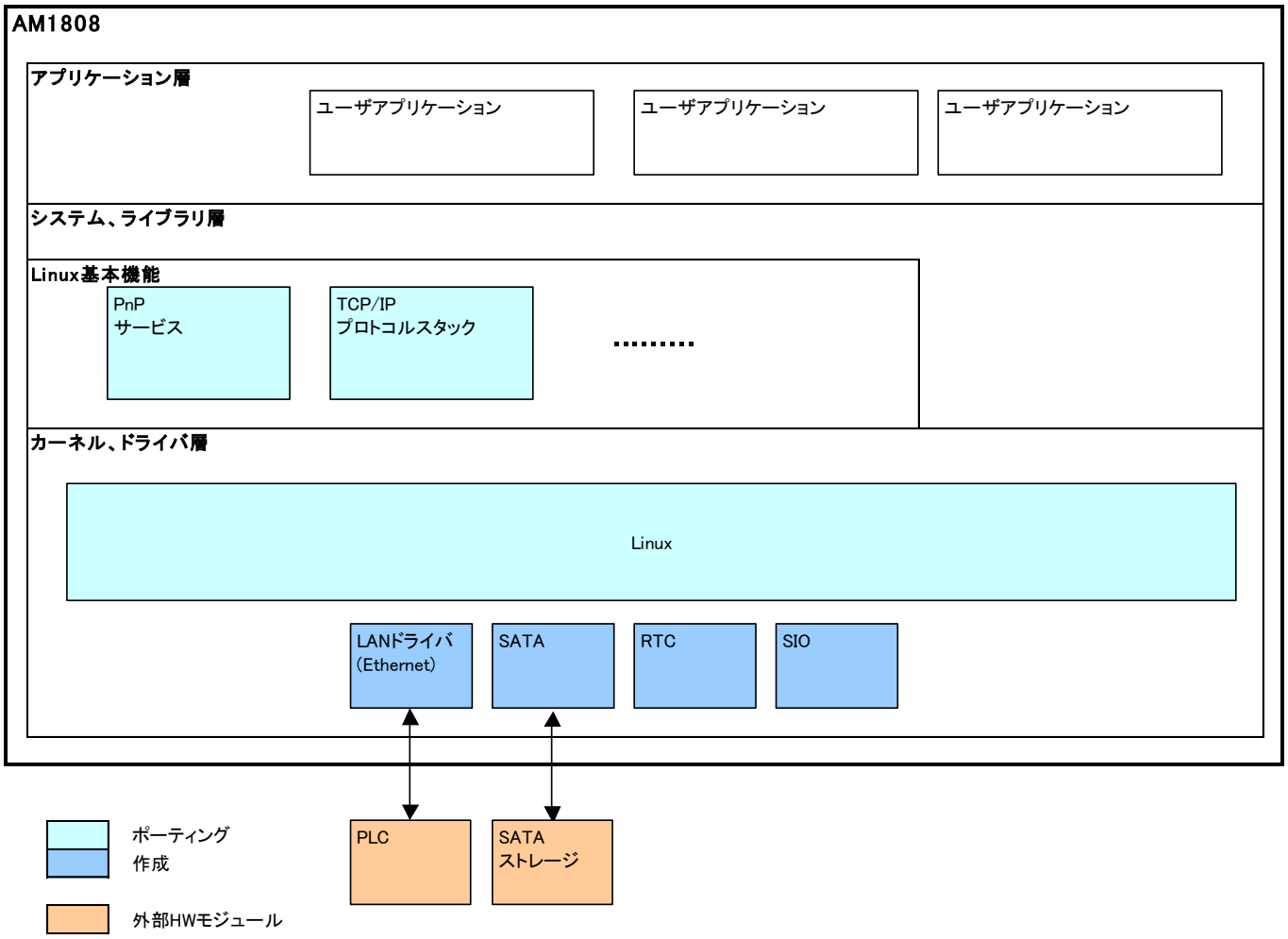


図 a-1-6　要素部品管理モジュール PLC 版ソフトウエア構成図

5)ブロック図

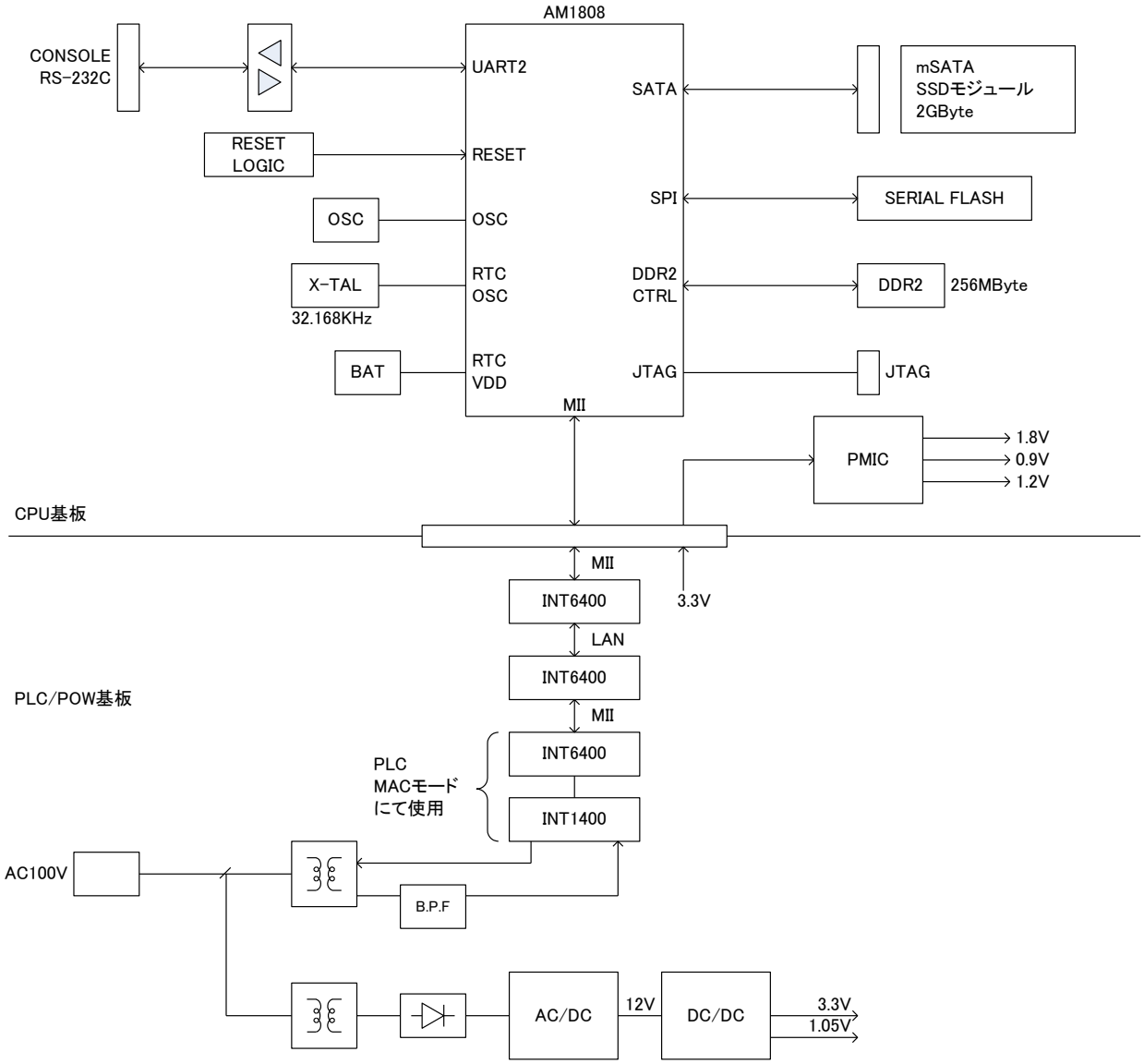
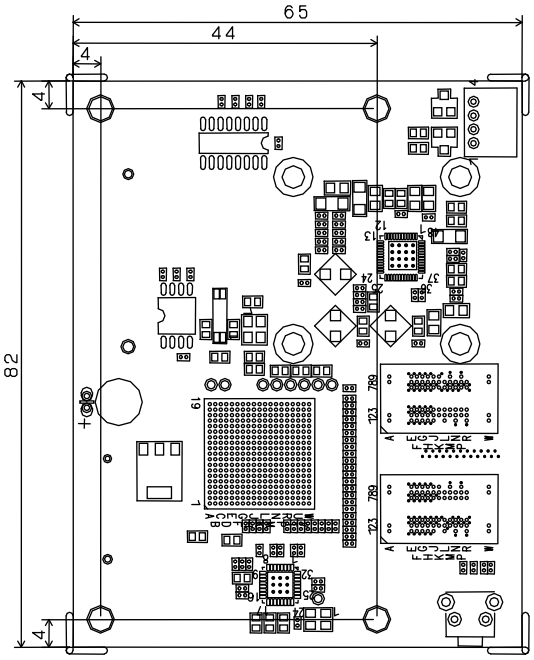


図 a-1-7　要素部品管理モジュール PLC 版ブロック図

6)外形寸法図・外観図

①外形寸法図

＊CPU 部



＊電源／PLC 部

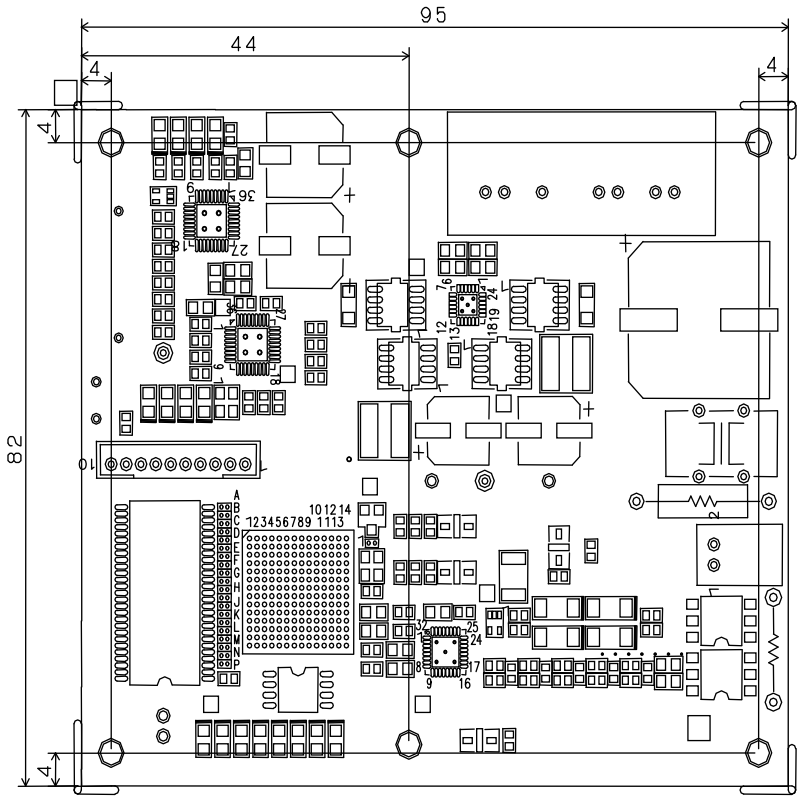


図 a-1-8　要素部品管理モジュール PLC 版外形寸法図

②外観図

（CPU 基板と PLC 基板を組み合わせたところ）

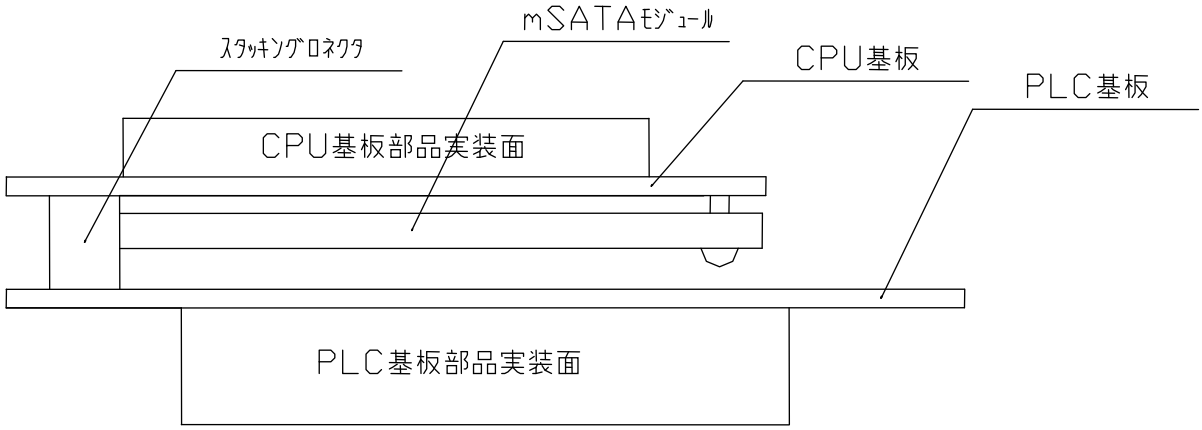


図 a-1-9　要素部品管理モジュール PLC 版外観図

a-1-3 CAN 版 基盤通信モジュールの開発

1)概要と特徴

- 本製品は CPU にSH7214 100MHz SH－2コアを搭載。
- 保存領域としてCPU内部に1MByteを搭載。
- PLC制御にアセロス製HomePlug AV ChipSetを搭載。

本製品の特長を以下に示す。

① CPUはルネサス製RISC CPUであるSH7214（100MHz）を搭載
② CPU内蔵メモリ 128KByte搭載
③ CPU内蔵FLASH 1MByte搭載
④ 高速PLC用に、アセロス製 INT6400／INT1400を搭載
⑤ シリアルインタフェース 1chを搭載（コンソール用）
⑥ CAN 1ch搭載

図 a-1-10 基盤通信モジュール（CAN 版）

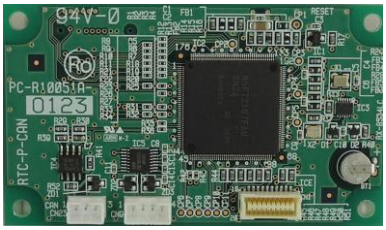


図 a-1-11 CPU 部

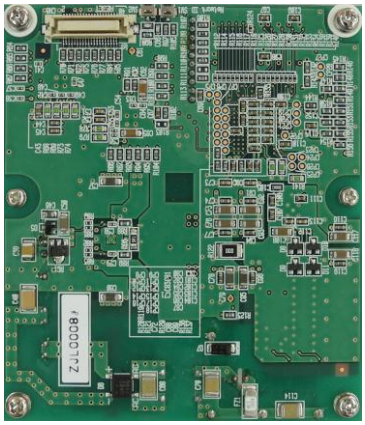


図 a-1-12 電源/PLC 部

2)仕様

・ 電気仕様

表 a-1-6　電気仕様

項　目		仕　様
電 源	定格電圧	AC100V
	電圧許容範囲	AC85～132V
	許容瞬時停電時間	1ms 以下
	内部消費電力	5W 以下

・ 環境仕様及び質量

表 a-1-7　環境仕様及び質量

項　目		仕　様
物理的環境	使用周囲温度	0～50℃(取付け角度による制限有り)
	保存周囲温度	-25～70℃
	使用周囲湿度	30～90%RH(結露無きこと)
	保存周囲湿度	30～90%RH(結露無きこと)
	使用雰囲気	腐食性ｶﾞｽ無きこと

・ 機能仕様

表 a-1-8　機能仕様

項　目	仕　様
CPU	ﾙﾈｻｽﾃｸﾉﾛｼﾞ SH-7214(SH-2) 100MHz
RAM	CPU 内蔵 128Kbyte
ROM	CPU 内蔵 1Mbyte
CAN	CPU 内蔵 1ch
PLC	Intellon INT6400/INT1400 ChipSET 200 Mbps PHY rate
RS-232C	1ch(Max38400bps) PH4ﾋﾟﾝ ｺﾈｸﾀ制御信号なし

3)各部の名称

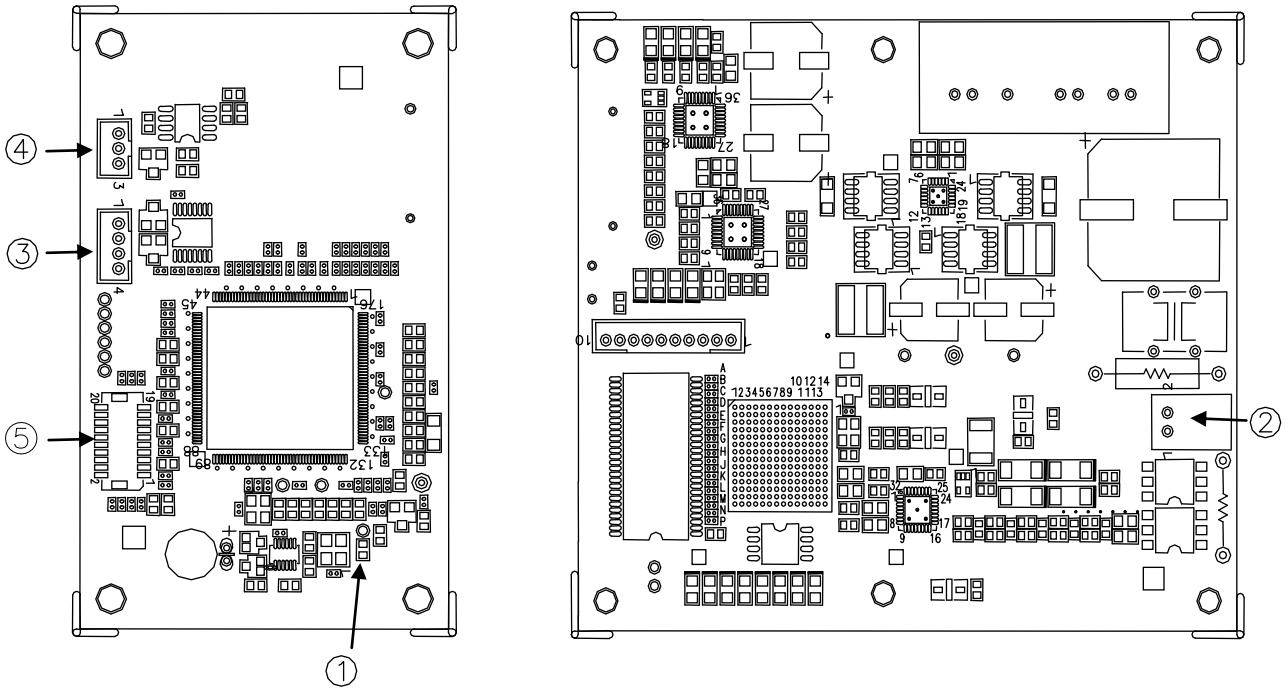
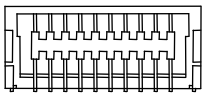


図 a-1-13　各部の名称

表 a-1-9　名称の説明

<table>
<tr><th>No</th><th>名　称</th><th>内　容</th></tr>
<tr><td>①</td><td>POWER LED</td><td>電源投入後しばらくすると LED が点灯します</td></tr>
<tr><td>②</td><td>AC100V 電源ｺﾈｸﾀ</td><td>
<table>
<tr><td>1</td><td>AC(L)</td></tr>
<tr><td>2</td><td>AC(N)</td></tr>
</table>
適合ｺﾈｸﾀ　　:XHP-2 (JST 製)
適合電線ｻｲｽﾞ　:AWG#30～AWG#22
</td></tr>
<tr><td>③</td><td>ｺﾝｿｰﾙｺﾈｸﾀ</td><td>ｺﾝｿｰﾙ用ｺﾈｸﾀです
<table>
<tr><td>1</td><td>XD</td></tr>
<tr><td>2</td><td>RXD</td></tr>
<tr><td>3</td><td>NC</td></tr>
<tr><td>4</td><td>GND</td></tr>
</table>
38400bps,ﾃﾞｰﾀ長 8bit,ﾊﾟﾘﾃｨなし,ｽﾄｯﾌﾟ 1 の設定でﾀｰﾐﾅﾙと接続します

適合ｺﾈｸﾀ　　:PHR-4 (JST 製)
適合電線ｻｲｽﾞ　:AWG#30～AWG#22</td></tr>
<tr><td>④</td><td>CAN ｺﾈｸﾀ</td><td>CAN 用ｺﾈｸﾀです
<table>
<tr><td>1</td><td>CAN0_H</td></tr>
<tr><td>2</td><td>CAN0_L</td></tr>
<tr><td>3</td><td>GND</td></tr>
</table>
適合ｺﾈｸﾀ　　:PHR-3 (JST 製)
適合電線ｻｲｽﾞ　:AWG#30～AWG#24</td></tr>
<tr><td>⑤</td><td>H-UDI ｴﾐｭﾚｰﾀ
接続ｺﾈｸﾀ</td><td>ﾙﾈｻｽ純正 ICE E10 を接続する為のｺﾈｸﾀです
ｺﾈｸﾀ変換ﾎﾞｰﾄﾞを介して ICE と接続します
19番ピン　1番ピン
 
20番ピン　2番ピン
<table>
<tr><td>1</td><td>+3.3V</td><td>2</td><td>TCK</td></tr>
<tr><td>3</td><td>GND</td><td>4</td><td>GND</td></tr>
<tr><td>5</td><td>RST</td><td>6</td><td>TDO</td></tr>
<tr><td>7</td><td>ASEBRK</td><td>8</td><td>TMS</td></tr>
<tr><td>9</td><td>TDI</td><td>10</td><td>RESET</td></tr>
<tr><td>11</td><td>MPMD</td><td>12</td><td>AUDSYNC</td></tr>
<tr><td>13</td><td>AUDATA0</td><td>14</td><td>AUDATA1</td></tr>
<tr><td>15</td><td>AUDATA2</td><td>16</td><td>AUDATA3</td></tr>
<tr><td>17</td><td>GND</td><td>18</td><td>GND</td></tr>
<tr><td>19</td><td>GND</td><td>20</td><td>AUDCK</td></tr>
</table>
適合ｺﾈｸﾀ　　:SHLDP-20V-S (JST 製)
適合電線ｻｲｽﾞ　:AWG#28～AWG#32</td></tr>
</table>

4)ソフトウエア構成

● 基盤通信モジュール PLC版 ソフトウェア構成図

基板通信モジュール PLC版のソフトウェア構成を下図に示します。

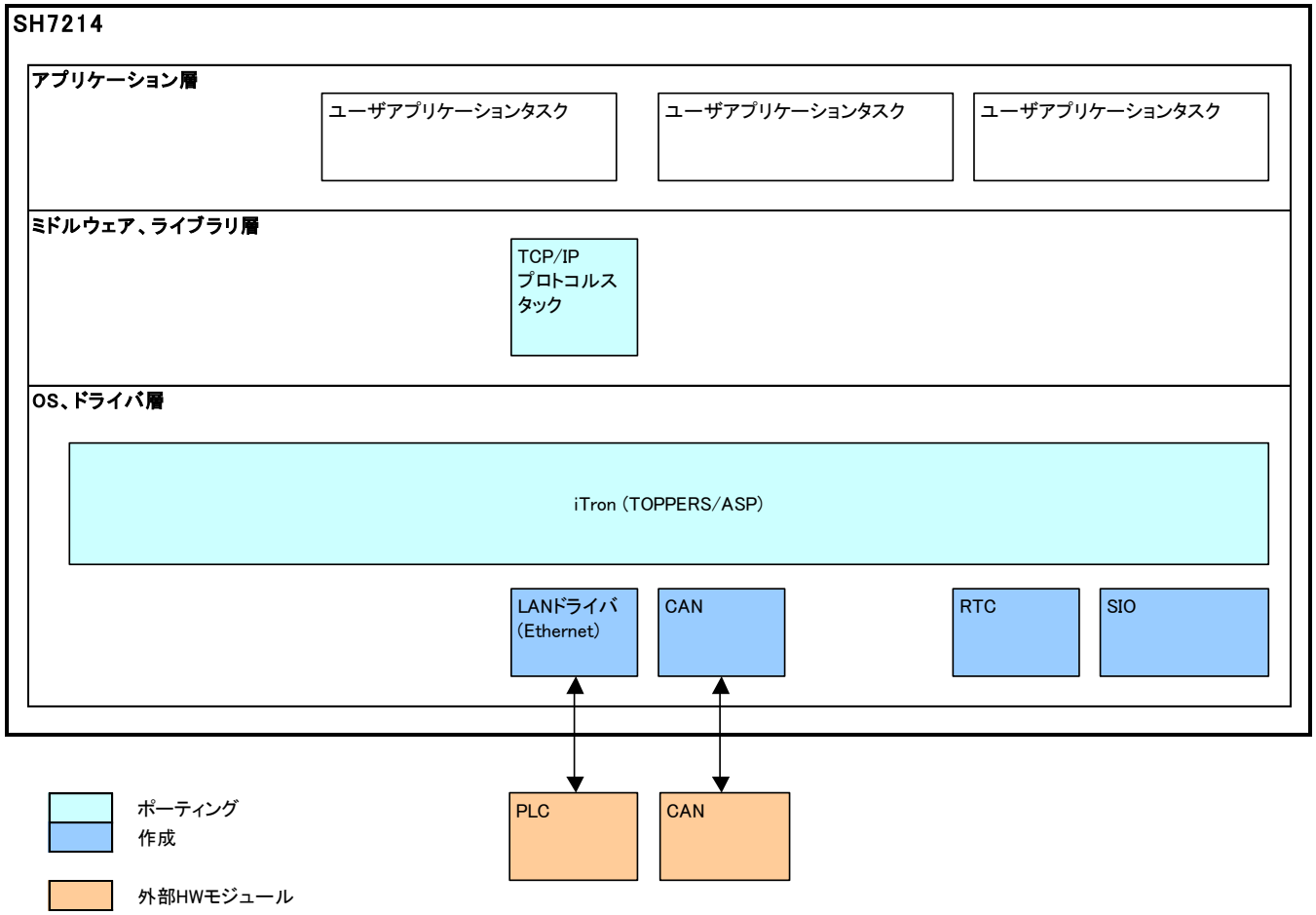


図 a-1-14　基盤通信モジュール PLC 版ソフトウエア構成図

5)ブロック図

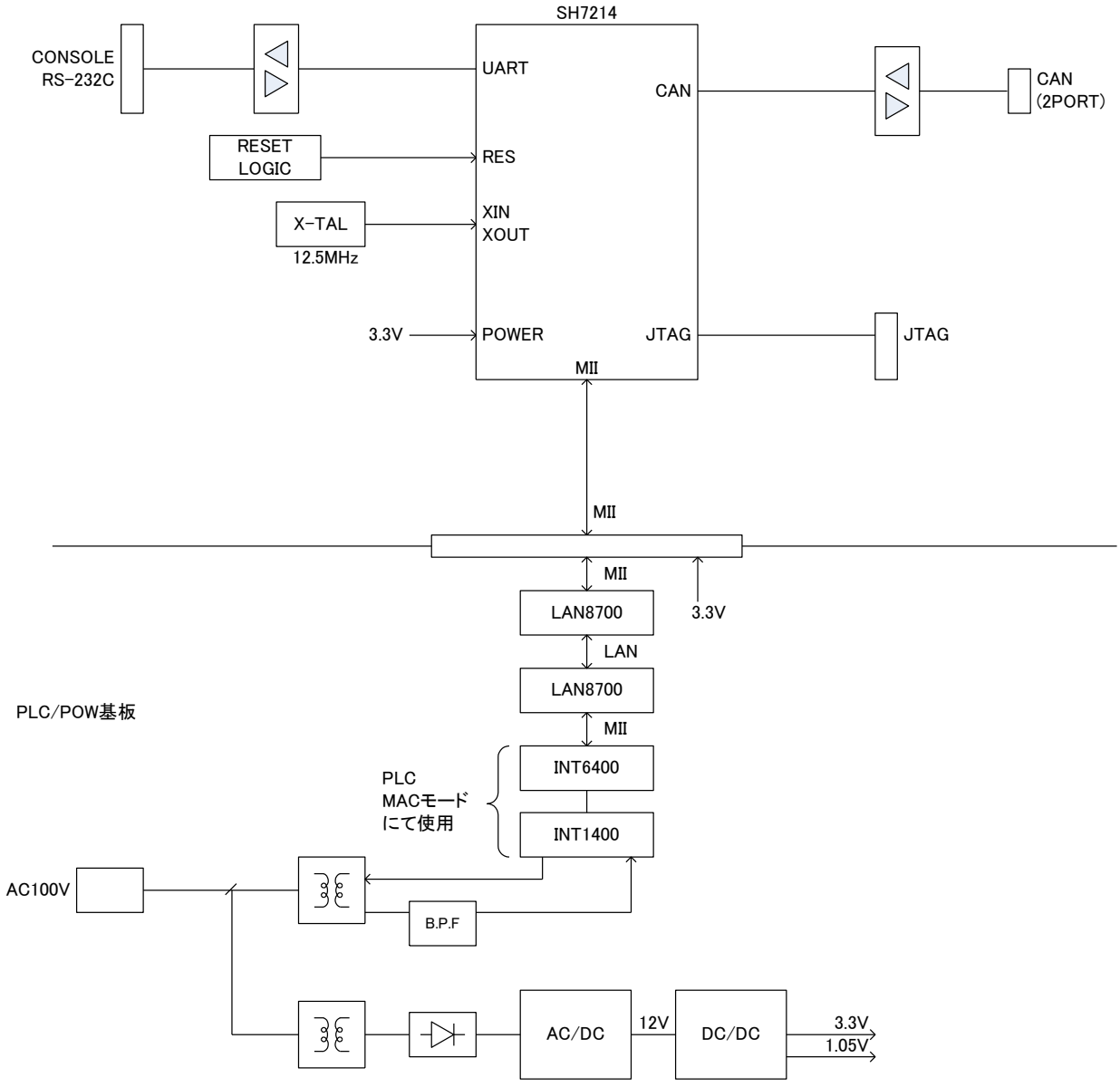
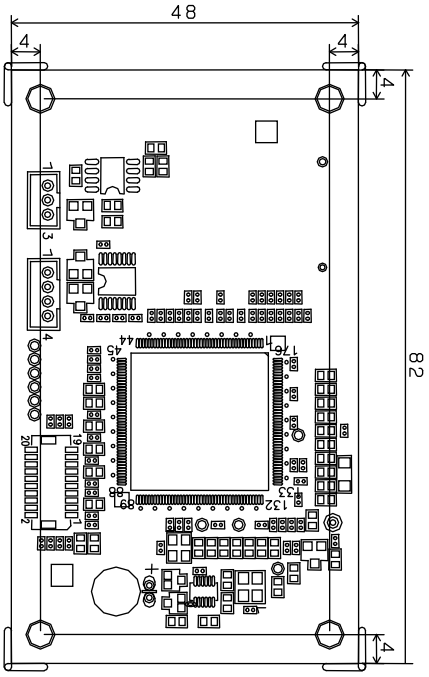


図 a-1-15　基盤通信モジュール PLC 版ソフトウエア構成図

6)外形寸法図・外観図

①外形寸法図

＊CPU 部



＊電源／PLC 部

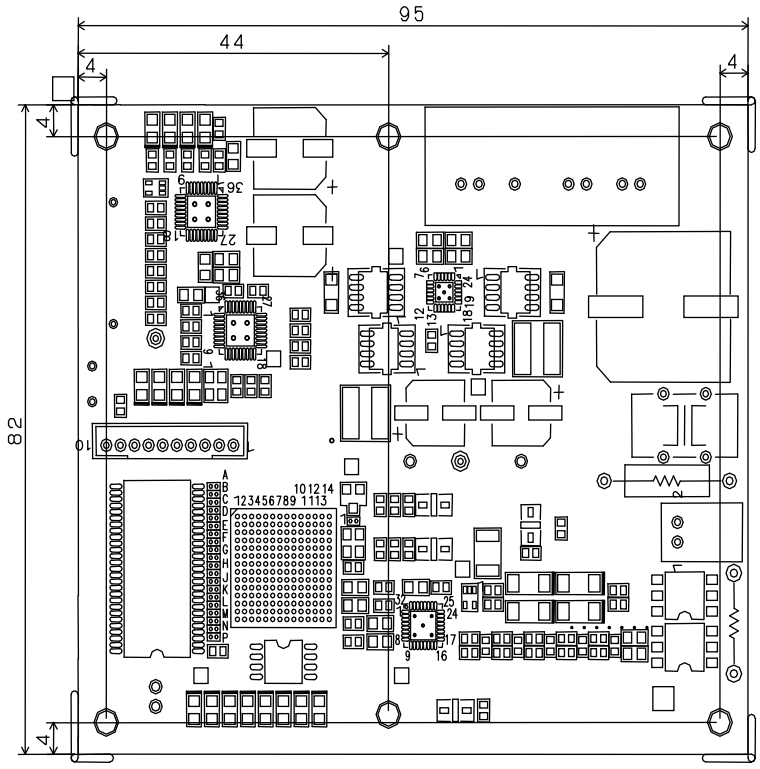


図 a-1-16　基盤通信モジュール PLC 版外形寸法図

②外観図

（CPU 基板と PLC 基板を組み合わせたところ）

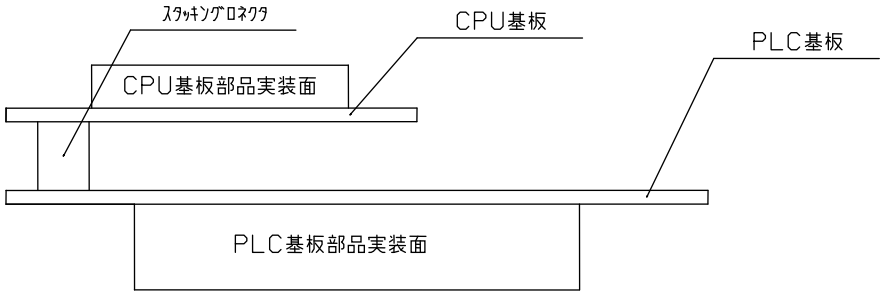


図 a-1-17　基盤通信モジュール PLC 版外観図

a-1-4 Zigbee 版 基盤通信モジュール(Zigbee ステーション)

1)概要と特徴

- 本製品は CPU にSH7214 100MHz SH－2コアを搭載。
- PLC制御にアセロス製HomePlug AV ChipSetを搭載。
- 保存領域としてCPU内部に1MByteが搭載。
- Zigbee 通信はTI製 CC2530を搭載。

本製品の特長を以下に示します。

① CPUはルネサス製RISC CPUであるSH7214(100MHz)を搭載
② CPU内蔵メモリ 128KByte搭載
③ CPU内蔵FLASH 1MByte搭載
④ 高速PLC用に、Intellon製 INT6400／INT1400を搭載
⑤ シリアルインタフェース 1chを搭載(コンソール用)
⑥ Zigbee通信用LSI CC2530(TI製)を使用、CPUとシリアルで接続

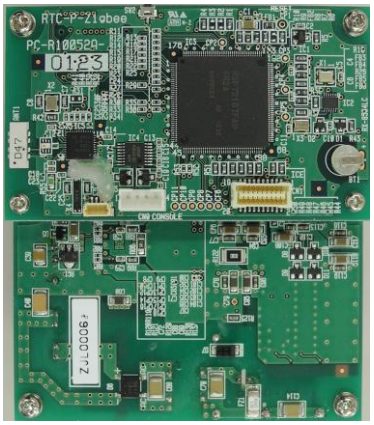


図 a-1-18 基盤通信モジュール(Zigbee 版)

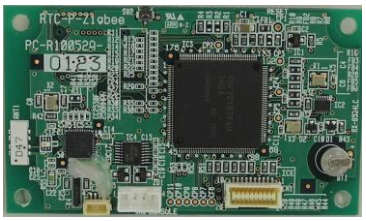


図 a-1-19 CPU 部

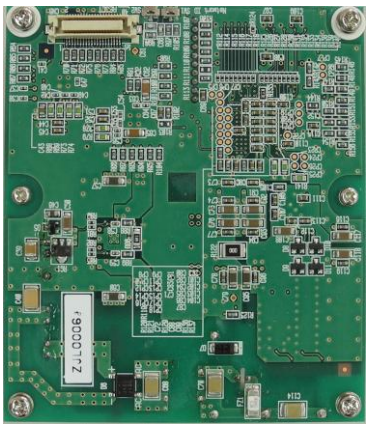


図 a-1-20 電源/PLC 部

2)仕様

・ 電気仕様

表 a-1-10 電気仕様

項 目		仕 様
電 源	定格電圧	AC100V
	電圧許容範囲	AC85～132V
	許容瞬時停電時間	1ms 以下
	内部消費電力	5W 以下

・ 環境仕様及び質量

表 a-1-11 環境仕様及び質量

項 目		仕 様
物理的環境	使用周囲温度	0～50℃(取付け角度による制限有り)
	保存周囲温度	-25～70℃
	使用周囲湿度	30～90%RH(結露無きこと)
	保存周囲湿度	30～90%RH(結露無きこと)
	使用雰囲気	腐食性ｶﾞｽ無きこと

・ 機能仕様

表 a-1-12 機能仕様

項 目	仕 様
CPU	ﾙﾈｻｽﾃｸﾉﾛｼﾞ SH-7214(SH-2) 100MHz
RAM	CPU 内蔵 128Kbyte
ROM	CPU 内蔵 1Mbyte
PLC	Intellon INT6400/INT1400 ChipSET 200 Mbps PHY rate
RS-232C	1ch(Max38400bps) PH4 ﾋﾟﾝ ｺﾈｸﾀ制御信号なし

3)各部の名称

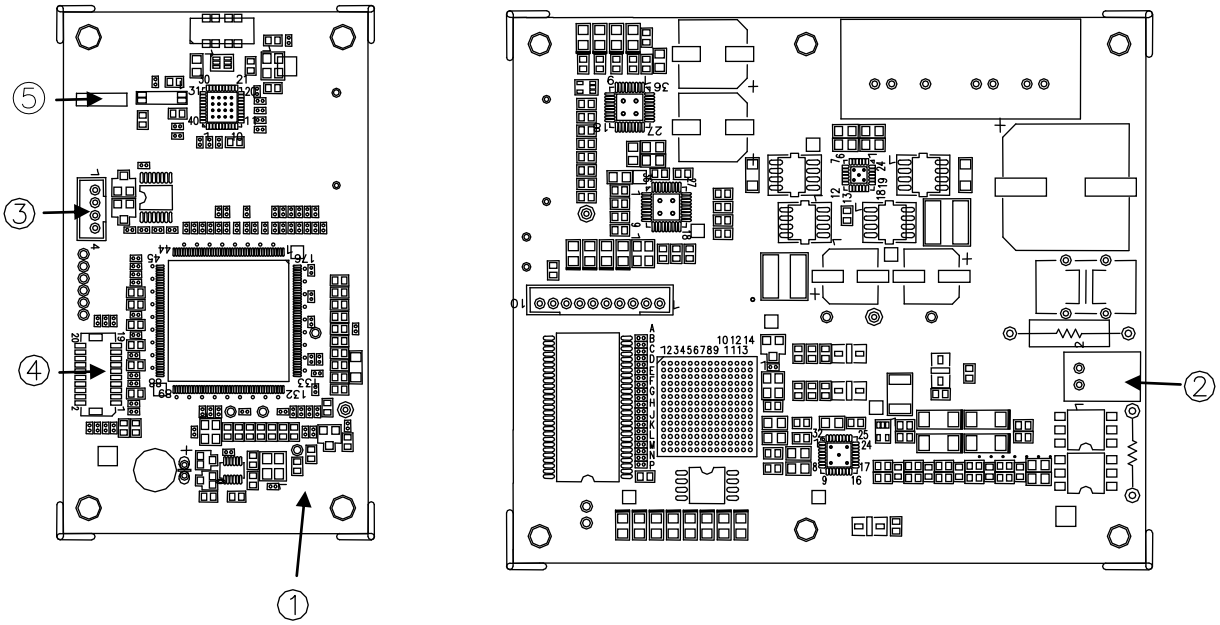
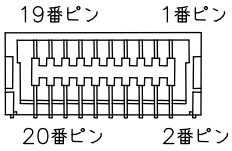


図 a-1-21　各部の名称

表 a-1-13　名称の説明

<table>
<tr><th>No.</th><th>名　称</th><th>内　容</th></tr>
<tr><td>①</td><td>POWER LED</td><td>電源投入後しばらくすると LED が点灯します</td></tr>
<tr><td>②</td><td>AC100V 電源ｺﾈｸﾀ</td><td>
<table>
<tr><td>1</td><td>AC(L)</td></tr>
<tr><td>2</td><td>AC(N)</td></tr>
</table>
適合ｺﾈｸﾀ　　:XHP-2 (JST 製)
適合電線ｻｲｽﾞ　:AWG#30～AWG#22
</td></tr>
<tr><td>③</td><td>ｺﾝｿｰﾙｺﾈｸﾀ</td><td>ｺﾝｿｰﾙ用ｺﾈｸﾀです
<table>
<tr><td>1</td><td>TXD</td></tr>
<tr><td>2</td><td>RXD</td></tr>
<tr><td>3</td><td>+5V</td></tr>
<tr><td>4</td><td>GND</td></tr>
</table>
38400bps,ﾃﾞｰﾀ長 8bit,ﾊﾟﾘﾃｨなし,ｽﾄｯﾌﾟ 1 の設定でﾀｰﾐﾅﾙと接続します
適合ｺﾈｸﾀ　　:PHR-4 (JST 製)
適合電線ｻｲｽﾞ　:AWG#30～AWG#22
</td></tr>
<tr><td>④</td><td>H-UDI ｴﾐｭﾚｰﾀ
接続ｺﾈｸﾀ</td><td>ﾙﾈｻｽ純正 ICE E10 を接続する為のｺﾈｸﾀです
ｺﾈｸﾀ変換ﾎﾞｰﾄﾞを介して ICE と接続します


<table>
<tr><td>1</td><td>+3.3V</td><td>2</td><td>TCK</td></tr>
<tr><td>3</td><td>GND</td><td>4</td><td>GND</td></tr>
<tr><td>5</td><td>TRST</td><td>6</td><td>TDO</td></tr>
<tr><td>7</td><td>ASEBRK</td><td>8</td><td>TMS</td></tr>
<tr><td>9</td><td>TDI</td><td>10</td><td>REST</td></tr>
<tr><td>11</td><td>MPMD</td><td>12</td><td>AUDSYNC</td></tr>
<tr><td>13</td><td>AUDATA0</td><td>14</td><td>AUDATA1</td></tr>
<tr><td>15</td><td>AUDATA2</td><td>16</td><td>AUDATA3</td></tr>
<tr><td>17</td><td>GND</td><td>18</td><td>GND</td></tr>
<tr><td>19</td><td>GND</td><td>20</td><td>AUDCK</td></tr>
</table>
適合ｺﾈｸﾀ　　:SHLDP-20V-S (JST 製)
適合電線ｻｲｽﾞ　:AWG#28～AWG#32
</td></tr>
<tr><td>⑤</td><td>Zigbee 用 ICE ｺﾈｸﾀ</td><td>Zigbee 用 ICE ｺﾈｸﾀです
<table>
<tr><td>1</td><td>TCK</td></tr>
<tr><td>2</td><td>TDA</td></tr>
<tr><td>3</td><td>RST</td></tr>
<tr><td>4</td><td>GND</td></tr>
</table>
適合ｺﾈｸﾀ　　:SHR-04V-S-B (JST 製)
適合電線ｻｲｽﾞ　:AWG#30～AWG#28
</td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

4)ソフトウエア構成

● **ZigBeeステーション PLC版 ソフトウェア構成図**

ZigBeeステーション PLC版のソフトウェア構成を下図に示します。

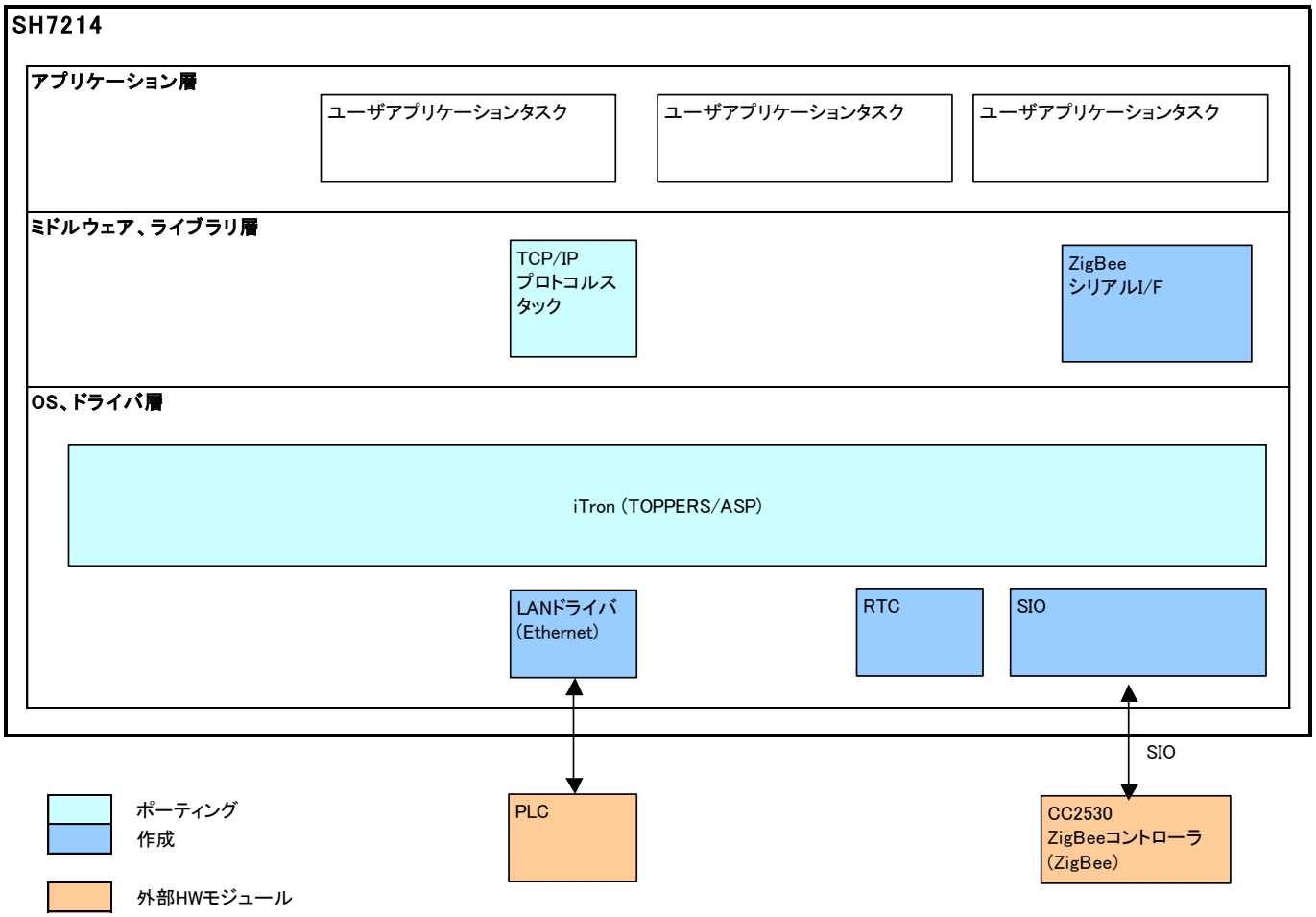


図 a-1-22　ZigBee ステーション PLC 版ソフトウエア構成図

5)ブロック図

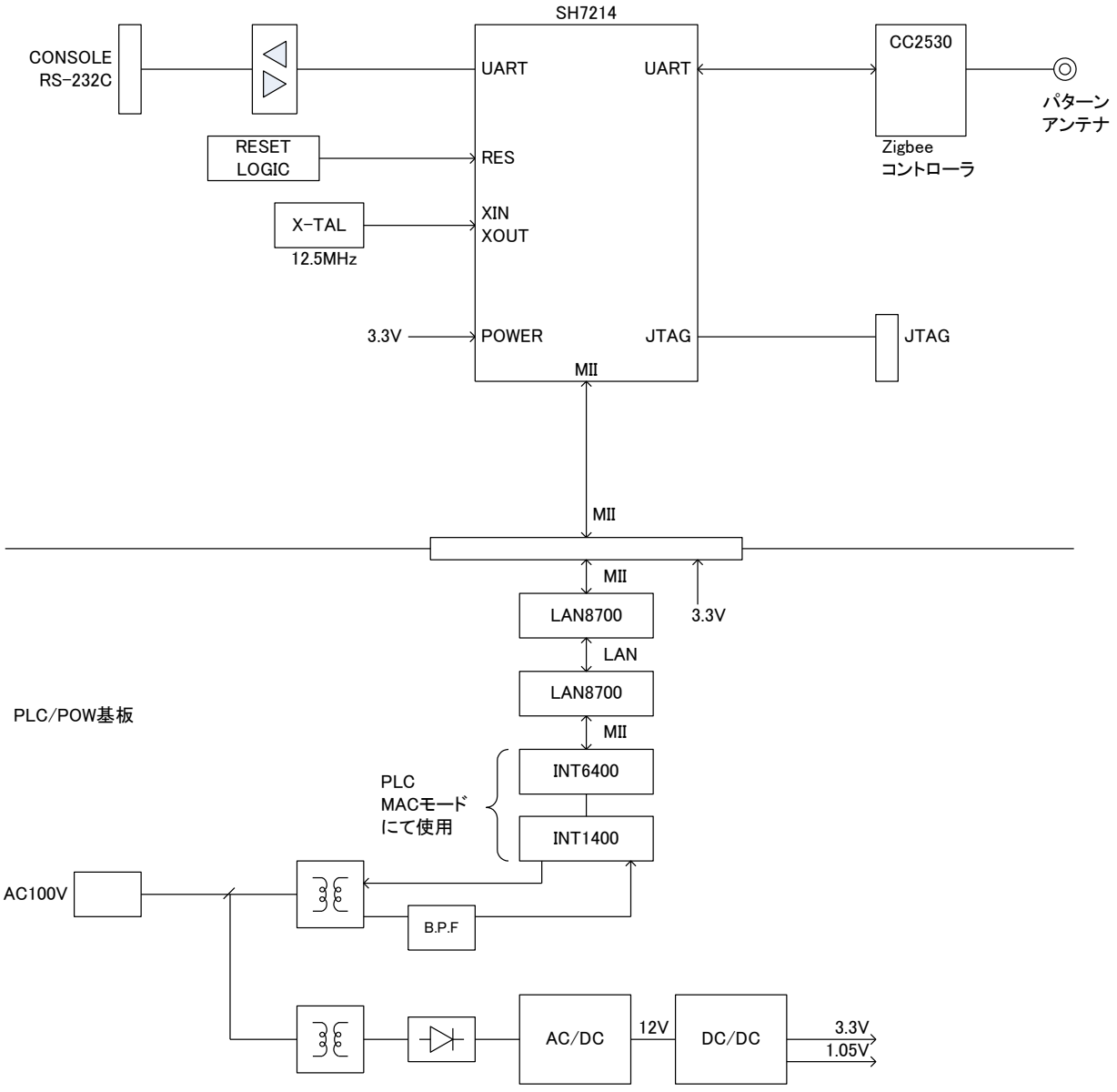


図 a-1-23　ZigBee ステーション PLC 版ソフトウエアブロック図

6)外形寸法図

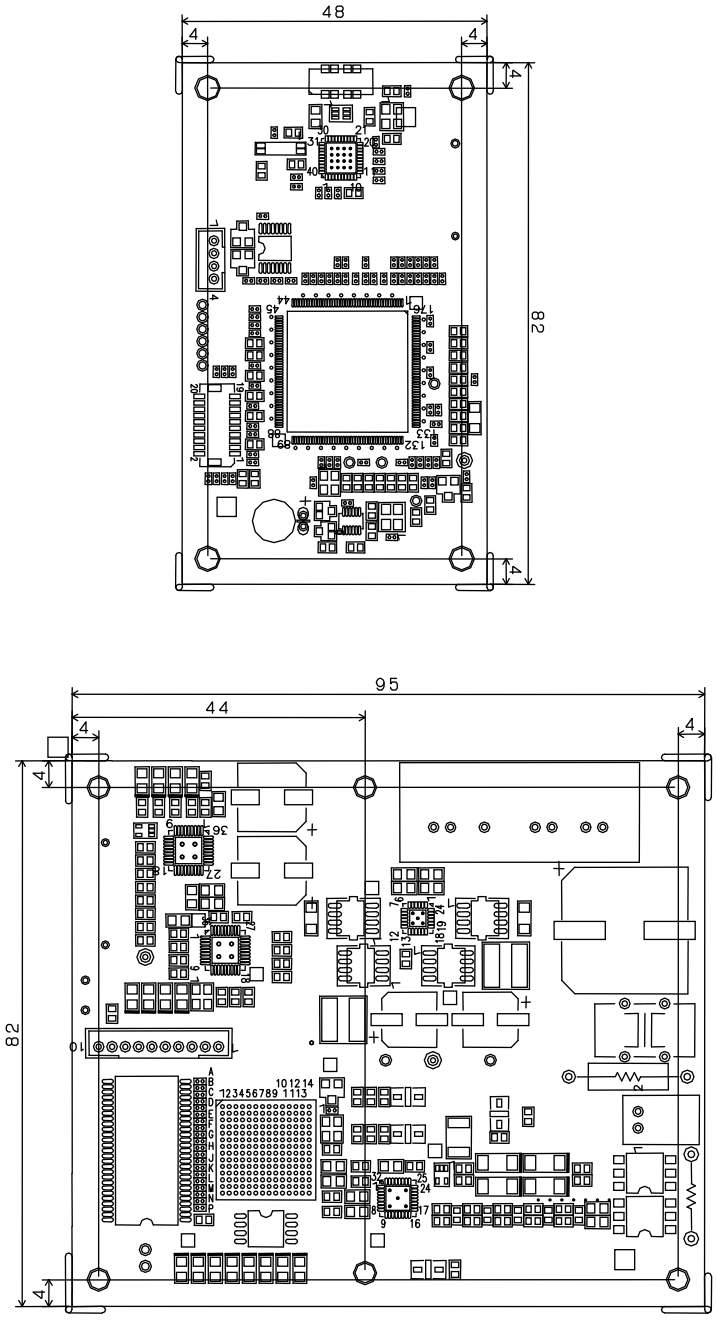


図 a-1-24　ZigBee ステーション PLC 版外形寸法図

外観図
（CPU 基板と PLC 基板を組み合わせたところ）

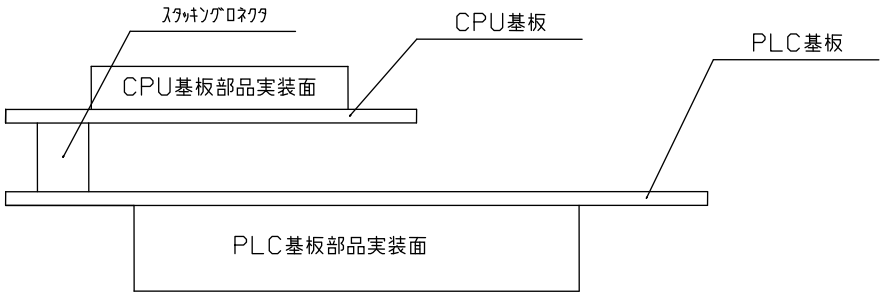


図 a-1-25　ZigBee ステーション PLC 版外観図

a-1-5 開発の経緯

＊平成21年度

従来RTミドルウエアの動作として必要であったPCに変わるものとして、各RT要素部品が容易にRTミドルウエア上に参加できることを可能とする分散型RT要素部品管理モジュールおよび、基盤通信モジュールのプロトタイプを設計・試作する。

・平成 22 年1月から 2 月　：Ethernet 版 1 次試作品を関係機関に供給した。
・平成 22 年 5 月から 6 月　：Ethernet 版 2 次試作品を関係機関に供給した。

住宅内で利用する可能性の高い通信プロトコルを選択し、RT要素部品管理モジュールとRT要素部品間の多様な通信プロトコルを許容するためのブリッジ機能を、RT要素部品管理モジュールに組込み、ホームネットワークシステムとしての拡張性を広げる。

・平成 22 年 5 月から　：SHCPU の PLC 版の試作を製作した。
・平成 22 年 7 月　：消費電力が想定より大きいことが分かり、搭載 CPU を ARM に変更した。

＊平成22年度

住宅用実証 RT システムに利用する基盤通信モジュール及び RT 要素部品管理モジュールを提供し、実証システムにおいて各モジュールの総合評価を行う。

・平成 22 年 9 月　：PLC 版の製作を行った。
・平成 22 年 12 月　：PLC 版の評価完了し、関係機関に配布を完了した。
・平成 23 年 1 月　：追加製作の基盤通信モジュール、Zigbee ステーションを関係機関に配布を完了した。
・平成 23 年 2,3 月　：電源部分改版をした要素部品管理モジュールを関係機関に配布を完了した。

(a-2) RTC-Lite フレームワークに基づく基盤通信モジュール RT ミドルウェアの研究開発
（委託先：株式会社セック）
a-2-1 概要

RTC-Lite フレームワークに基づく基盤通信モジュール RT ミドルウェアの研究開発では、これまで独立行政法人産業技術総合研究所で開発してきた RTC-Lite のフレームワークをベースとして、OMG の RTC 仕様に準拠し OpenRTM-aist と相互運用可能な基盤通信モジュール向けの軽量な RT ミドルウェアを開発することを目的としている。

基盤通信モジュール向けの RT ミドルウェアとして、通信プロトコルに CAN を採用した「高速制御用 RTC-Lite（miniRTCs）」、および、無線通信プロトコルの ZigBee を採用した「低速制御用 RTC-Lite（microRTCs）」の 2 つの RT ミドルウェアを開発した。miniRTCs は、THK 株式会社が開発した小型通信ドライバモジュールおよび、株式会社アルゴシステムが開発した基盤通信モジュール（PLC-CAN 版）上に実装し、microRTCs は、株式会社アルゴシステムが開発した基盤通信モジュール（PLC-Zigbee 版）上に実装した。
本件での目標、ならびに達成度は表 a-2-1、および表 a-2-2 の通りである。

表 a-2-1 研究項目に対する目標ならびに達成度

目標	研究開発成果	達成度
RTC-Lite フレームワークに基づく基盤通信モジュール RT ミドルウェアの研究開発		
（1）基盤通信モジュールで動作する RT ミドルウェアとして、OMG RTC 仕様に準拠した RTC-Lite フレームワークを実現する。この RTC-Lite フレームワークは、H8、PIC などの省資源なマイコンで動作可能なものとする。物理的な通信プロトコルとして、微弱無線、ZigBee、CAN を主要候補として、2 種以上の通信プロトコルをサポートする。	（1）RTC-Lite フレームワークに基づく、CAN 通信対応の miniRTCs および、Zigbee 通信に対応した microRTCs の 2 つの組込み用軽量 RT ミドルウェアを開発した。	（1）達成
（2）基盤通信モジュールの実証評価版は、平成 21 年度中に RT 要素部品開発機関ならびに、RT システム開発機関に無償で提供し、RT 要素部品の開発や RT システムの開発に供する。この際、RT ミドルウェアだけではなく、仕様書および取扱説明書もあわせて提供する。	（2）本プロジェクトのコンソーシアムメンバの他、コンソーシアム外メンバ（滝田技研株式会社、株式会社オカテック、株式会社リバスト、旭光電機株式会社）に miniRTCs および、microRTCs のソースコードならびに、インタフェース仕様書、利用マニュアルを提供した。	（2）達成
（3）実証評価の結果をフィードバックし、基盤通信モジュールの最終版をプロジェクト終了時までにリリースする。	（3）実証評価で発生した課題・問題を解決した最終版をコンソーシアムメンバにリリースした。 最終的な RT ミドルウェアのインタフェース仕様は成果報告書として一般に公開する。	（3）達成

a-2-2 住宅向け RT システムの構成

本システムでは、住宅に設置される窓やドア、家電、各種センサなどを、それぞれ一つの RT 要素部品とし、それらを RT ミドルウェア技術により連係させることで、住宅向けの環境分散型 RT システムを実現している。本システムでは、ホームコントローラによる集中管理ではなく、モジュールごとの分散管理のアーキテクチャを採用している。それぞれの RT 要素部品は、基盤通信モジュール、要素部品管理モジュール、ホームコントローラにより、階層的に管理がおこなわれる。各モジュールは独立して動作が可能であり、システムの一部に異常があった場合でも、他のモジュールは安全に動作することが可能である。

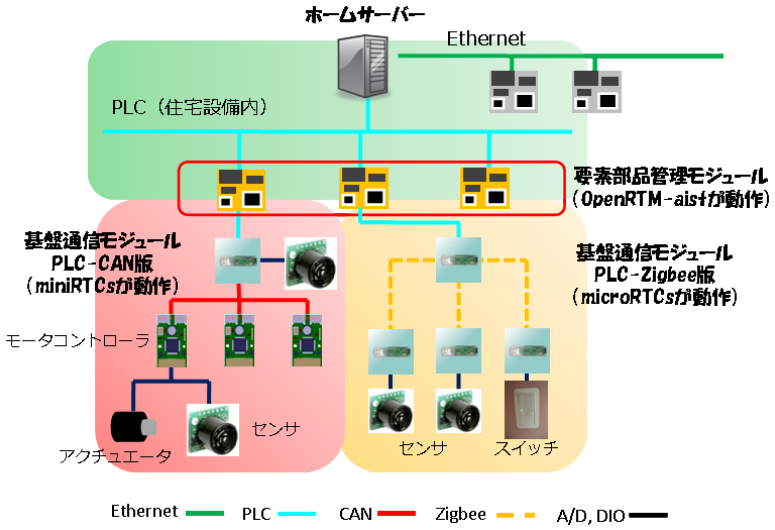


図 a-2-1 住宅向け RT システムの構成

窓 RT 要素部品は、リニアモータ、人感センサ、接触センサを備え、窓の開閉の際のパワーアシスト機能、セキュリティのための自動施錠機能などを実現する。窓 RT 要素部品は、RT ミドルウェアが実装された基盤通信モジュール、エンドデバイスで管理、制御される。窓 RT 要素部品では、窓開閉などのリアルタイム制御、通信の信頼性が要求されるため、各モジュール間は車載ネットワークで利用されている CAN（Control Area Network）で接続されている。

家電 RT 要素部品やセンサ RT 要素部品についても、窓 RT 要素部品と同様に、基盤通信モジュールとエンドデバイスにより構成される。ただし、これらの RT 要素部品では、数ミリ秒オーダのリアルタイム制御は不要なこと、通信が一時途絶えても人体への危険性がないことより、配置の自由度を考慮し、低消費電力で低コストの無線ネットワークである ZigBee を採用している。

ホームコントローラは、パソコン相当の性能を持ち、GUI を備えるパネルコンピュータを想定し、ユーザからの操作受け付け、ユーザへの情報提示をおこなう。また、ホームコントローラは、インターネット経由でデータセンターと接続され、データセンターとの間で、システム運用のための情報授受をおこなう。

要素部品管理モジュールは、複数の RT 要素部品を管理するためのモジュールである。要素部品管理モジュールは、部屋やブレーカーなどの任意の区画単位に設置され、その中に設置されている RT 要素部品の管理をおこなう。

本システムでは、ホームコントローラと要素部品管理モジュール、基盤通信モジュール間のネットワークに Ethernet ではなく、PLC（Power Line Communication）を採用した。住宅内の様々な RT 要素部品をネットワークに接続することを考える場合、施工業者による機器の設置や配線の引き回しが容易である必要がある。PLC は、既存の電力線を利用でき、設置にかかるコストを抑えることができる。近年ではスマートメーター実現のための技術として注目されており、住宅への普及も期待できる。

a-2-3 高速制御用 RTC-Lite(miniRTCs)仕様

a-2-3-1 miniRTCs 構成

miniRTCs は RTC-Lite の構成を基本としており、図 a-2-2 に示すように構成される。このうち、省資源マイコン上に実装されるのはデバイスコンポーネントのみである。

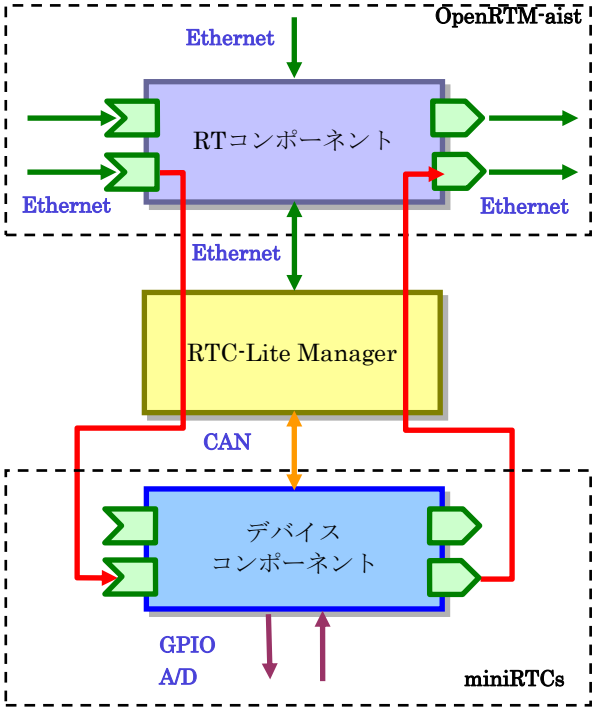


図 a-2-2 miniRTCs 構成

RT コンポーネント

OpenRTM-aist などの RT ミドルウェア環境で動作するコンポーネントであり、RT ミドルウェアとしての全機能を持つ。

RTC-Lite Manage

複数のデバイスコンポーネントと接続し、Ethernet と CAN/ZigBee の通信を変換する。また、各ポートに対し、入出力データを振り分ける。

デバイスコンポーネント

miniRTCs 環境で動作するコンポーネントであり、RT ミドルウェアとしての機能は制限される。デバイスと 1 対 1 で対応し、センサやモータの制御を行う。

a-2-3-2 機能

miniRTCs の機能一覧を表 a-2-2 に示す。

表 a-2-2 機能一覧

機能	機能対応	備考
状態遷移	対応する	図 a-2-3 参照
データポート	15 個まで （IN ポート、OUT ポート合わせて）	ポート ID は 1～15
RTC 間の通信	直接通信可能	
コンフィギュレーション	対応しない	
サービスポート	持たない	データポートで代用する
プラグアンドプレイ	対応する	
生存情報の定期送信	対応する	1 秒周期で送信する
データポートで使用可能な型	byte 配列のみ	最大 8byte
ノード数	15 個まで	ノード ID は 1～15

miniRTCs の状態遷移図を図 a-2-3 に、状態遷移表を表 a-2-3 にそれぞれ示す。

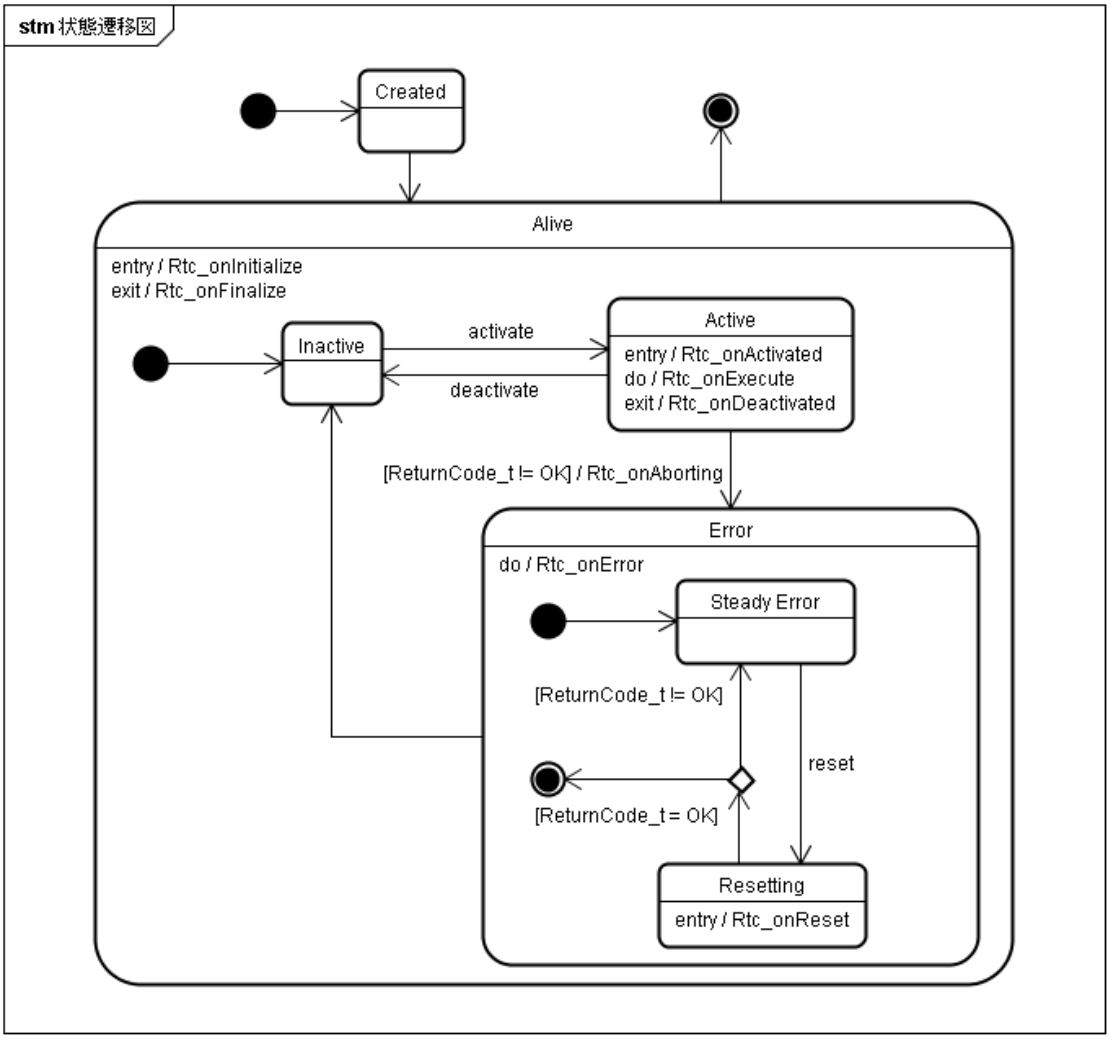


図 a-2-3 状態遷移図

表 a-2-3 状態遷移表

イベント / 内部状態	activate	deactivate	reset	exit	タイマ割込み
Inactive	Active / onActivated	PRECONDITION_NOT_MET※1	PRECONDITION_NOT_MET	終了※2 / onFinalize	Inactive / 何もしない
Active	PRECONDITION_NOT_MET	Inactive / onDeactivated	PRECONDITION_NOT_MET	終了※3 / onDeactivated, onFinalize	Active / onExecute
Error	PRECONDITION_NOT_MET	PRECONDITION_NOT_MET	Inactive / onReset	終了※4 / onFinalize	Error / onError

※1 「操作のための前提条件が満たされていない」としてエラーを返す。(表 a-2-11 参照)
※2 onFinalize を実行後、終了する。
※3 onDeactivated を実行後、Inactive 状態へ遷移し、onFinalize を実行後、終了する。
※4 onReset は実行せず Inactive 状態へ遷移し、onFinalize を実行後、終了する。

a-2-3-3 インタフェース仕様

a-2-3-3-1 CAN 通信プロトコル

miniRTCs では、独自の CAN 通信プロトコルを利用し、統合ミドルウェアとデバイス RTC 間、及びデバイス RTC 同士のコマンド、データ、及び生存情報の送受信を行っている。

a-2-3-3-2 CAN 通信について

CAN では、図 a-2-4 に示すメッセージプロトコルを用いて通信を行っている。このうち、ユーザが設定可能な領域は、ID フィールドとデータフィールドである。

CAN では、ID フィールドを用いて、メッセージの優先度やフィルタリングなどの通信調停を行う。

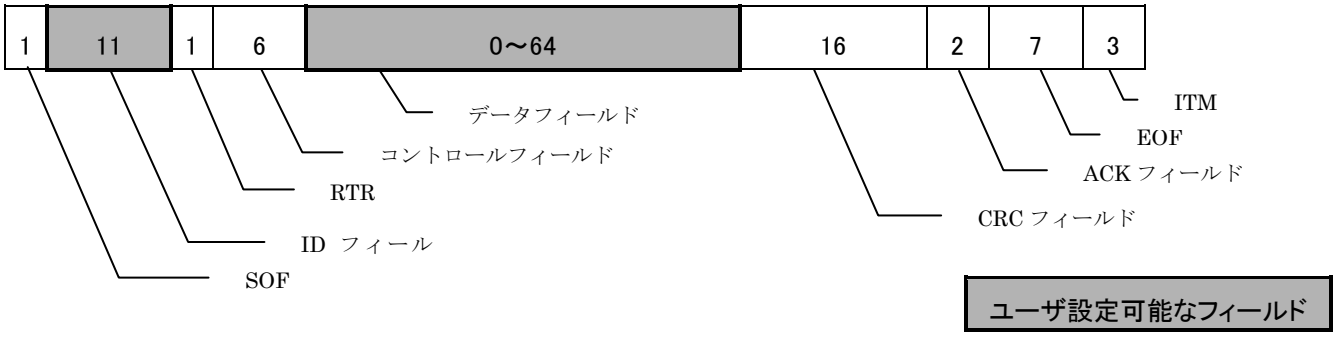


図 a-2-4　CAN のメッセージプロトコル

ID フィールドの利用形態を図 a-2-5 に示す。miniRTCs では、メッセージの経路及びメッセージの種別により、ID フィールドの利用形態が異なる。

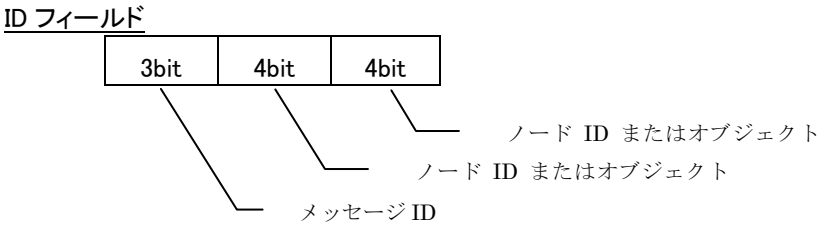


図 a-2-5　ID フィールドの利用形態

ただし、上位の 3 ビットは、全メッセージにおいて共通であり、表 a-2-4 に示すメッセージ ID を指定する。これにより、メッセージの種別に応じた優先度での通信を行うことができる。

下位 8 ビットは、メッセージの送信元、もしくはメッセージの送信先を表すノード ID やオブジェクト ID を指定する。ノード ID には、各 CPU に割り当てられた、0000B～1111B の 16 種類の ID を指定する。オブジェクト ID には、デバイス CPU 上のコンポーネントや入力ポート、出力ポートを表す 0000B～1111B の 16 種類の ID を指定する。

表 a-2-4　メッセージ ID 一覧

ID	名称	内容	備考
000	特権通信	miniRTCs 以外のメッセージ	今後の拡張
001	コマンド送信	コンポーネントに対して実行させるコマンドのメッセージ	
010	コマンド返信	コンポーネントが実行したコマンドの応答メッセージ	
011	ショートデータ送信	8byte 以下の送信データメッセージ	
100	ロングデータ送信	分割された送信データメッセージ	今後の拡張
101	ハートビート	デバイス RTC から周期的に送信される生存情報	
110	—	リザーブ	
111	—	リザーブ	

a-2-3-3-3 メッセージ種別

表 a-2-4 に示したメッセージ毎の ID フィールドとデータフィールドの利用方法を以下に述べる。

(1) 特権通信メッセージ(今後の拡張)

特権通信メッセージの ID フィールド、及びデータフィールドの利用方法を図 a-2-6 に示す。特権通信メッセージは miniRTCs 以外によって処理されるメッセージであり、フィールドの内容を自由に設定することが可能である。

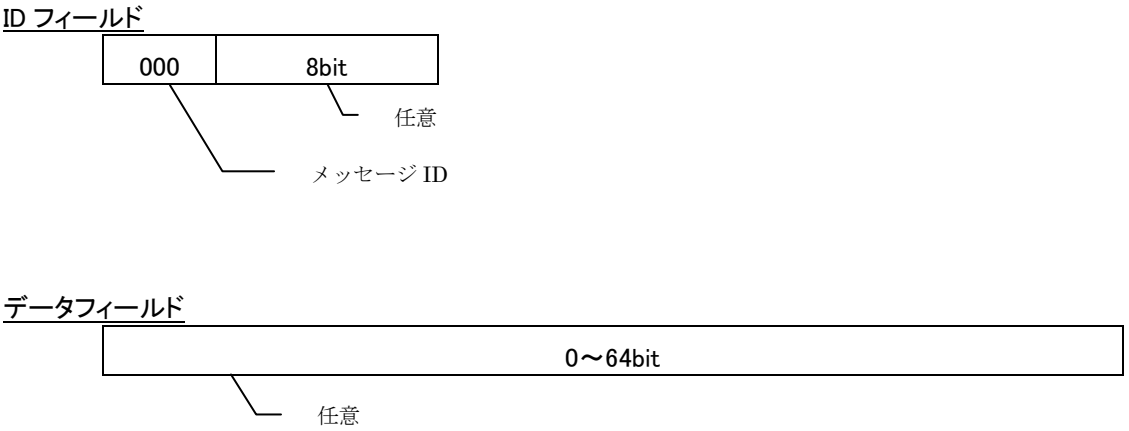


図 a-2-6 特権通信メッセージのデータフォーマット

(2) コマンド送信メッセージ

コマンド送信メッセージの ID フィールド、及びデータフィールドの利用方法を図 a-2-7 に示す。

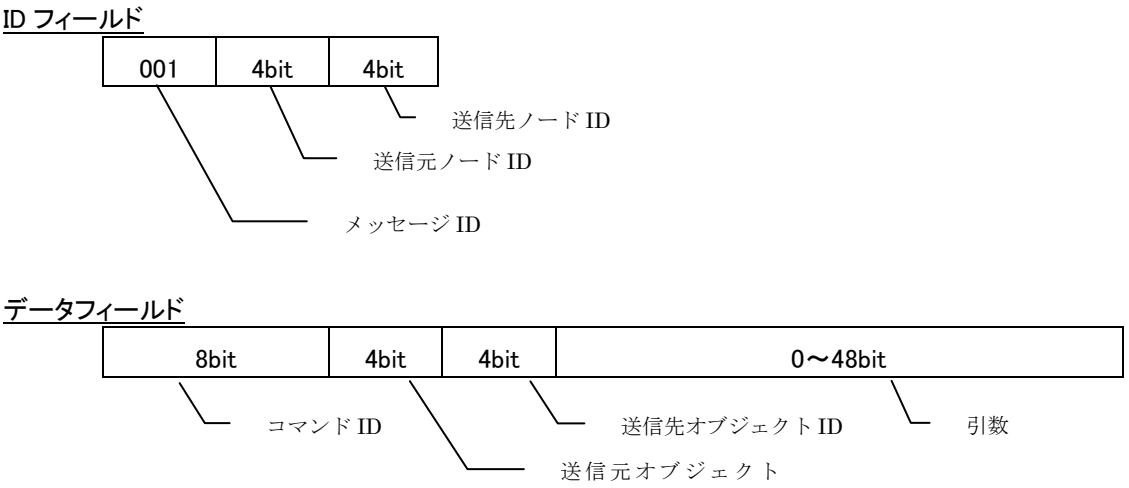


図 a-2-7 コマンド送信メッセージのデータフォーマット

データフィールドで設定するコマンド ID の一覧を、表 a-2-5 に示す。

表 a-2-5 コマンド ID 一覧

ID	コマンド名	内容	備考
0x00	ACTIVATE	RT コンポーネントの活性化	
0x01	DEACTIVATE	RT コンポーネントの非活性化	
0x02	RESET	RT コンポーネントのリセット	
0x03	EXIT	RT コンポーネントの終了	
0x10	SET_RATE	実行周期の設定	今後の拡張
0x11	GET_RATE	実行周期の取得	今後の拡張
0x12	GET_KIND	実行種別の取得	今後の拡張
0x20	CONNECT	入出力ポート間の接続	
0x21	DISCONNECT	入出力ポート間の切断	
0x30	GET_ERROR_LOG	エラーログの取得	今後の拡張

コマンド毎のデータフィールド設定項目を表 a-2-6 に示す。

表 a-2-6 コマンド毎のデータフィールド設定項目

コマンド名	送信元オブジェクト ID	送信先オブジェクト ID	引数	備考
ACTIVATE	ゼロ固定	ゼロ固定	なし	
DEACTIVATE	ゼロ固定	ゼロ固定	なし	
RESET	ゼロ固定	ゼロ固定	なし	
EXIT	ゼロ固定	ゼロ固定	なし	
SET_RATE	—	—	—	今後の拡張
GET_RATE	—	—	—	今後の拡張
GET_KIND	—	—	—	今後の拡張
CONNECT	ゼロ固定	送信先オブジェクト ID	接続先ノード ID：4bit 接続先オブジェクト ID：4bit	
DISCONNECT	ゼロ固定	送信先オブジェクト ID	なし	
GET_ERROR_LOG	—	—	—	今後の拡張

(3) コマンド返信メッセージ

コマンド返信メッセージの ID フィールド、及びデータフィールドの利用方法を図 a-2-8 に示す。

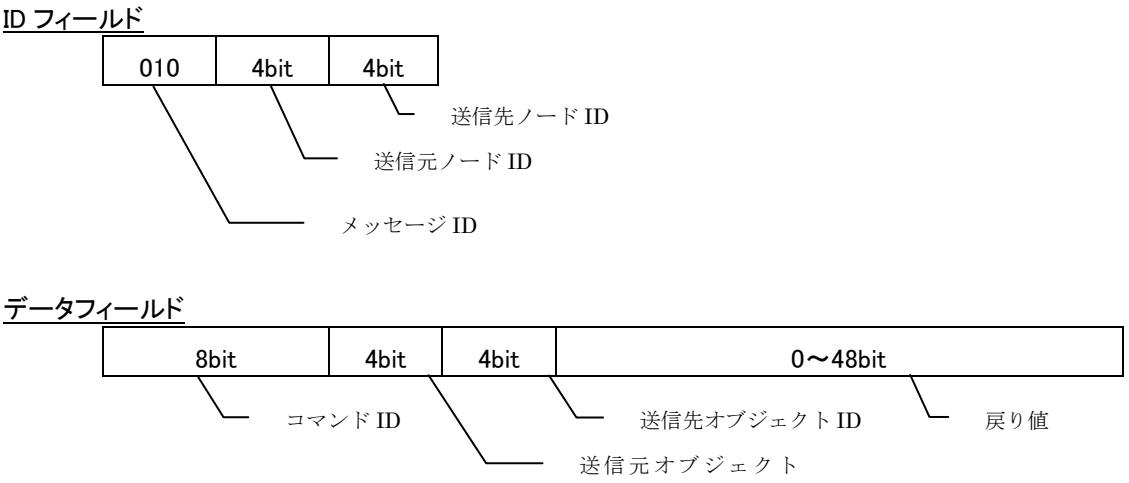


図 a-2-8 コマンド返信セージのデータフォーマット

コマンド毎の戻り値を表 a-2-7 に示す。

表 a-2-7 コマンド毎の戻り値

コマンド名	引数	備考
ACTIVATE	エラーコード：1byte	
DEACTIVATE	エラーコード：1byte	
RESET	エラーコード：1byte	
EXIT	エラーコード：1byte	
SET_RATE	—	今後の拡張
GET_RATE	—	今後の拡張
GET_KIND	—	今後の拡張
CONNECT	エラーコード：1byte	
DISCONNECT	エラーコード：1byte	
GET_ERROR_LOG	—	今後の拡張

（4 ）ショートデータ送信メッセージ

ショートデータ送信メッセージの ID フィールド、及びデータフィールドの利用方法を図 a-2-9 に示す。

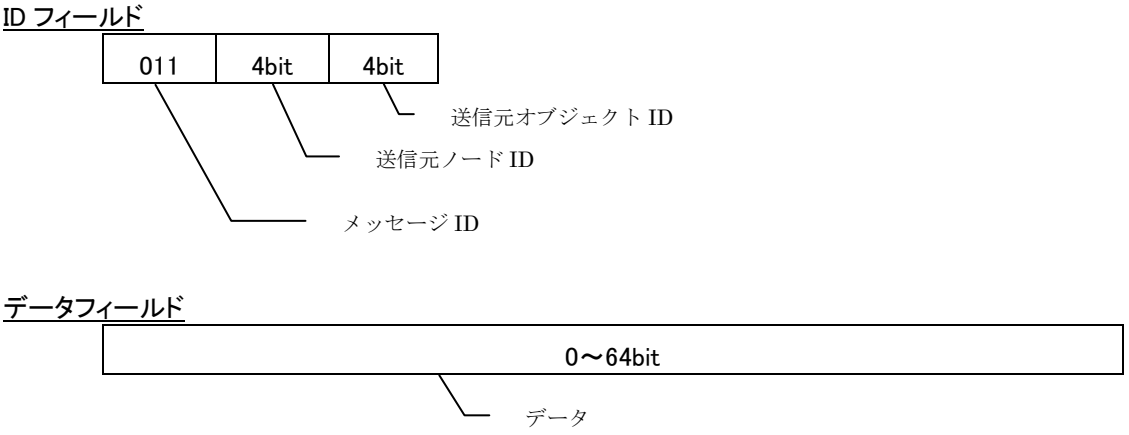


図 a-2-9 ショートデータ送信メッセージのデータフォーマット

(5) ロングデータ送信メッセージ（今後の拡張）

ロングデータ送信メッセージの ID フィールド、及びデータフィールドの利用方法を図 a-2-10 に示す。

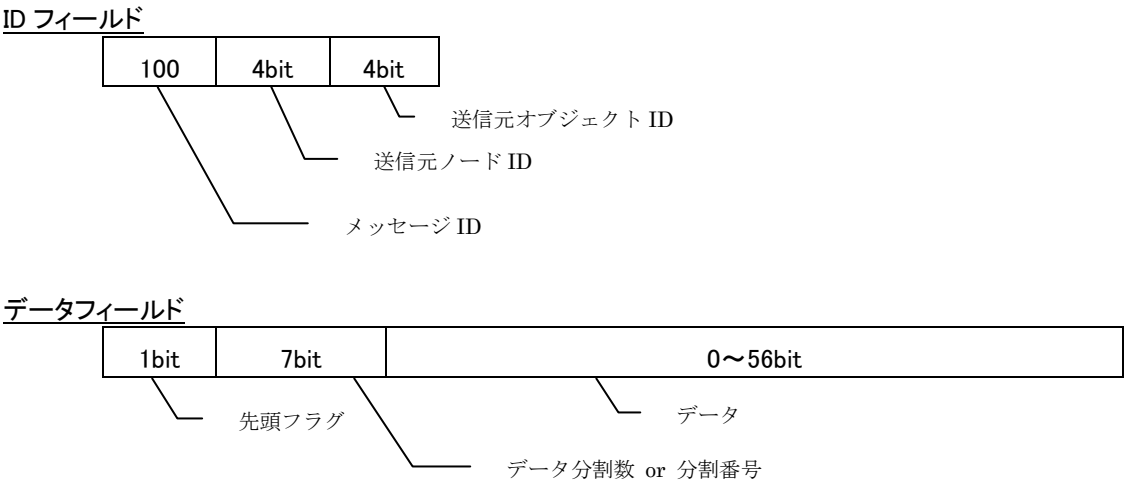


図 a-2-10 ロングデータ送信メッセージのデータフォーマット

CAN 通信では、一度に送信するこのできるデータの最大サイズは 8byte である。そのため、大きなサイズのデータを送信するには、データを分割する必要がある。そこで、送受信データサイズが 8byte より大きい場合は、図 a-2-11 に示すように、データフィールドの先頭 1 バイトを利用し、データの分割送信を実現する。これにより、最大 896byte（7byte × 128）のデータの送受信が可能となる。

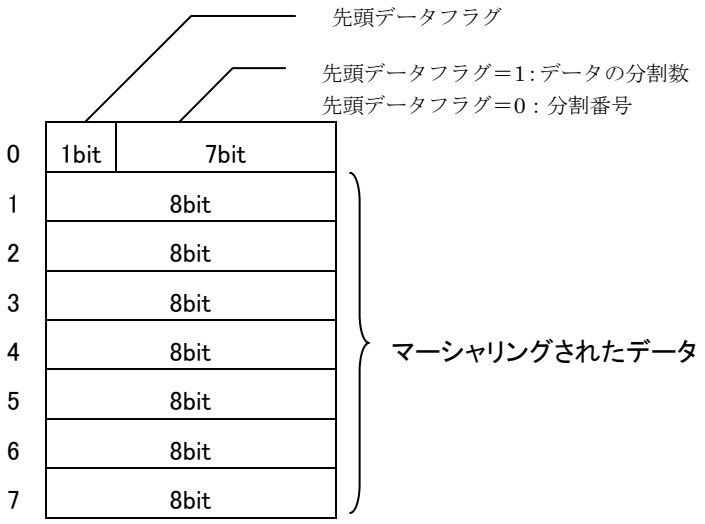
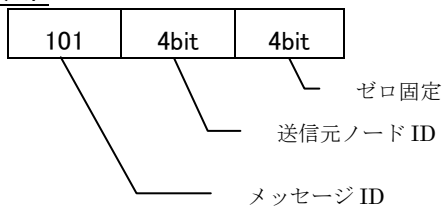


図 a-2-11 データ分割フォーマット

(6) ハートビートメッセージ

ハートビートメッセージの ID フィールド、及びデータフィールドの利用方法を図 a-2-12 に示す。

ID フィールド



データフィールド

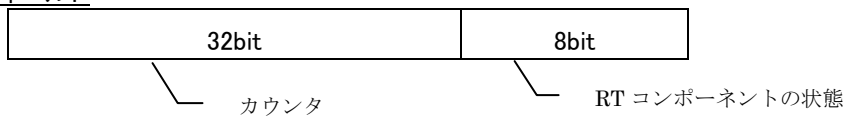


図 a-2-12 ハートビートメッセージのデータフォーマット

データフィールドのカウンタは、ハートビートを送信する毎にカウントアップする。0 からカウントアップし、最大値（0xFFFFFFFF）を超えるとロールオーバーする。

RT コンポーネントの状態は、ステートマシンにおける RT コンポーネントの状態を表す。ID と RT コンポーネントの状態との対応を表 a-2-8 に示す。

表 a-2-8 RT コンポーネントの状態一覧

ID	RT コンポーネントの状態
0x00	未定義状態
0x01	Created 状態
0x02	Inactive 状態
0x03	Active 状態
0x04	Error 状態

a-2-3-4 デバイス RTC インタフェース関数

RT コンポーネントのアクションのインタフェース関数について、miniRTCs での対応関係を表 a-2-9 に示す。各関数は、表 a-2-3 に示したタイミングで呼ばれる。

表 a-2-9 アクションインタフェース関数一覧

インタフェース関数	対応	備考
Rtc_onInitialize	○	
Rtc_onActivated	○	
Rtc_onExecute	○	
Rtc_onDeactivated	○	
Rtc_onAborting	○	
Rtc_onReset	○	
Rtc_onError	○	
Rtc_onFinalize	○	
Rtc_onStateUpdate	×	
Rtc_onStartup	×	
Rtc_onShutdown	×	

a-2-3-5 データポートインタフェース

データポートのインタフェース関数を表 a-2-10 に示す。

表 a-2-10 データポートインタフェース関数一覧

インタフェース関数	処理概要	備考
DataPort_getInPortID	IN ポート ID を取得する	
DataPort_getOutPortID	OUT ポート ID を取得する	
DataPort_inPortRead	IN ポートからデータを読み込む	
DataPort_outPortWrite	OUT ポートへデータを書き込む	

表 a-2-10 に示したデータポートのインタフェース関数の詳細を以下に述べる。

(1) DataPort_getInPortID

DataPort_getInPortID

[概要]

未使用の IN ポート ID を取得する。
関数を呼ぶごとに、0x01 からポート ID が返される。

[C 言語記述形式]

UByte DataPort_getInPortID(void)

[パラメタ]

なし。

[復帰値]

シンボル / 値	説明
0x01～0x0F	ポートD の取得に成功
0xFF	ポート ID の空きがない

[補足説明]

取得可能なポート ID は、0x01～0x0F。

(2) DataPort_getOutPortID

DataPort_getOutPortID

[概要]

未使用の OUT ポート ID を取得する。
関数を呼ぶごとに、0x01 からポート ID が返される。

[C 言語記述形式]

UByte DataPort_getOutPortID(void)

[パラメタ]

なし。

[復帰値]

シンボル / 値	説明
0x01～0x0F	ポート ID の取得に成功
0xFF	ポート ID の空きがない

[補足説明]

取得可能なポート ID は、0x01～0x0F。

(3) DataPort_inPortRead

DataPort_inPortRead

[概要]

指定した IN ポート ID のデータを取得する。

[C 言語記述形式]

RCode DataPort_inPortRead(UByte portID, UByte *dataBuffer, UInt32 *dataSize)

[パラメタ]

I/O	パラメタ	説明
I	portID	ポート ID（0x01～0x0F）
O	dataBuffer	取得データ格納バッファ
O	dataSize	取得データサイズ （受信データがなければゼロが設定される）

[復帰値]

シンボル / 値	説明
E_OK	正常復帰
E_ID	0x01～0x0F 以外のポート ID がパラメタで指定された
E_PORT	未使用のポート ID もしくは OUT ポート ID がパラメタで指定された

[補足説明]

取得可能なデータの最大サイズは 8byte。

(4) DataPort_outPortWrite

DataPort_outPortWrite

[概要]

指定した OUT ポート ID へデータを設定する。

[C 言語記述形式]

RCode DataPort_outPortWrite(UByte portID, const UByte *dataBuffer, UInt32 dataSize)

[パラメタ]

I/O	パラメタ	説明
I	portID	ポート ID（0x01～0x0F）
I	dataBuffer	設定データ格納バッファ
I	dataSize	設定データサイズ（0～8） （ゼロを指定すると何も設定されない）

[復帰値]

シンボル / 値	説明
E_OK	正常復帰
E_ID	0x01～0x0F 以外のポート ID がパラメタで指定された
E_PORT	未使用のポート ID もしくは IN ポート ID がパラメタで指定された
E_SIZE	9byte 以上のサイズがパラメタで設定された

[補足説明]

設定可能なデータの最大サイズは 8byte。

a-2-3-6 特記事項

データポートのデータ送信時において、以下の注意点がある。

- 複数ポートへの出力が重なった場合、ポート ID の小さいポートからデータを送信する。
- 送信に失敗した場合であっても送信要求をクリアする。

a-2-3-7 エラーコード一覧

表 a-2-11 及び表 a-2-12 にエラーコード一覧を示す。

表 a-2-11 ReturnCode_t 型エラーコード一覧

ID	エラーコード	内容
RTC_OK	0x00	操作が成功した
RTC_ERROR	0x01	一般的なエラー
RTC_BAD_PARAMETER	0x02	無効な引数を渡した
RTC_UNSUPPORTED	0x03	サポートされていない操作を行った
RTC_OUT_OF_RESOURCES	0x04	操作を行うために必要なリソースが不足している
RTC_PRECONDITION_NOT_MET	0x05	操作のための前提条件が満たされていない

表 a-2-12 Rcode 型エラーコード一覧

ID	エラーコード	内容
E_OK	0x00	正常終了
E_ERROR	0x11	異常終了
E_BUSY	0x21	ビジー状態
E_ID	0x22	ID 異常
E_NOINIT	0x23	未初期化状態
E_SIZE	0x24	データサイズ異常
E_STATE	0x25	状態異常
E_OVER	0x26	領域を超える
E_SETTING	0x27	設定状態異常
E_EVENT	0x28	イベント異常
E_PAR	0x29	パラメタ異常
E_PORT	0x30	ポート異常
E_NODATA	0x31	データがない

a-2-4 低速制御用 RTC-Lite(microRTCs)仕様

a-2-4-1 microRTCs 構成

microRTCs は RTC-Lite の構成を基本としており、図 a-2-13 に示すように構成される。このうち、省資源マイコン上に実装されるのはデバイスコンポーネントのみである。

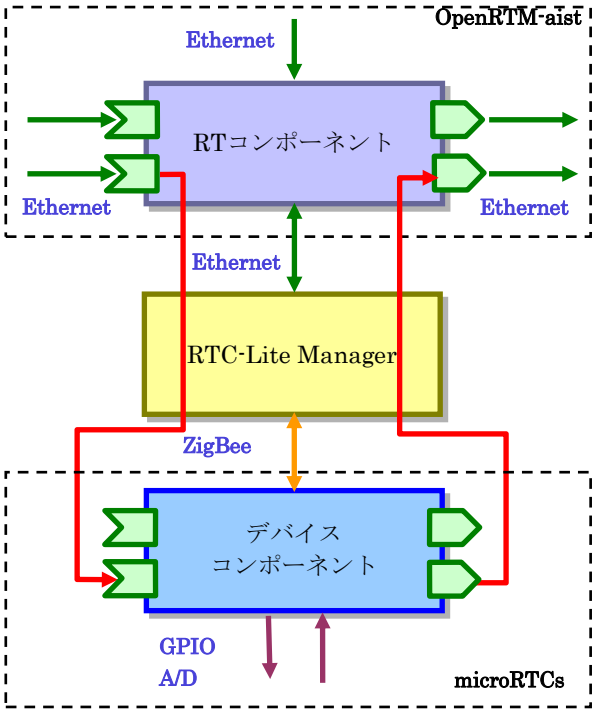


図 a-2-13 microRTCs 構成

RT コンポーネント

OpenRTM-aist などの RT ミドルウェア環境で動作するコンポーネントであり、RT ミドルウェアとしての全機能を持つ。

RTC-Lite Manage

複数のデバイスコンポーネントと接続し、Ethernet と CAN/ZigBee の通信を変換する。また、各ポートに対し、入出力データを振り分ける。

デバイスコンポーネント

microRTCs 環境で動作するコンポーネントであり、RT ミドルウェアとしての機能は制限される。デバイスと 1 対 1 で対応し、センサやモータの制御を行う。

a-2-3-2 機能

microRTCs の機能一覧を表 a-2-13 に示す。

表 a-2-13 機能一覧

機能	機能対応	備考
状態遷移	対応する	図 a-2-14 参照
データポート	15 個まで （IN ポート、OUT ポート合わせて）	ポート ID は 1～15
RTC 間の通信	直接通信はできない	
コンフィギュレーション	対応しない	
サービスポート	持たない	データポートで代用する
プラグアンドプレイ	対応する	
生存情報の定期送信	対応する	1 秒周期で送信する
データポートで使用可能な型	byte 配列のみ	最大 81byte
ノード数	255 個まで	ノード ID は 1～255

microRTCs の状態遷移図を図 a-2-14 に、状態遷移表を表 a-2-14 にそれぞれ示す。

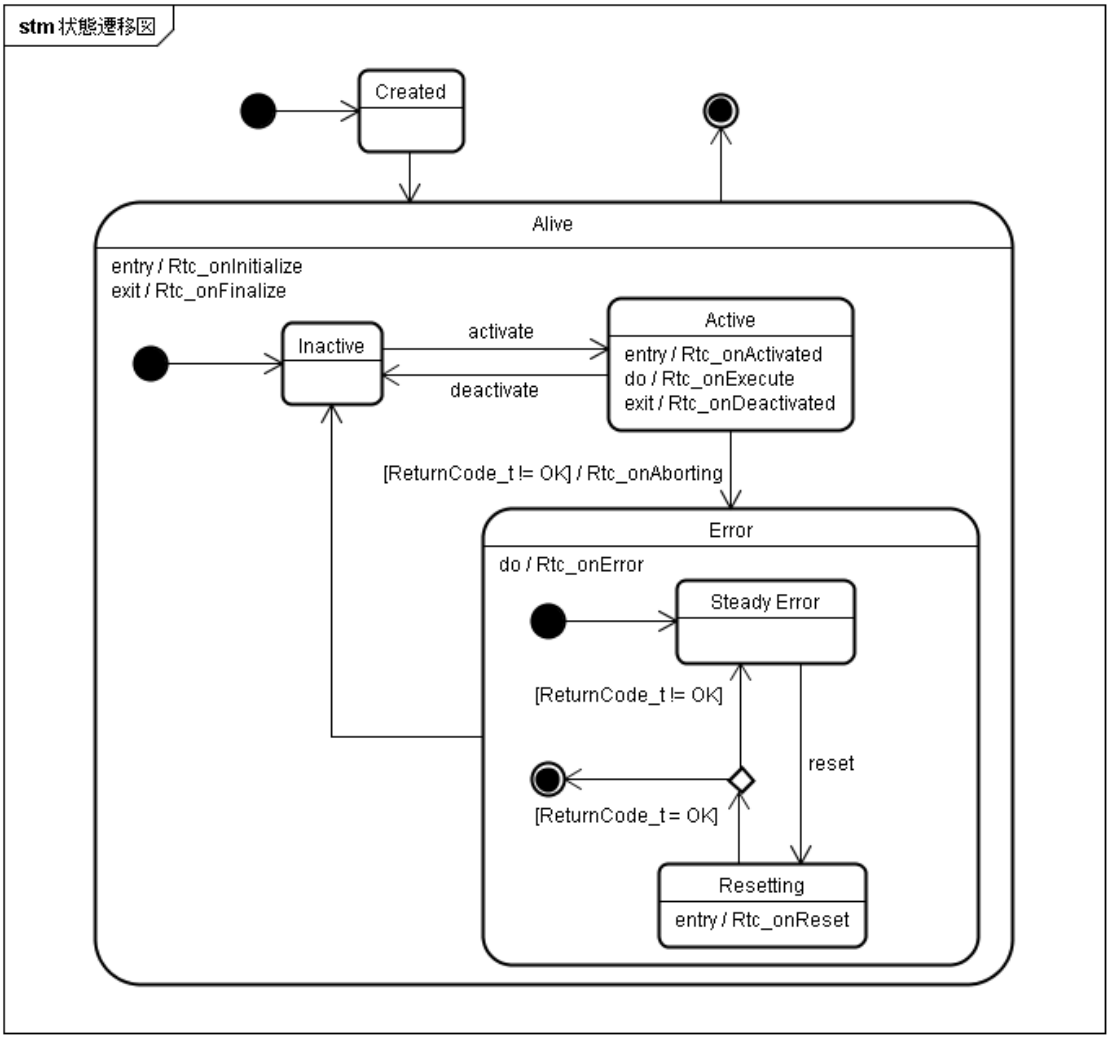


図 a-2-14 状態遷移図

表 a-2-14 状態遷移表

イベント / 内部状態	activate	deactivate	reset	exit	タイマ割込み
Inactive	Active / onActivated	PRECONDITION_NOT_MET※1	PRECONDITION_NOT_MET	終了※2 / onFinalize	Inactive / 何もしない
Active	PRECONDITION_NOT_MET	Inactive / onDeactivated	PRECONDITION_NOT_MET	終了※3 / onDeactivated, onFinalize	Active / onExecute
Error	PRECONDITION_NOT_MET	PRECONDITION_NOT_MET	Inactive / onReset	終了※4 / onFinalize	Error / onError

※1 「操作のための前提条件が満たされていない」としてエラーを返す。(表 a-2-22 参照)
※2 onFinalize を実行後、終了する。
※3 onDeactivated を実行後、Inactive 状態へ遷移し、onFinalize を実行後、終了する。
※4 onReset は実行せず Inactive 状態へ遷移し、onFinalize を実行後、終了する。

a-2-4-3 インタフェース仕様

microRTCs のネットワークトポロジは、以下の理由により図 a-2-15 に示すスター・トポロジとする。

・エンドデバイスは常に動作する必要がなく、省電力のスリープ状態が可能である。
・住宅内システムのためマルチホップの必要がなく、遅延時間が少ない。
・シンプルな構成のため、実装コストが低い。

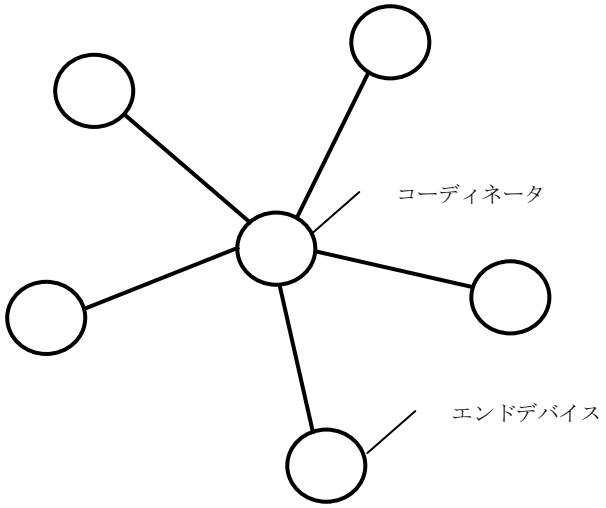


図 a-2-15　microRTCs のネットワークトポロジ

a-2-4-3-1 ZigBee 通信プロトコル

microRTCs では、独自の ZigBee 通信プロトコルを利用し、統合ミドルウェアとデバイス RTC 間、及びデバイス RTC 同士のコマンド、データ、及び生存情報の送受信を行っている。

a-2-4-3-2 ZigBee 通信について

ZigBee では、図 a-2-16 に示すメッセージプロトコルを用いて通信を行っている。網掛けの部分がユーザによって設定可能な領域である。

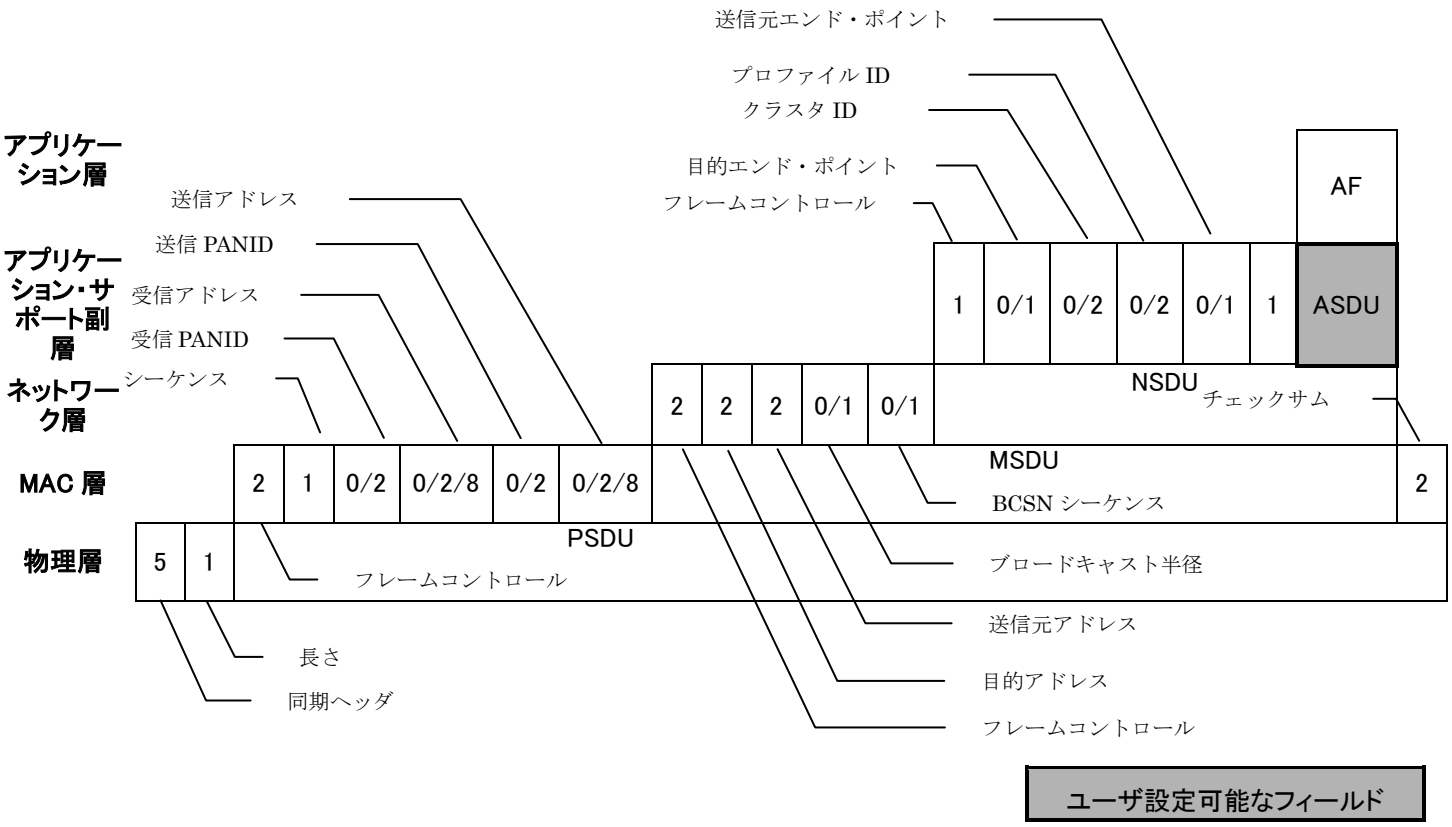


図 a-2-16　ZigBee のメッセージプロトコル

ASDU(アプリケーション・サポート副層ペイロード)の利用形態を図 a-2-17 に示す。microRTCs では、メッセージの経路及びメッセージの種別により、ASDU の利用形態が異なる。

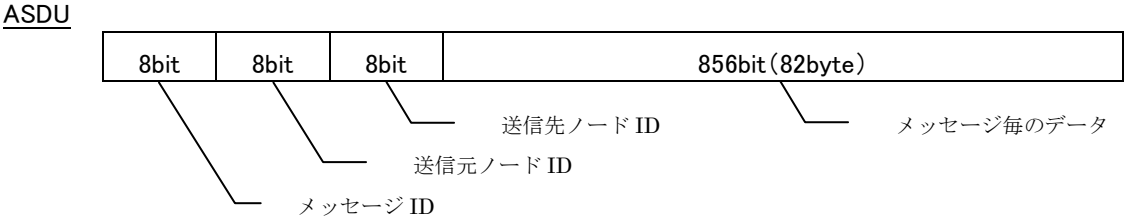


図 a-2-17　ASDU の利用形態

ASDU の先頭 3byte は各メッセージで共通のフォーマットとし、メッセージ ID、送信元ノード ID、及び送信先ノード ID を指定する。メッセージ ID には、表 a-2-15 に示す値を設定する。

表 a-2-15　メッセージ ID 一覧

ID	名称	内容	備考
000	特権通信	microRTCs 以外のメッセージ	今後の拡張
001	コマンド送信	コンポーネントに対して実行させるコマンドのメッセージ	
010	コマンド返信	コンポーネントが実行したコマンドの応答メッセージ	
011	ショートデータ送信	81byte 以下の送信データメッセージ	
100	ロングデータ送信	分割された送信データメッセージ	今後の拡張
101	ハートビート	デバイス RTC から周期的に送信される生存情報	
110	—	リザーブ	
111	—	リザーブ	

a-2-4-3-3 メッセージ種別

表 a-2-15 に示したメッセージ毎の ASDU の利用方法を以下に述べる。

(1) 特権通信メッセージ(今後の拡張)

特権通信メッセージのアドレス部、及び ASDU の利用方法を図 a-2-18 に示す。特権通信メッセージは microRTCs 以外によって処理されるメッセージであり、フィールドの内容を自由に設定することが可能である。

ASDU

0x00	672bit(84byte)

図 a-2-18 特権通信メッセージのデータフォーマット

(2) コマンド送信メッセージ

コマンド送信メッセージの ASDU の利用方法を図 a-2-19 に示す。

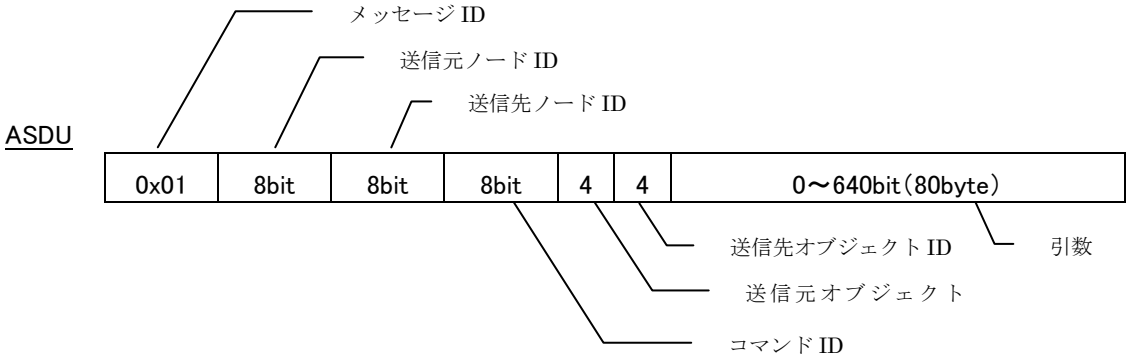


図 a-2-19 コマンド送信メッセージのデータフォーマット

ASDU で設定するコマンド ID の一覧を、表 a-2-16 に示す。

表 a-2-16 コマンド ID 一覧

ID	コマンド名	内容	備考
0x00	ACTIVATE	RT コンポーネントの活性化	
0x01	DEACTIVATE	RT コンポーネントの非活性化	
0x02	RESET	RT コンポーネントのリセット	
0x03	EXIT	RT コンポーネントの終了	
0x10	SET_RATE	実行周期の設定	今後の拡張
0x11	GET_RATE	実行周期の取得	今後の拡張
0x12	GET_KIND	実行種別の取得	今後の拡張
0x20	CONNECT	入出力ポート間の接続	
0x21	DISCONNECT	入出力ポート間の切断	
0x30	GET_ERROR_LOG	エラーログの取得	今後の拡張

コマンド毎の ASDU 設定項目を表 a-2-17 に示す。

表 a-2-17 コマンド毎のデータフィールド設定項目

コマンド名	送信元オブジェクト ID	送信先オブジェクト ID	引数	備考
ACTIVATE	ゼロ固定	ゼロ固定	なし	
DEACTIVATE	ゼロ固定	ゼロ固定	なし	
RESET	ゼロ固定	ゼロ固定	なし	
EXIT	ゼロ固定	ゼロ固定	なし	
SET_RATE	—	—	—	今後の拡張
GET_RATE	—	—	—	今後の拡張
GET_KIND	—	—	—	今後の拡張
CONNECT	ゼロ固定	送信先オブジェクト ID	接続先ノード ID：8bit 接続先オブジェクト ID：8bit	
DISCONNECT	ゼロ固定	送信先オブジェクト ID	なし	
GET_ERROR_LOG	—	—	—	今後の拡張

(3) コマンド返信メッセージ

コマンド返信メッセージの ASDU の利用方法を図 a-2-20 に示す。

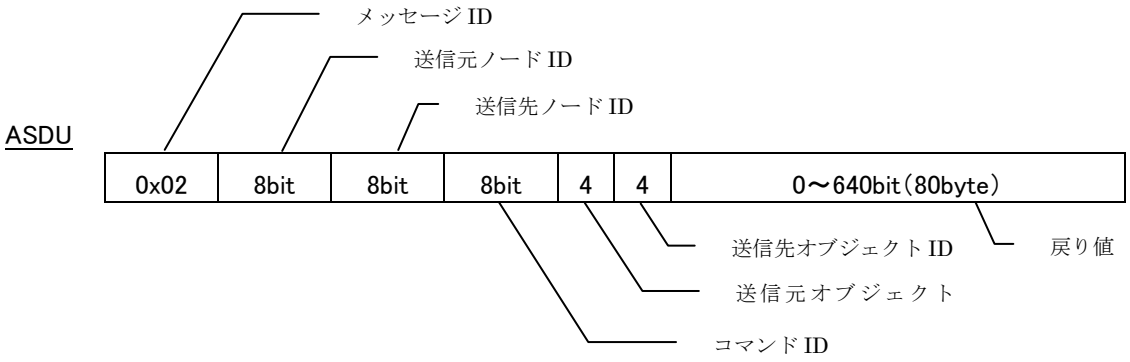


図 a-2-20 コマンド返信セージのデータフォーマット

コマンド毎の戻り値を表 a-2-18 に示す。

表 a-2-18 コマンド毎の戻り値

コマンド名	引数	備考
ACTIVATE	エラーコード：1byte	
DEACTIVATE	エラーコード：1byte	
RESET	エラーコード：1byte	
EXIT	エラーコード：1byte	
SET_RATE	—	今後の拡張
GET_RATE	—	今後の拡張
GET_KIND	—	今後の拡張
CONNECT	エラーコード：1byte	
DISCONNECT	エラーコード：1byte	
GET_ERROR_LOG	—	今後の拡張

（4）ショートデータ送信メッセージ

ショートデータ送信メッセージの ASDU の利用方法を図 a-2-21 に示す。

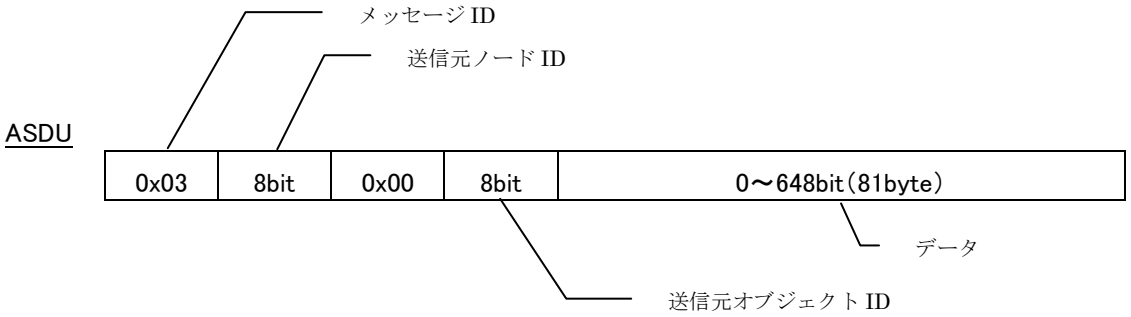


図 a-2-21　ショートデータ送信メッセージのデータフォーマット

(5) ロングデータ送信メッセージ（今後の拡張）

ロングデータ送信メッセージの ASDU の利用方法を図 a-2-22 に示す。

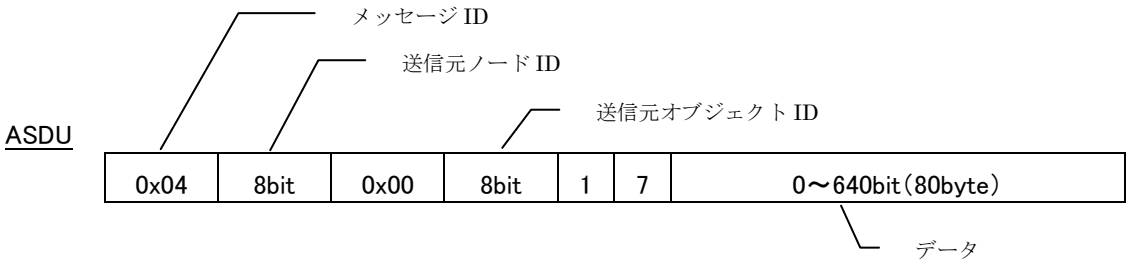


図 a-2-22 ロングデータ送信メッセージのデータフォーマット

ZigBee 通信では、一度に送信可能な PSDU(物理層データ・ペイロード)の最大サイズは 127byte であるが、MAC 層、ネットワーク層、及びアプリケーション・サポート副層のアドレス部を除くと 85byte となる。microRTCs では、メッセージ ID などの情報を付与するため、一度に送信可能なデータサイズを 81byte とする。そのため、大きなサイズのデータを送信するには、データを分割する必要がある。そこで、送受信データサイズが 81byte より大きい場合は、図 a-2-23 に示すように、データフィールドの先頭 1 バイトを利用し、データの分割送信を実現する。これにより、最大 10240byte（80byte × 128）のデータの送受信が可能となる。

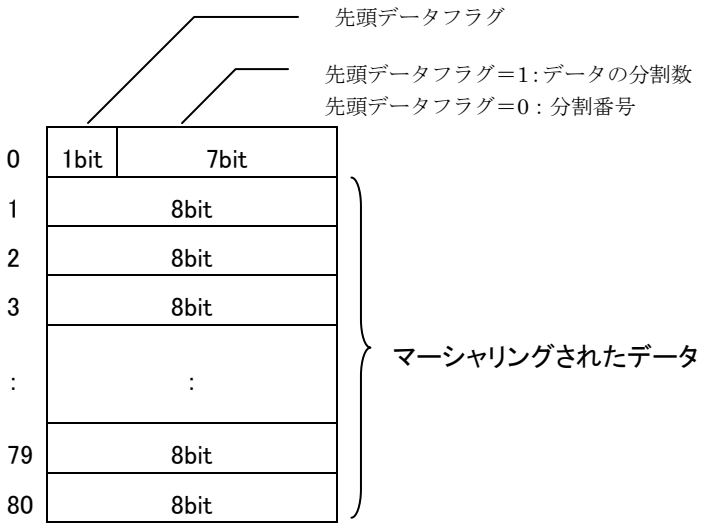


図 a-2-23 データ分割フォーマット

(6) ハートビートメッセージ

ハートビートメッセージの ASDU の利用方法を図 a-2-24 に示す。

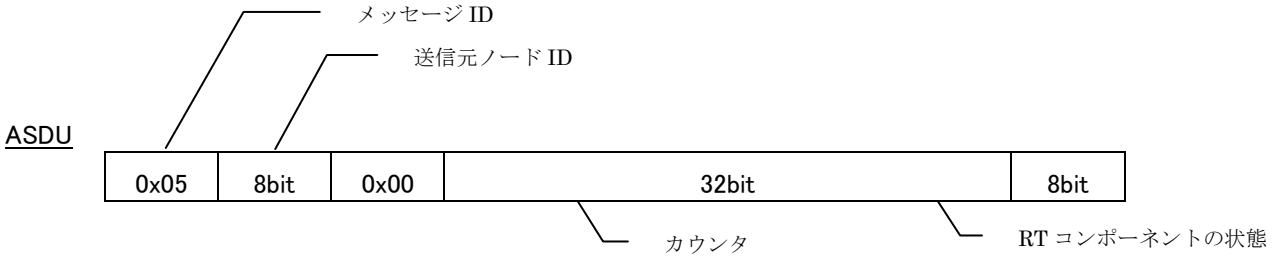


図 a-2-24 ハートビートメッセージのデータフォーマット

ASDU のカウンタは、ハートビートを送信する毎にカウントアップする。0 からカウントアップし、最大値(0xFFFFFFFF)を超えるとロールオーバーする。

RT コンポーネントの状態は、ステートマシンにおける RT コンポーネントの状態を表す。ID と RT コンポーネントの状態との対応を表 a-2-19 に示す。

表 a-2-19 RT コンポーネントの状態一覧

ID	RT コンポーネントの状態
0x00	未定義状態
0x01	Created 状態
0x02	Inactive 状態
0x03	Active 状態
0x04	Error 状態

a-2-4-4 デバイス RTC インタフェース関数

RT コンポーネントのアクションのインタフェース関数について、microRTCs での対応関係を表 a-2-20 に示す。各関数は、表 a-2-14 に示したタイミングで呼ばれる。

表 a-2-20 アクションインタフェース関数一覧

インタフェース関数	対応	備考
Rtc_onInitialize	○	
Rtc_onActivated	○	
Rtc_onExecute	○	
Rtc_onDeactivated	○	
Rtc_onAborting	○	
Rtc_onReset	○	
Rtc_onError	○	
Rtc_onFinalize	○	
Rtc_onStateUpdate	×	
Rtc_onStartup	×	
Rtc_onShutdown	×	

a-2-4-5 データポートインタフェース

データポートのインタフェース関数を表 a-2-21 に示す。

表 a-2-21 データポートインタフェース関数一覧

インタフェース関数	処理概要	備考
DataPort_getInPortID	IN ポート ID を取得する	
DataPort_getOutPortID	OUT ポート ID を取得する	
DataPort_inPortRead	IN ポートからデータを読み込む	
DataPort_outPortWrite	OUT ポートへデータを書き込む	

表 a-2-21 に示したデータポートのインタフェース関数の詳細を以下に述べる。

(1) DataPort_getInPortID

DataPort_getInPortID

[概要]

未使用の IN ポート ID を取得する。
関数を呼ぶごとに、0x01 からポート ID が返される。

[C 言語記述形式]

UByte DataPort_getInPortID(void)

[パラメタ]

なし。

[復帰値]

シンボル / 値	説明
0x01～0x0F	ポート ID の取得に成功
0xF	ポート ID の空きがない

[補足説明]

取得可能なポート ID は、0x01～0x0F。

(2) DataPort_getOutPortID

DataPort_getOutPortID

[概要]

未使用の OUT ポート ID を取得する。
関数を呼ぶごとに、0x01 からポート ID が返される。

[C 言語記述形式]

UByte DataPort_getOutPortID(void)

[パラメタ]

なし。

[復帰値]

シンボル / 値	説明
0x01～0x0F	ポート ID取得に成功
0xFF	ポート ID の空きがない

[補足説明]

取得可能なポート ID は、0x01～0x0F。

(3) DataPort_inPortRead

DataPort_inPortRead

[概要]

指定した IN ポート ID のデータを取得する。

[C 言語記述形式]

RCode DataPort_inPortRead(UByte portID, UByte *dataBuffer, UInt32 *dataSize)

[パラメタ]

I/O	パラメタ	説明
I	portID	ポート ID（0x01～0x0F）
O	dataBuffer	取得データ格納バッファ
O	dataSize	取得データサイズ （受信データがなければゼロが設定される）

[復帰値]

シンボル / 値	説明
E_OK	正常復帰
E_ID	0x01～0x0F 以外のポート ID がパラメタで指定された
E_PORT	未使用のポート ID もしくは OUT ポート ID がパラメタで指定された

[補足説明]

取得可能なデータの最大サイズは 81byte。

(4) DataPort_outPortWrite

DataPort_outPortWrite

[概要]

指定した OUT ポート ID へデータを設定する。

[C 言語記述形式]

RCode DataPort_outPortWrite(UByte portID, const UByte *dataBuffer, UInt32 dataSize)

[パラメタ]

I/O	パラメタ	説明
I	portID	ポート ID（0x01～0x0F
I	dataBuffer	設定データ格納バッファ
I	dataSize	設定データサイズ（0～81） （ゼロを指定すると何も設定されない）

[復帰値]

シンボル / 値	説明
E_OK	正常復帰
E_ID	0x01～0x0F 以外のポート ID がパラメタで指定された
E_PORT	未使用のポート ID もしくは IN ポート ID がパラメタで指定された
E_SIZE	82byte 以上のサイズがパラメタで設定された

[補足説明]

設定可能なデータの最大サイズは 81byte。

a-2-4-6 特記事項

データポートのデータ送信時において、以下の注意点がある。

- 複数ポートへの出力が重なった場合、ポート ID の小さいポートからデータを送信する。
- 送信に失敗した場合であっても送信要求をクリアする。

a-2-4-7 エラーコード一覧

表 a-2-22 及び表 a-2-23 にエラーコード一覧を示す。

表 a-2-22 ReturnCode_t 型エラーコード一覧

ID	エラーコード	内容
RTC_OK	0x00	操作が成功した
RTC_ERROR	0x01	一般的なエラー
RTC_BAD_PARAMETER	0x02	無効な引数を渡した
RTC_UNSUPPORTED	0x03	サポートされていない操作を行った
RTC_OUT_OF_RESOURCES	0x04	操作を行うために必要なリソースが不足している
RTC_PRECONDITION_NOT_MET	0x05	操作のための前提条件が満たされていない

表 a-2-23 Rcode 型エラーコード一覧

ID	エラーコード	内容
E_OK	0x00	正常終了
E_ERROR	0x11	異常終了
E_BUSY	0x21	ビジー状態
E_ID	0x22	ID 異常
E_NOINIT	0x23	未初期化状態
E_SIZE	0x24	データサイズ異常
E_STATE	0x25	状態異常
E_OVER	0x26	領域を超える
E_SETTING	0x27	設定状態異常
E_EVENT	0x28	イベント異常
E_PAR	0x29	パラメタ異常
E_PORT	0x30	ポート異常
E_NODATA	0x31	データがない

(a-3) RTC-Lite およびプラグアンドプレイに対応した RT ミドルウェア開発支援ツールの研究開発
(委託先:株式会社テクノロジックアート)
a-3-1 概要
本事業における開発目標ならびに成果を以下にまとめる。

表 a-3-1 開発目標とそれに対する研究開発成果ならびに達成度

目標	研究開発成果	達成度
①RT 要素部品(RT コンポーネント)化支援ツールの開発		
組み込み MPU 上で動作する組み込み RT コンポーネントの仕様から雛形コードを自動生成するツールの設計, 開発	本ツールでは, 本プロジェクトにて開発を行った各種基盤通信モジュールおよび汎用的な組み込み MPU 上で動作する RT コンポーネントの雛形コードを生成可能である. 本ツールにて生成可能な RT コンポーネントを以下に示す. ・高速制御用(CAN 用)RTC-Lite(miniRTC) ・低速制御用(Zigbee 用) RTC-Lite (microRTC) ・PIC/dsPIC 用 RTC-Lite	達成
通常の RT コンポーネントを作成する場合と類似の操作で, 組み込み RT コンポーネントの雛形コードを生成する機能.	本ツールは, 既存ツールとの操作性の統一, 連携の容易化を図るため, 独立行政法人 産業技術総合研究所が開発を行っている RTCBuilder を拡張する形式で Eclipse プラグインとして実現している.	達成
生成対象コンポーネントの仕様を,「RT コンポーネント仕様記述方式」を拡張した形式として保存/読み込みする機能.	本ツールは各組み込み MPU に固有な拡張情報を,「RT コンポーネント仕様記述方式(RTCProfile)」の拡張領域に保持する形式とし, 生成したコンポーネント情報の保存／読み込みを行えるようにした.	達成
プラグイン方式を採用し, 用途に応じて各種機能追加, カスタマイズを容易に実行できる構成.	各 RT コンポーネント向けの雛形コード生成ツールは, 別々の独立したプラグインとして実現しているため, 利用者が使用したい RT コンポーネント用のツールのみを選択, インストールする事が可能となっている. また, 必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで, 利用者自身が使用する開発環境全体をカスタマイズする事ができるようになっている.	達成

デバイス分類毎に共通な制御・信号処理の部分を生成する機能.	PIC/dsPIC 用 RTC-Lite 向けツールについては，コンポーネント・パターンの仕組みを採用し，使用頻度が高いと考えられる「デジタル入力」「デジタル出力」「アナログ入力」「PWM 制御」の種類を選択するだけで，より詳細なコードを自動生成することができる．また，他の RT コンポーネントについては，ECHONET(Energy Conservation and Homecare Network)にて規定されている各種標準コマンドを送受信するためのインターフェース定義に対応した雛形コード生成機能を実現するとともに，ユーザーが設定画面にてカスタマイズできるようになっている.	達成
②RT 要素部品(RT コンポーネント)コンフィギュレーション支援ツールの開発		
組み込み RT コンポーネントのコンフィギュレーション情報の設定支援を行う機能	本ツールでは，RT コンポーネント(組み込み RT コンポーネント含む)にて必要となる各種コンフィギュレーション情報の入力項目の名称や単位変換などを行い，設備機器メーカー視点からの各種項目設定を実現した．また，各種パラメータに対して設定された制限値を用いることで，利用者が誤って不正な値を設定することを防止する機能を実現した.	達成
プラグイン方式を採用し，必要に応じて各種機能追加，カスタマイズを容易に実行できる構成.	本ツールは，既存ツールとの操作性の統一，連携の容易化を図るため，独立行政法人　産業技術総合研究所が開発を行っている RTSystemEditor を拡張する形式で Eclipse プラグインとして実現している．このため，必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで，利用者自身が使用する開発環境全体をカスタマイズする事ができるようになっている.	達成

類似デバイスの設定情報の再利用などを容易に行える機能.	本ツールでは，各種設定に用いる情報をチューニング定義ファイルとして，外部ファイルに保持する形式を採用している．そして，必要に応じてこの情報のみのインポート/エクスポートが行えるようになっている．このため，類似の RT コンポーネントを使用してシステムを構築する場合は，チューニング定義ファイルを再利用することで，各種詳細設定を行う負担を削減でき，開発効率を向上させることができる.	達成

③RT システム構築支援ツールの開発		
通常の RT コンポーネントのみを用いて RT システムを構築する場合と類似の操作で，組み込み RT コンポーネントを含んだ RT システムを構築する機能.	今回開発したシステムでは，システムの可用性・信頼性を高めるため，コントローラが複数箇所に分散配置されている．また，高速制御用 RTC-Lite(miniRTC)，低速制御用 RTC-Lite(microRTC)は，実際の制御/センサ用基盤の上で各コンポーネントが動作しているが，これらのデバイスを設備機器アプリケーション単位で管理するためのモジュール(RTC-Lite Manager)が，基盤通信モジュール上で動作している．そこで，これらの仕組みを隠蔽し，通常の RT コンポーネントと同様に各種デバイス向けコンポーネントを操作するためのツールを開発した．また，各デバイス単位での操作，情報取得を実現するとともに，デバイス間のポートの接続/切断を実行する機能を実現した.	達成
システムインテグレータやある程度のスキルを持ったエンドユーザーが，容易に RT システムを構築することができる RT システム構築支援ツールの研究開発	今回開発を行ったシステムでは，要素部品管理モジュール上の RTCHub が，各種機器単位で管理，制御するため，RTC-Lite Manager 毎に疑似複合コンポーネントを作る形で動作している．そこで， RTCHub を構築する際に必要となるプロファイルテーブル情報を生成するツールを開発した．この機能を利用することで，システムインテグレーターやある程度のスキルを持ったエンドユーザーが，通常の RT コンポーネントのみを用いて RT システムを構築する場合と類似の操作で，組み込み RT コンポーネントを含んだ RT システムを容易に構築することが可能となる. また，オフライン状態で構築した RT システムの情報から，実際のシステムを起動するためのツールを開発した.	達成
構築した RT システムの情報を「RT システム仕様記述方式」を拡張した形式で保存/読み込みする機能.	RTCHub の仕組みを隠蔽するためのツールおよびプラグアンドプレイ機能に必要となる情報を設定するツールを用いて設定を行った内容を，「RT システム仕様記述方式(RTSProfile)」を利用して保存/読み込みができる機能を実現した.	達成
プラグイン方式を採用し，用途に応じて各種機能追加，カスタマイズを容易に実行できる構成.	本ツール群は，既存ツールとの操作性の統一，連携の容易化を図るため，独立行政法人 産業技術総合研究所が開発を行っている RTSystemEditor を拡張する形式で Eclipse プラグインとして実現している．このため，必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで，利用者自身が使用する開発環境全体をカスタマイズする事ができるようにな	達成

	っている.	
プラグアンドプレイ機能を実現するために必要な各種情報を設定可能であるとともに，構築したRTシステムのRTシステム仕様記述情報ファイルの内容から，各 RT コンポーネントを設定された手順で起動/終了可能なツールの研究開発.	プラグアンドプレイ設定ツールでは，プラグアンドプレイ機能を実現するために必要となる各種情報を RT システム構築時に設定するとともに，設定したプラグアンドプレイ情報を用いて，実際の各 RT コンポーネントを制御する機能を実現している．プラグアンドプレイ機能を実現するために，対象システムを構成する各コンポーネントのアクション実行順序，アクション実行条件を設定することができるようになっている．また，設定したプラグアンドプレイ情報に基づいて，指定された実行順序でアクションを実行するとともに，制約条件が成立していない場合のアクションの実行待機などを制御する機能も実現している.	達成
④RTシステム開発支援ツールの開発		
よりエンドユーザーに近い立場の人が，RT コンポーネントの仕様から，できるだけノンプログラミングで RT システムとして提供すべきサービスを構築する機能.	本ツールは，対象となるRTシステム内に配置されたRTコンポーネントの仕様情報(RTCProfile)から，システム全体の振る舞いを定義する機能を実現している．また，作成した振る舞いに従って，システム全体を制御する機能も実現している．更に，システムの振る舞い定義は，グラフィカルなエディタを用いて，プログラミングレスで行うことが可能である.	達成
定義済みのサービス情報を再利用し，効率的にシステム構築を行うための機能.	既存の状態遷移定義を再利用したり，粒度が細かくより詳細な状態遷移定義から，より広い視点での状態遷移を定義するために，複数の状態遷移をマージする機能も実現している．この機能を利用することで，ユーザーは類似のサービスを構築する際に，過去の資産を再利用することができ，開発効率を向上させることができる．また，定義した振る舞い，サービスを実現するために必要となる各 RT コンポーネント間の接続情報をRTシステム仕様記述方式（RTSProfile）の形式で自動生成することができるため，対象システムを容易に構築することが可能となる.	達成

システム開発を有効に支援するツールチェーンを構築するためには，各種支援ツールに要求される機能および想定される利用者を明確にする必要がある．また，一般的なシステム開発では，複数の関係者(ステークホルダー)が，それぞれの役割に応じて，段階(フェーズ)毎に様々な開発作業を実施するが，その内容は全体の開発サイクル，対象となるシステム構成によって大きく変化する．このため本研究開発では，最初に本プロジェクトが対象としている「住宅システム」に関して，全体のシステム構成の検討，ステークホルダーおよび開発サイクルの明確化を行った．

「住宅システム」では，各種家電製品や窓・サッシなどの建具製品など完成した複数の設備機器を組み合わせる形で全体のシステムを構成する．また，各設備機器はアクチュエータやセンサなど複数の要素部品を組み合わせる事で必要な機能を実現する．従って，建物内におけるインテリジェントな環境埋め込み型ロボットシステムを構築する場合には，まずはモジュールメーカーが各要素部品を RT 化する必要がある．そして，設備機器メーカーがこれら RT 化された要素部品を組み合わせて設備機器単位でサブシステムを構成するとともに，住宅メーカー/システムインテグレーターがこれらを組み合わせて全体システムを構築する必要がある．

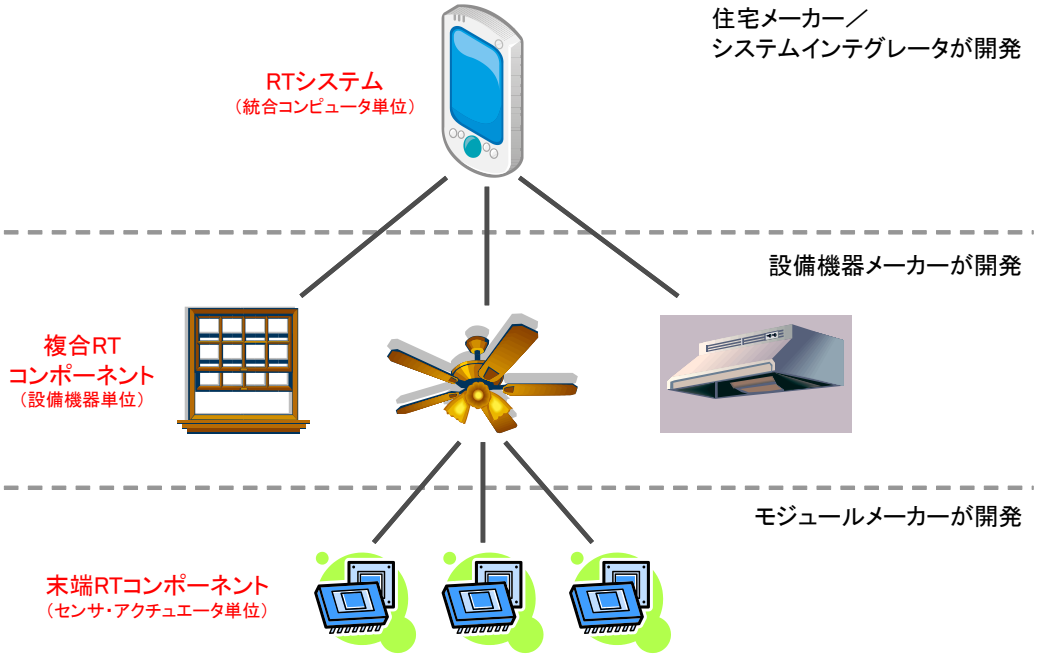


図 a-3-1 住宅システム全体の構成

更に，このような構成をベースとした場合，各機器で使用可能な計算リソースもそれぞれの階層に応じて制限がでてくるので，利用可能な RT コンポーネントの種類，構成も階層構造に応じた形となることが想定される．つまり，ベースとなる各種要素部品は少ないリソースでも動作する必要があるとともに，各種設備機器および全体システムとしては，下位の要素を統合管理する必要がある．従って，システムを構成する RT コンポーネントは以下のような構造となることが想定される．

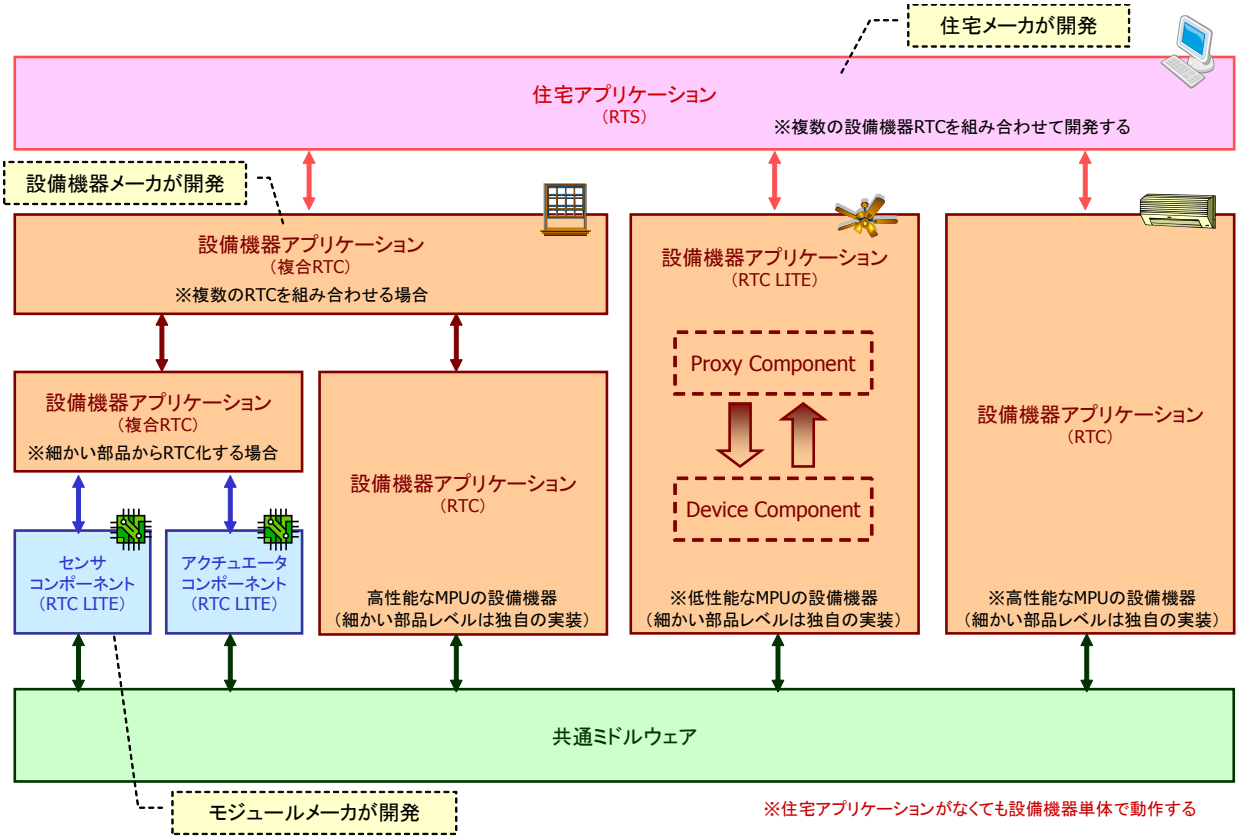


図 a-3-2 住宅システム全体のソフトウェア構成

次に，上述の構成をベースに，「住宅システム」を開発する際に関係するステークホルダーおよび各フェーズにおける成果物の洗い出し，開発サイクルの検討を行った．その結果，システム開発全体のフェーズとしては，大きく住宅そのものの機能／サービスを開発する「商品開発フェーズ」と，開発したサービスを基に個々の住宅を構築する「個別住宅建設フェーズ」に分類できる事がわかった．また，「商品開発フェーズ」では，「住宅メーカー」からの要件を基に，「システムインテグレーター」が全体システムである住宅アプリケーションの開発を行い，基本となる RT システムの設計も行うことがわかった．また「設備機器メーカー」からの要件を基に，「モジュールメーカー」が設備機器アプリケーションの開発や個々のコンポーネント開発を行い，基本的な RT コンポーネント／複合 RT コンポーネントの設計を行うことがわかった．そして，場合によっては「設備機器メーカー」自身が「モジュールメーカー」として設備機器アプリケーションの開発を行う場合があることもわかった．一方，「個別住宅建設フェーズ」では，「顧客」からの要件を基に，「住宅メーカー」が住宅アプリケーションのカスタマイズを行い，ユーザニーズに対応した RT システムを構築するとともに，「設備機器メーカー」に対して設備機器アプリケーションのカスタマイズを依頼することがわかった．そして，「設備機器メーカー」は依頼内容を基に，設備機器アプリケーションのカスタマイズを実行し，実際の住宅に設置する RT コンポーネント／複合 RT コンポーネントを構築することがわかった．また，最終的には各要素の住宅への設置は「施工業者」が行い，現地での最終的なカスタマイズ，設置，動作確認を行うことがわかった．

このような分析結果から，「商品開発フェーズ」においては，住宅システムを構築する上では，「システムインテグレーター」が住宅アプリケーションを開発する際に RT システムの構築を支援するツール，「モジュールメーカー」が設備機器アプリケーションを開発する際に複合 RT コンポーネントの構築を支援するツール，またコンポーネントを開発する際に RT コンポーネント化を支援するツールが有効である．また「個別住宅建設フェーズ」においては，「住宅メーカー」および「設備機器メーカー」がそれぞれ，住宅アプリケーションおよび設備機器アプリケーションをカスタマイズすることを支援するツール，また「施工業者」が現地で各要素を設置する際に各種パラメータの調整，テストを行うことを支援するツールが有効である．

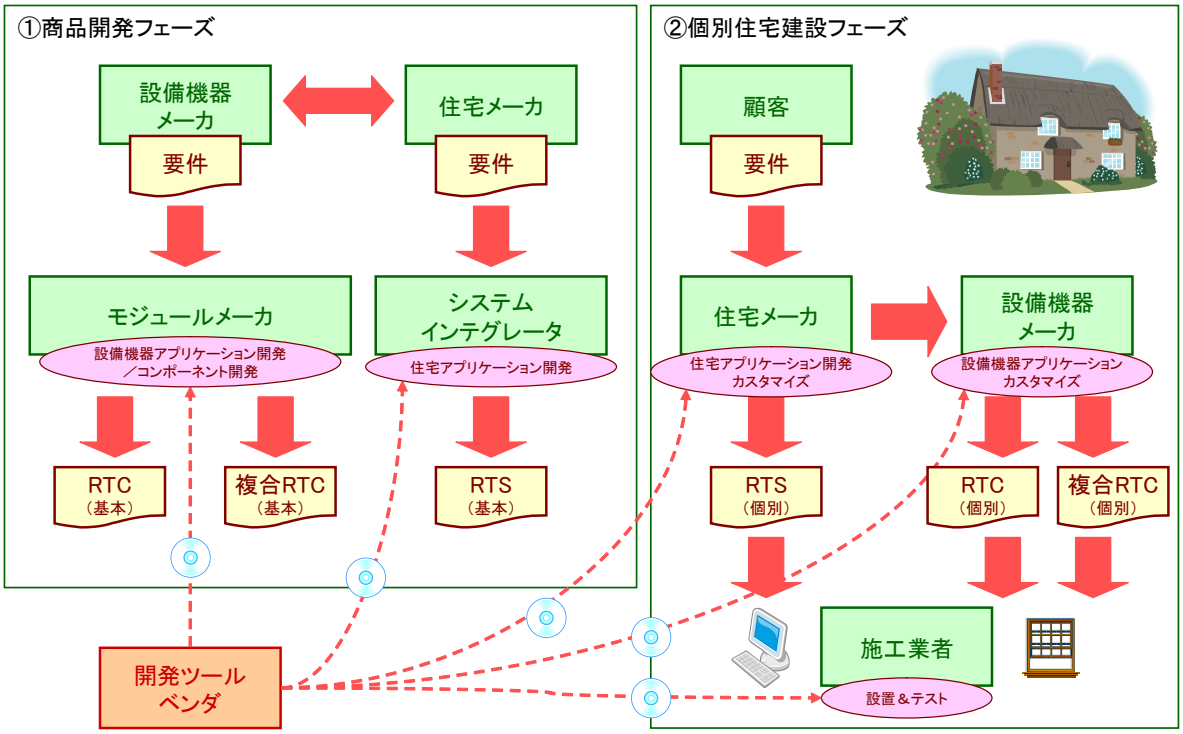


図 a-3-3 住宅システム全体の開発サイクルと支援ツール

a-3-2　RT 要素部品（RT コンポーネント）化支援ツールの開発

◎目的

本ツールは，「モジュールメーカー」(RT 要素部品開発機関)が，個々のコンポーネント開発する際の開発負荷を低減させることを目的としている．「住宅システム」における各種要素部品を RT コンポーネント化する際には，コストの面からもより安価な組み込み MPU 上での動作が要求される．また，RT 要素部品（RT コンポーネント）のソースコードは定型的な部分が多く，開発効率を高めるためにはツールによってこの部分を自動生成することが効果的である．本ツールでは，組み込み MPU 上で動作する RT 要素部品（RT コンポーネント）の仕様情報から雛形ソースコードを生成し，RT 要素部品開発機関による RT 要素部品（RT コンポーネント）の開発を支援する．また，ユーザの利便性を高めるため，他の RT ミドルウェア向けツールとの操作性の統一，連携の容易化を図る．そして，通常の RT コンポーネントの仕様を記述する際にも用いている「RT コンポーネント記述方式」を拡張する形で，生成対象の組み込み MPU 上で動作する RT コンポーネントの仕様を，保存／読み込みできるようにする．更に，デバイス分類毎に共通な制御・信号処理についてもカスタマイズした上で，ソースコード上に生成できるようにする．また，ツール自身の実装形態としては，プラグイン方式を採用し，ツール群全体として各種機能追加，カスタマイズを容易に実行できる構成とする．本ツールを利用することで，RT 要素開発機関はミドルウェアに依存した定型的なコードを実装する必要がなくなるため，本来の注力点であるコアロジックの開発に専念できるようになる．全体システムの中において，本ツールが支援を行う範囲を以下に示す．

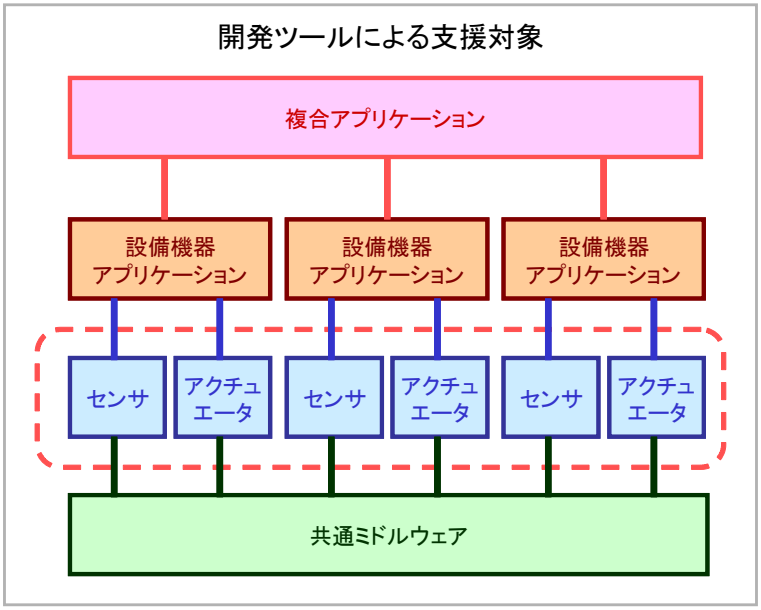


図 a-3-4 RT 要素部品（RT コンポーネント）化支援ツールの適用範囲

◎全体概要

本ツールでは，本プロジェクトにて開発を行った各種基盤通信モジュールおよび汎用的な組み込み MPU 上で動作する RT コンポーネントの雛形コードを生成可能である．本ツールにて生成可能な RT コンポーネントを以下に示す．

・高速制御用(CAN 用)RTC-Lite(miniRTC)
・低速制御用(Zigbee 用)RTC-Lite(microRTC)
・PIC/dsPIC 用 RTC-Lite

本ツールは，既存ツールとの操作性の統一，連携の容易化を図るため，独立行政法人　産業技術総合研究所が開発を行っている RTCBuilder を拡張する形式で Eclipse プラグインとして実現している．また，各 RT コンポーネント向けの雛形コード生成ツールは，別々の独立したプラグインとして実現しているため，利用者が使用したい RT コンポーネント用のツールのみを選択，インストールする事ができるようになっている．そして，必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで，利用者自身が使用する開発環境全体をカスタマイズする事ができるようになっている．

以下に開発した各ツールの画面例を示す．

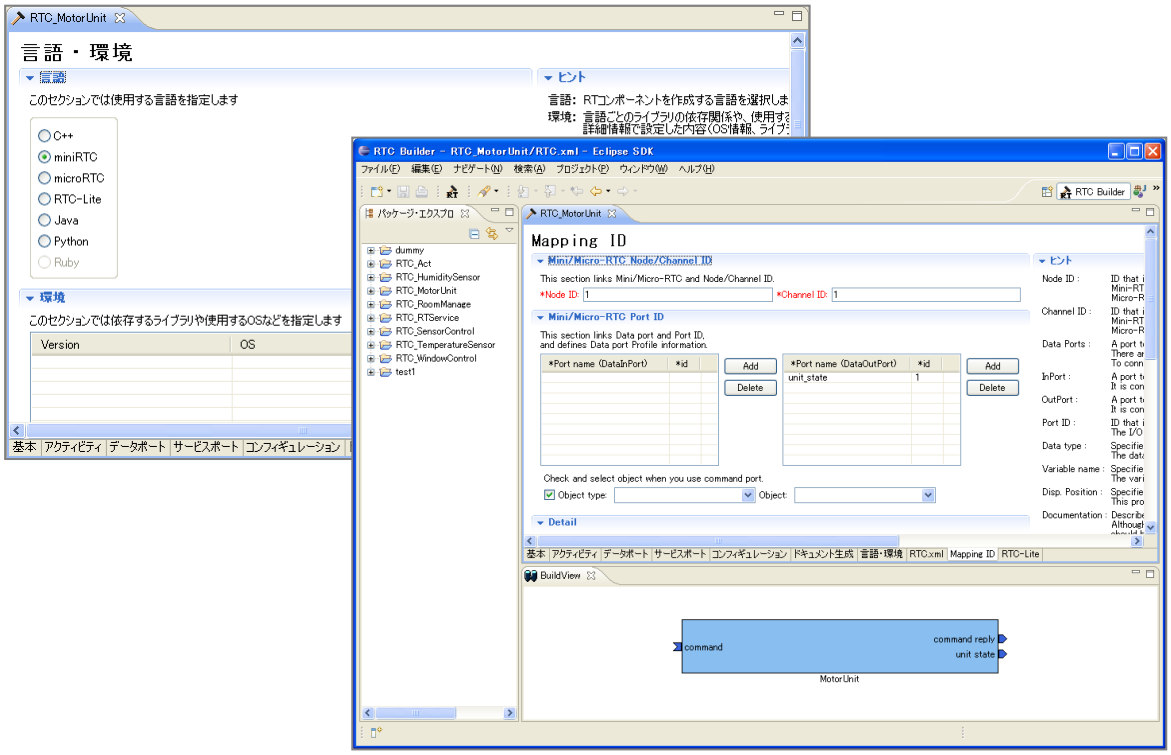


図 a-3-5 miniRTC/microRTC 用

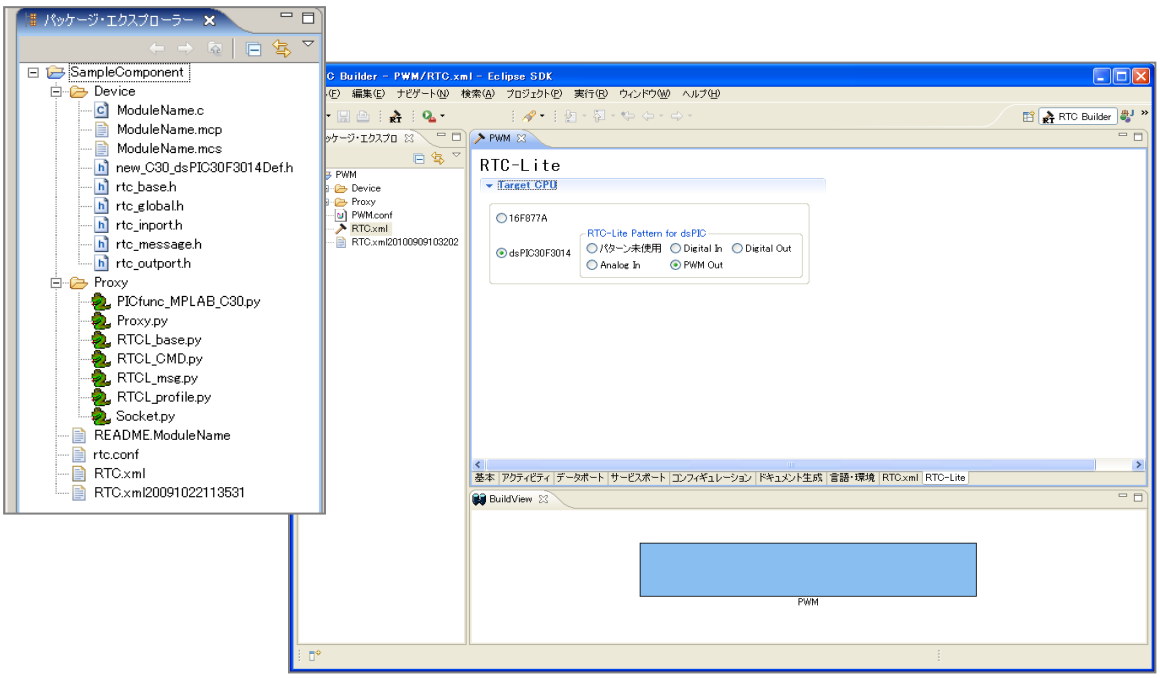


図 a-3-6 PIC/dsPIC 用

汎用的な RT コンポーネントの仕様を記述する形式として，独立行政法人 産業技術総合研究所では，「RT コンポーネント仕様記述方式(RTCProfile)」を規定している．この仕様では，RT コンポーネントにおいて基本となる部分の仕様，情報を規定するとともに，各ユーザ独自の固有情報を保持するための拡張領域が用意されている．そこで，本ツールは各組み込み MPU に固有な拡張情報をこの RTCProfile の拡張領域に保持する形式とし，生成したコンポーネント情報の保存／読み込みを行えるようにした．ツールでの設定項目と，各拡張領域の内容の対応関係の

例を以下に示す.

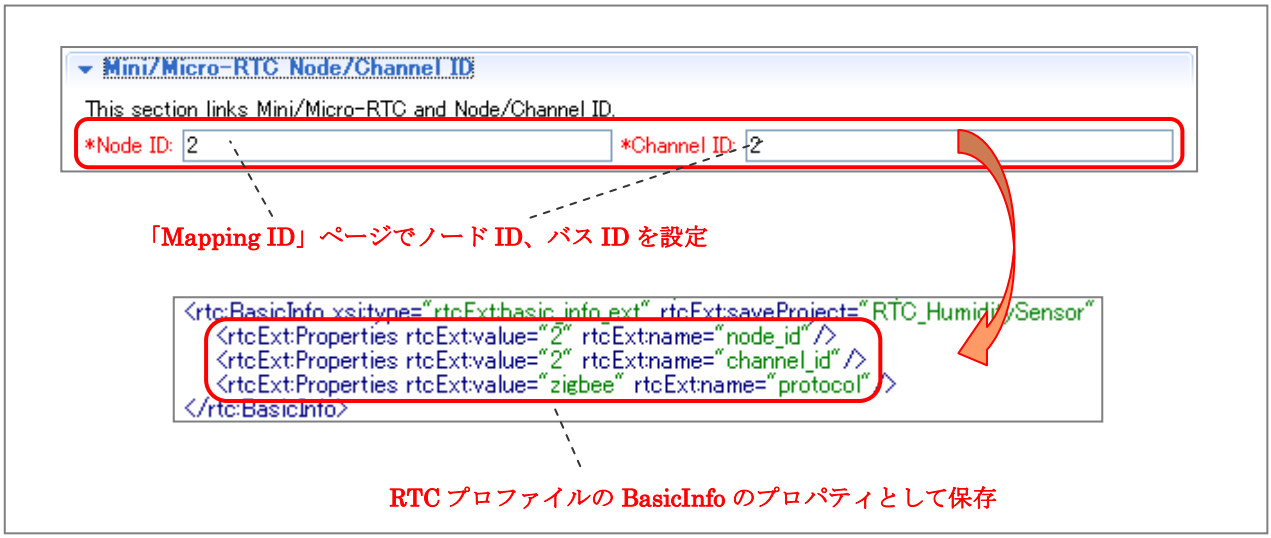


図 a-3-7 ノード ID，バス ID

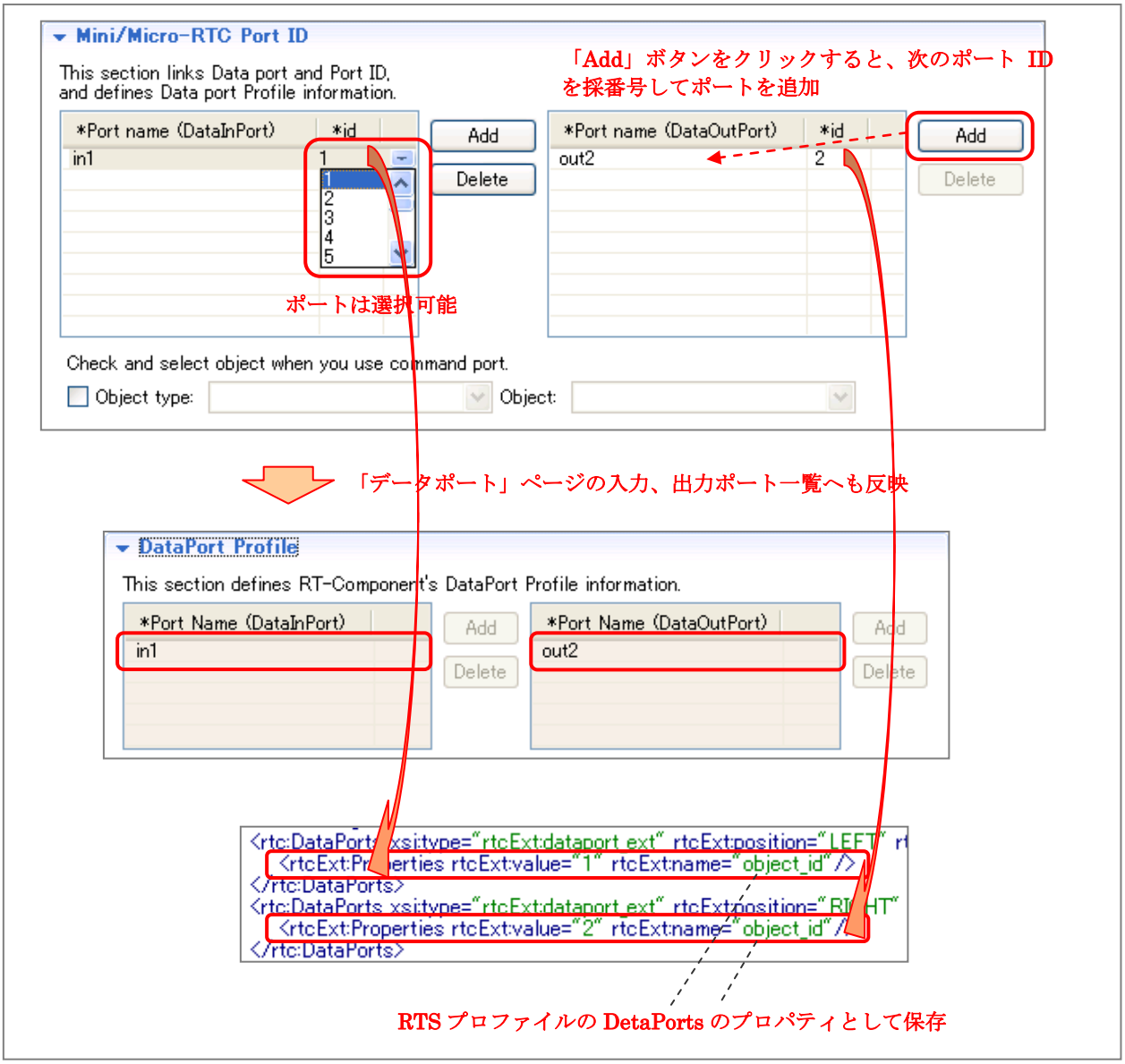


図 a-3-8 ポート

PIC/dsPIC 用 RTC-Lite については，利用可能なリソースが非常に限られているため，RT コンポーネント自身の用途としてもある程度限定された範囲で定式化(コンポーネント・パターン化)できることが大阪大学から提唱されている．このため，PIC/dsPIC 用 RTC-Lite 向けツールについては，ユーザの負担を更に減らすため，大阪大学が開発・提唱しているコンポーネント・パターンの仕組みを採用した．PIC/dsPIC 用 RTC-Lite として使用頻度が高いと考えられる「デジタル入力」「デジタル出力」「アナログ入力」「PWM 制御」については，コンポーネント種類を選択するだけで，より詳細なコードを自動生成することができる．

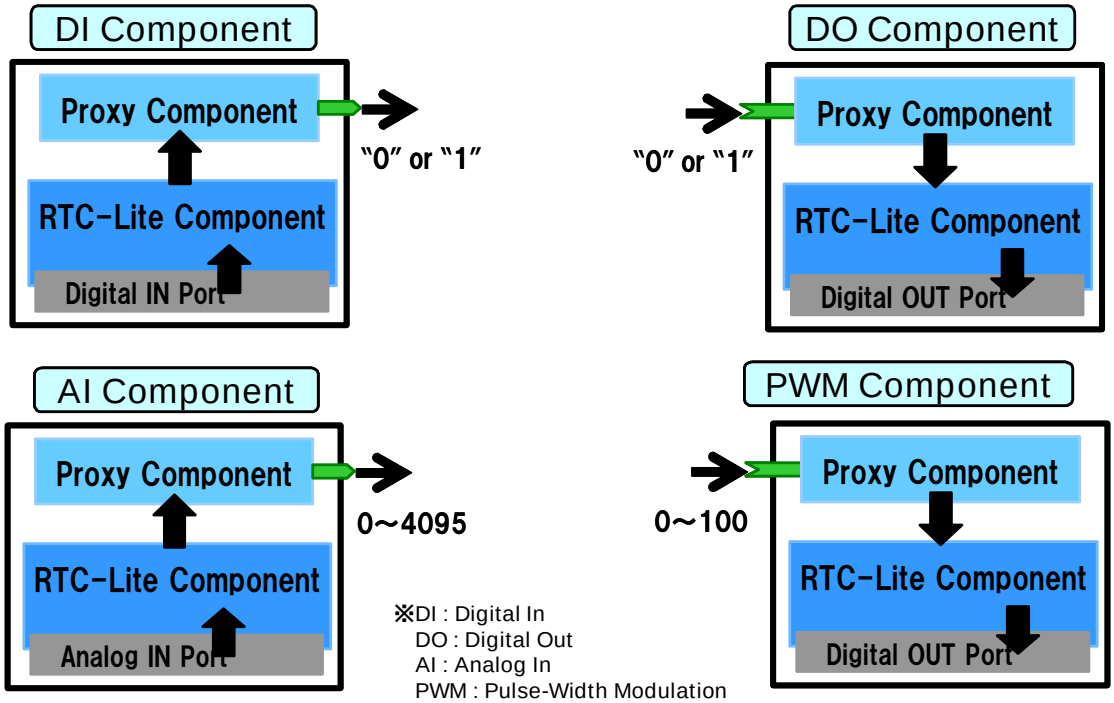


図 a-3-9 コンポーネント・パターンの採用

miniRTC/microRTC については，使用しているプロトコル(CAN/Zigbee)自体が広く普及しているとともに，採用している組み込み MPU も使用可能なリソースが比較的リッチであるため，RT コンポーネントの適用範囲はより広くなる．しかし，本プロジェクトにて対象としている「住宅システム」について考えた場合，RT コンポーネントの制御対象としては，家電などのホームネットワーク機器が多くなることが想定される．一方，家庭内の機器などを標準的なインターフェース/プロトコルを介して相互接続するための仕様として，ECHONET（Energy Conservation and Homecare Network）の標準化が行われている．そこで本ツールでは，ECHONET にて規定されている各種標準コマンドを送受信するためのインターフェース定義に対応した雛形コード生成機能を実装した．本機能では，ECHONET をベースとして，機器分類とコマンド候補を定義できるとともに，各オブジェクトが使用するコマンド群については，ユーザーが設定画面にてカスタマイズできるようになっている．このため，本機能を用いることで，必要なコマンドを送受信可能な RT コンポーネントの雛形を容易に生成することが可能である．

オブジェクト種別		オブジェクト	
空調関連機器	AirConditioner	家庭用エアコン	HomeAirConditioner
		扇風機	ElectricFan
住宅・設備関連機器	HousingFacilities	一般照明	GeneralLighting
音響・映像関連機器	AudioVisual	デジタルテレビ	DigitalTelevision
移動家具関連機器	MobileFurniture	テーブル	MobileTable
		椅子	MobileChair
		ドア	AutoDoor
		配膳カート	MobileDinningWagon

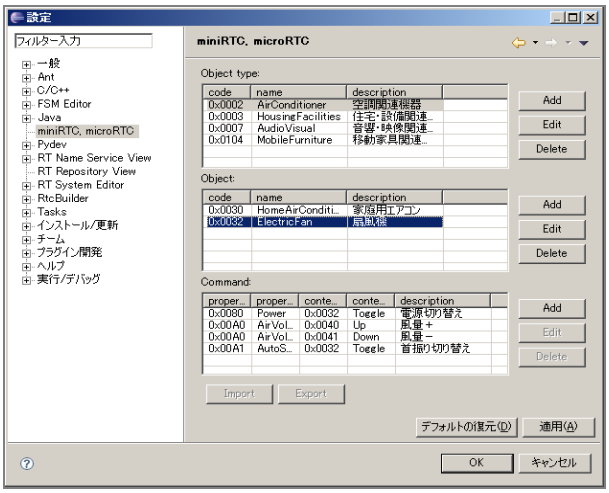


図 a-3-10 汎用コマンド(例)

◎詳細内容

〇機能概要

本ツールは大きく3つの機能から構成される．1つ目は，独立行政法人 産業技術総合研究所（AIST）が規定している「RT コンポーネント仕様記述方式」に基づき RT 要素部品（RT コンポーネント）の仕様を編集する機能である．2つ目の機能は，編集した内容に基づき，ソースコードの雛形を生成する機能である．3つ目の機能は，編集した RT 要素部品（RT コンポーネント）の仕様を「RT コンポーネント仕様記述方式」に基づき，XML 形式，YAML 形式にてインポート／エクスポートする機能である．以下に UML のユースケース図で表した機能概要図を示す．

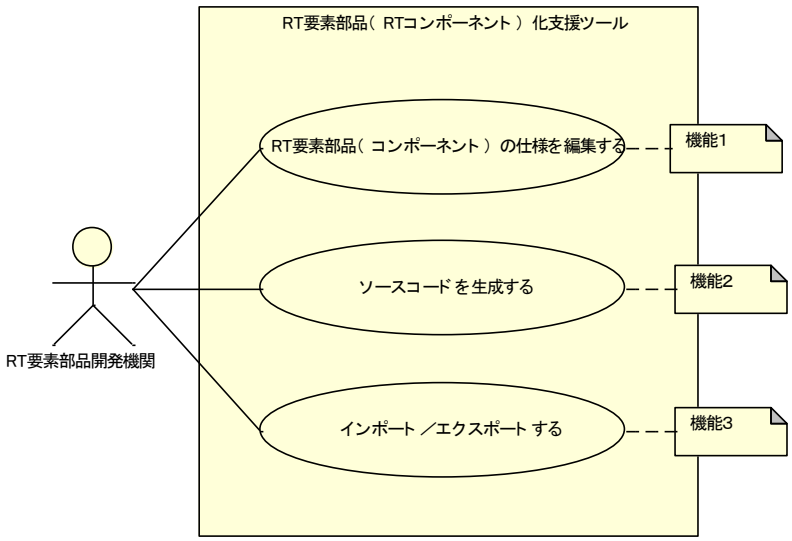


図 a-3-11 RT 要素部品(RT コンポーネント)化支援ツールのユースケース図

○RT コンポーネント仕様記述方式の概要

本ツールで編集する RT 要素部品(RT コンポーネント)の仕様情報は，独立行政法人 産業技術総合研究所(AIST)の「RT コンポーネント仕様記述方式」によって管理される．以下にこの方式の概要と，この方式で記述された RT 要素部品(RT コンポーネント)の仕様情報がソースコードへどのように反映されるかを示す．

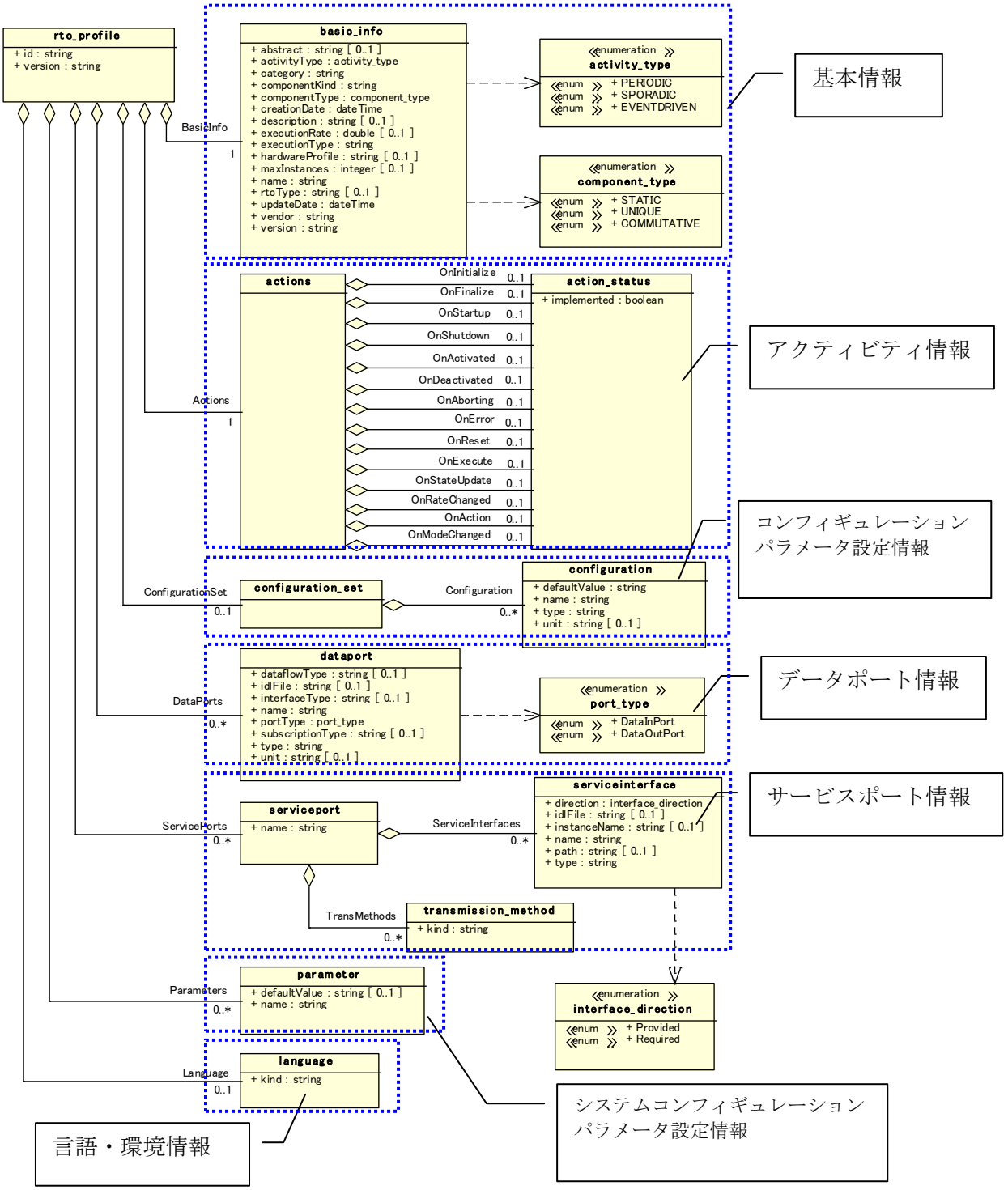


図 a-3-12 「RT コンポーネント仕様記述方式」の構成

基本情報には，当該 RT 要素部品（RT コンポーネント）の名称，種類，バージョン，開発元ベンダ等の情報が定義される．ここに定義された情報は，RTM 上で参照される値となるため，ソースコード上でも変数等で保持される．

アクティビティ情報には，各イベント（Initialize，Finalize 等）発生時に実行されるアクションが当該 RT 要素部品（RT コンポーネント）で実装されているかどうかについて定義される．実装されている（implemented=true の）アクションは，ソースコード上で関数やメソッドとして対応する処理が記述される．

データポート情報には，RT 要素部品（RT コンポーネント）同士がデータ交換する際の接続端情報（ポートの名称，

種類，変数名，データの単位等）が定義される．ソースコード上では，データポートに対応する変数が定義され，その変数を介した外部コンポーネントとのやり取りの処理が記述される．

サービスポート情報には，RT 要素部品（RT コンポーネント）同士が相互作用する際の接続端情報が定義される．なお，RTC-Lite フレームワークでは，サービスポートは使用できない．

コンフィギュレーションパラメータ情報には，当該 RT 要素部品（RT コンポーネント）に関する何らかの設定情報が定義される．

システムコンフィギュレーションパラメータ情報には，当該 RT 要素部品（RT コンポーネント）がシステム上で動作する際の設定情報が定義される．なお，RTC-Lite フレームワークにおいては，システムコンフィギュレーションパラメータ情報は使用できない．

言語・環境情報には，当該 RT 要素部品（RT コンポーネント）の動作環境となる対象プラットフォーム情報（プログラミング言語，OS，CPU，ライブラリ等）が定義される．本ツールではこの情報に基づき，対象となるソースコードの生成を行う．

〇全体アーキテクチャ概要

本ツールは統合開発環境（IDE）Eclipse プラットフォームのプラグインとして動作することを前提とする．さらに，本ツールのソースコード生成機能自体も Eclipse プラグインとして独立させ，必要なプラットフォーム（プログラミング言語，OS，CPU 等）に応じてプラグインを本体側に組み込み，容易に拡張できるような方式を採用する（以降では、この Eclipse プラグインを「言語プラグイン」と呼ぶ）．また，本ツールの開発では以下の外部ライブラリを使用する．

・ソースコード生成：velocity
・図形描画：EMF および GEF
・XML 入出力：JAXB　※共通ライブラリ側で使用している
・YAML 入出力：JYaml　※共通ライブラリ側で使用している

以下に UML の配置図で表した本ツールの全体アーキテクチャ概要を示す．

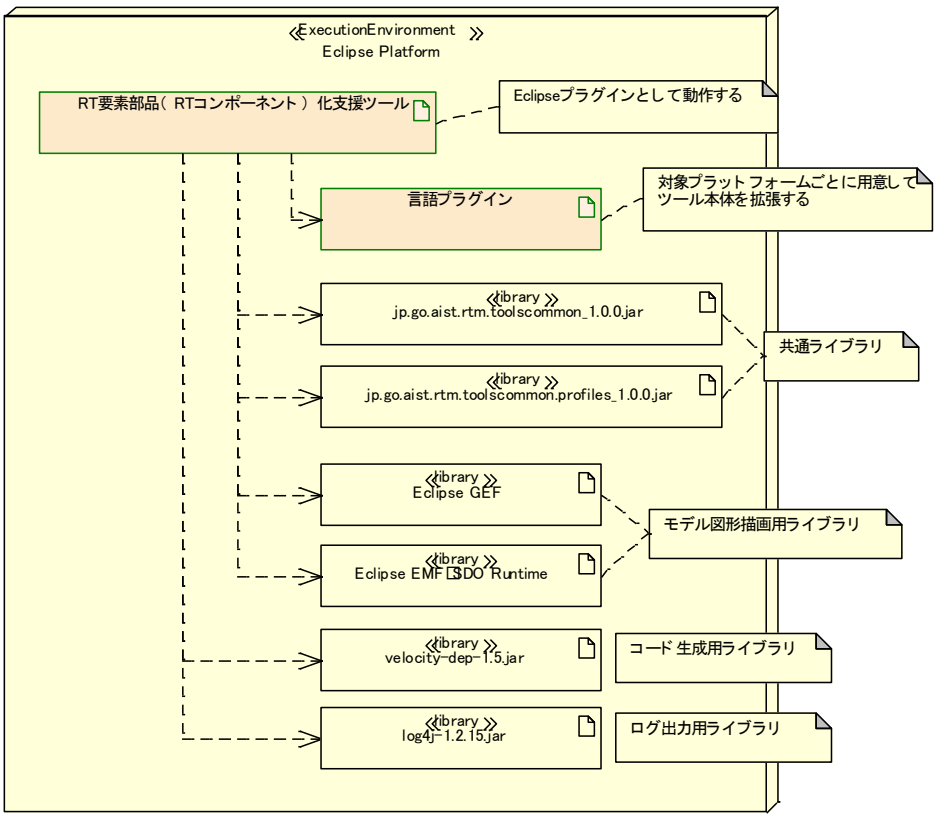


図 a-3-13 RT 要素部品（RT コンポーネント）化支援ツールの全体アーキテクチャ概要

〇外部仕様

以下に，本ツールの基本的な画面構成を示す．

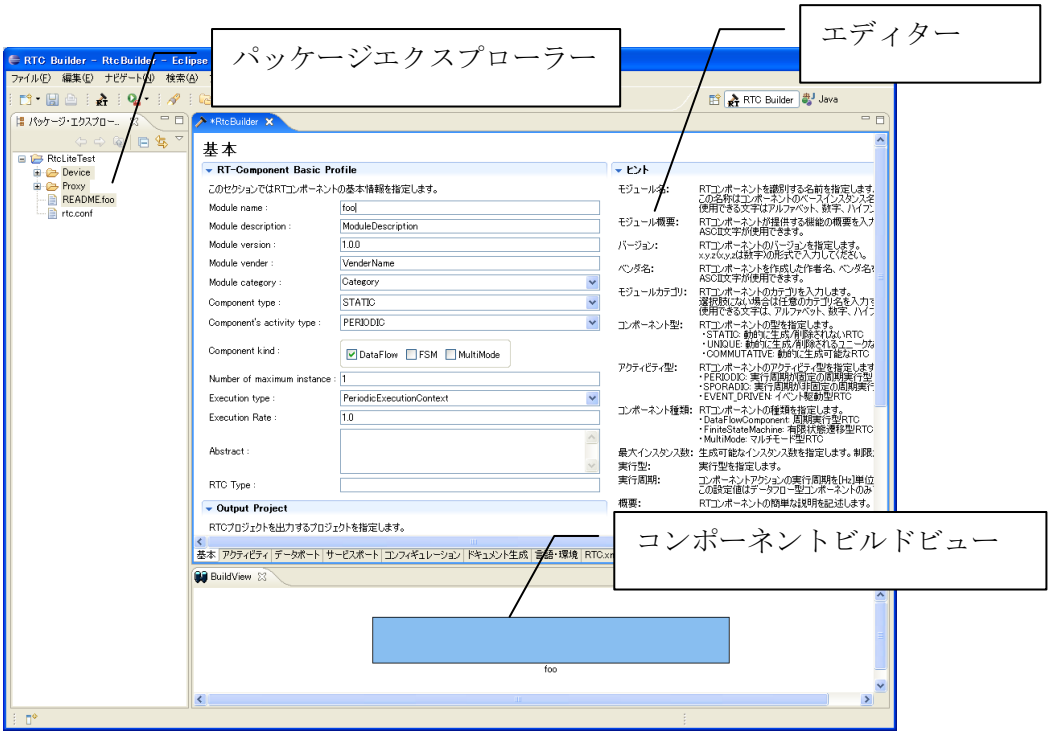


図 a-3-14 RT 要素部品(RT コンポーネント)化支援ツール 画面構成

本ツールの画面は大きく3つのパートに分けられる.「エディター」は RT 要素部品(RT コンポーネント)の仕様情報を編集する箇所である. 編集する情報をカテゴリ別に部類して編集ページとして独立させ, タブによってそれらを自由に切り替えられるようにする. 各編集ページの右側にはヒント情報を表示する.「コンポーネントビルドビュー」は編集中の RT 要素部品(RT コンポーネント)の仕様を図形によって分かりやすく表示する.「パッケージエクスプローラー」には, ソースコード生成を行う際の対象 Eclipse プロジェクトとその構成ファイルをツリー形式で表示する.

各ミドルウェアに対応するために, エディター内にそれぞれのミドルウェア向けの拡張ページを追加する. 以下に miniRTC 用の「ID 関連付けページ」を示す. なお, 初回のエディター起動時には,「ID 関連付け」ページは非活性化されており,「言語・環境設定」ページの言語選択にて「miniRTC」を選択することで, 活性化される.

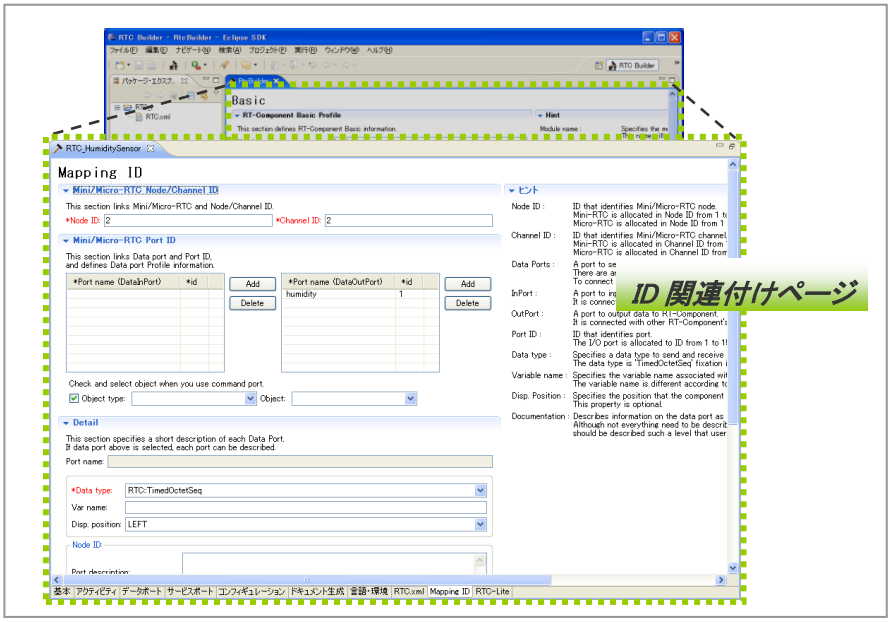


図 a-3-15 miniRTC 向け ID 関連付けページ

○パッケージ構成

本ツールは以下のような役割ごとにパッケージを構成する.

・ユーザインタフェース(ui)
・RT 要素部品(RT コンポーネント)仕様情報の保持と入出力処理(generator)
・ソースコード生成(manager)
・ソースコード生成用テンプレート(template)
・CORBA の IDL パース処理(corba)
・図形表示用 MVC の Model(model)
・共通処理(util)

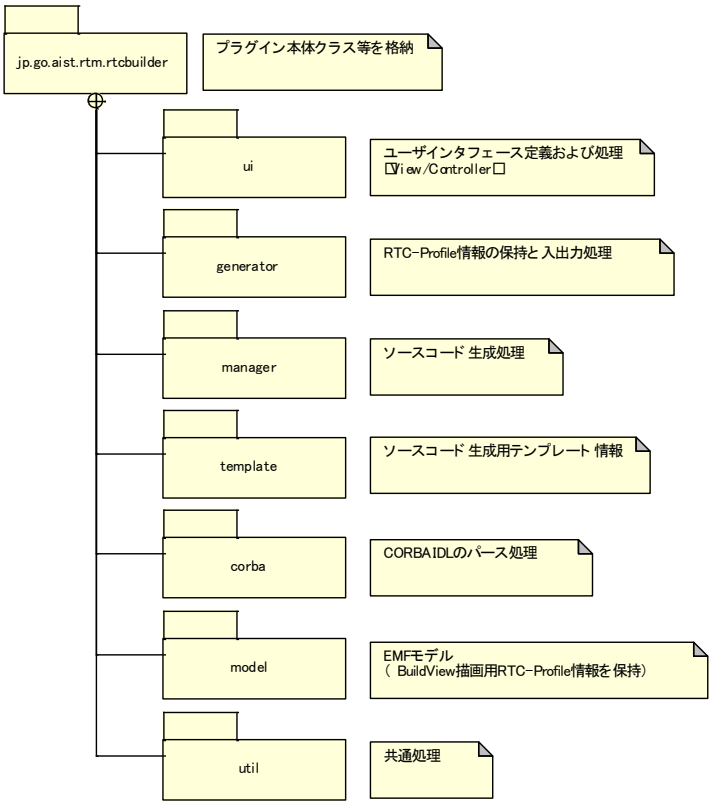


図 a-3-16 RT 要素部品(RT コンポーネント)化支援ツール パッケージ構成

これらパッケージの依存関係を以下に示す. ユーザインタフェースから各種機能を呼び出す形になるため, 以下のようなインポート関係となる.

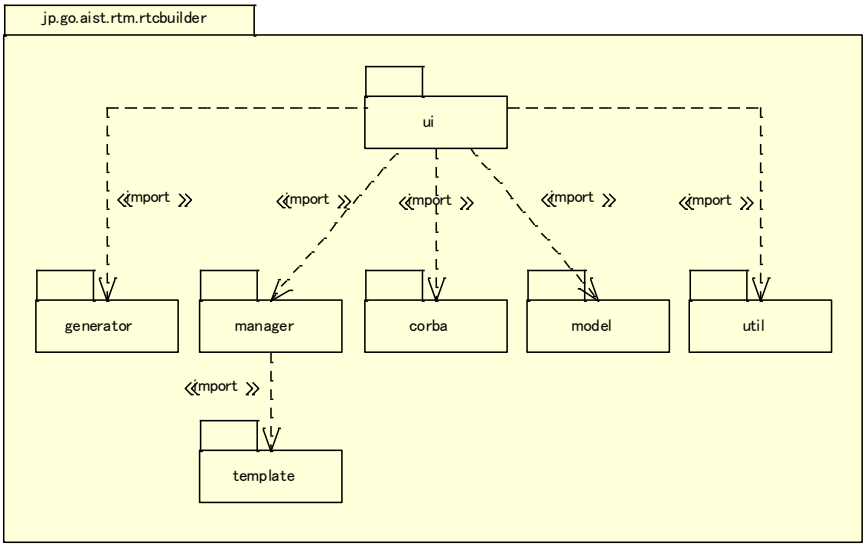


図 a-3-17 RT 要素部品（RT コンポーネント）化支援ツール パッケージ依存関係

パッケージ「ui」は，さらに以下のような役割でパッケージを分類する.
・エディタ／ページ（editors）
・ウィザード（wizard）
・ダイアログ（dialog）
・ソースコードの比較ダイアログ（compare）
・設定ページ（preference）
・ポップアップメニュー（action）
・コンポーネント図形描画用ビュワー（views）
・図形部品（figure）
・図形部品のコントローラ（editpart）
・GUI 部品（parts）
・パースペクティブ連携用クラス（perspective）

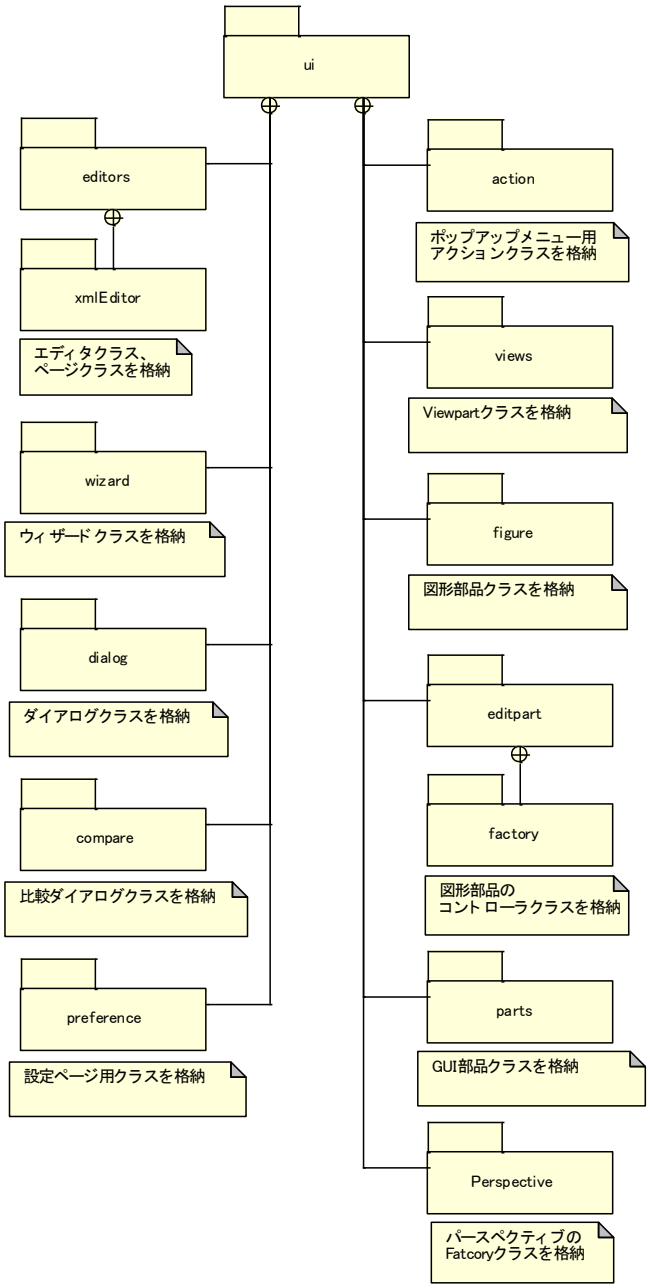


図 a-3-18 RT 要素部品(RT コンポーネント)化支援ツール UI パッケージの構成

パッケージ「generator」は，さらに以下のような役割でパッケージを分類する．
・編集中の RT 要素部品(RT コンポーネント)仕様情報を保持する(param)
・ソースコードをパース・マージする(parser)

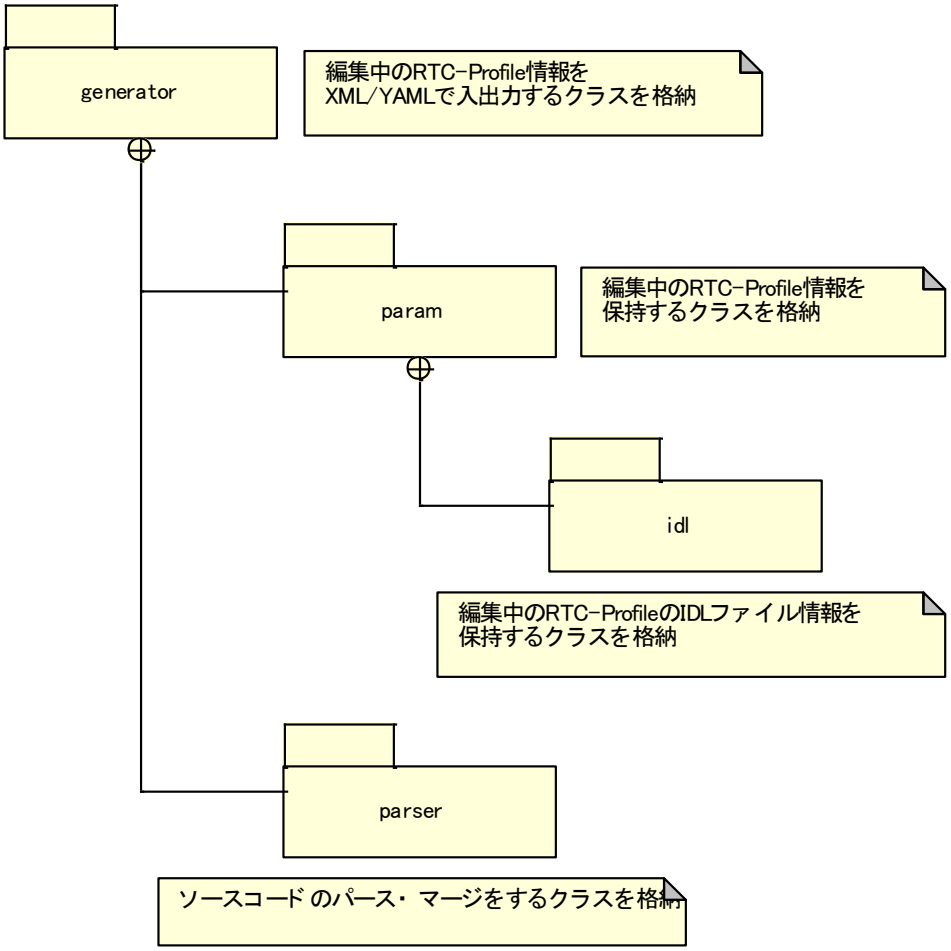


図 a-3-19 RT 要素部品(RT コンポーネント)化支援ツール generator パッケージの構成

パッケージ「template」は，さらに以下のような役割でパッケージを分類する.
・C++関連ソースコードの velocity テンプレート(cpp)
・Visual Studio 関連の velocity テンプレート(cppwin)
・RTM ver0.42 向け C++関連 velocity テンプレート(_042)
・RTM ver1.0 向け C++関連 velocity テンプレート(_100)
・共通 velocity テンプレート(common)

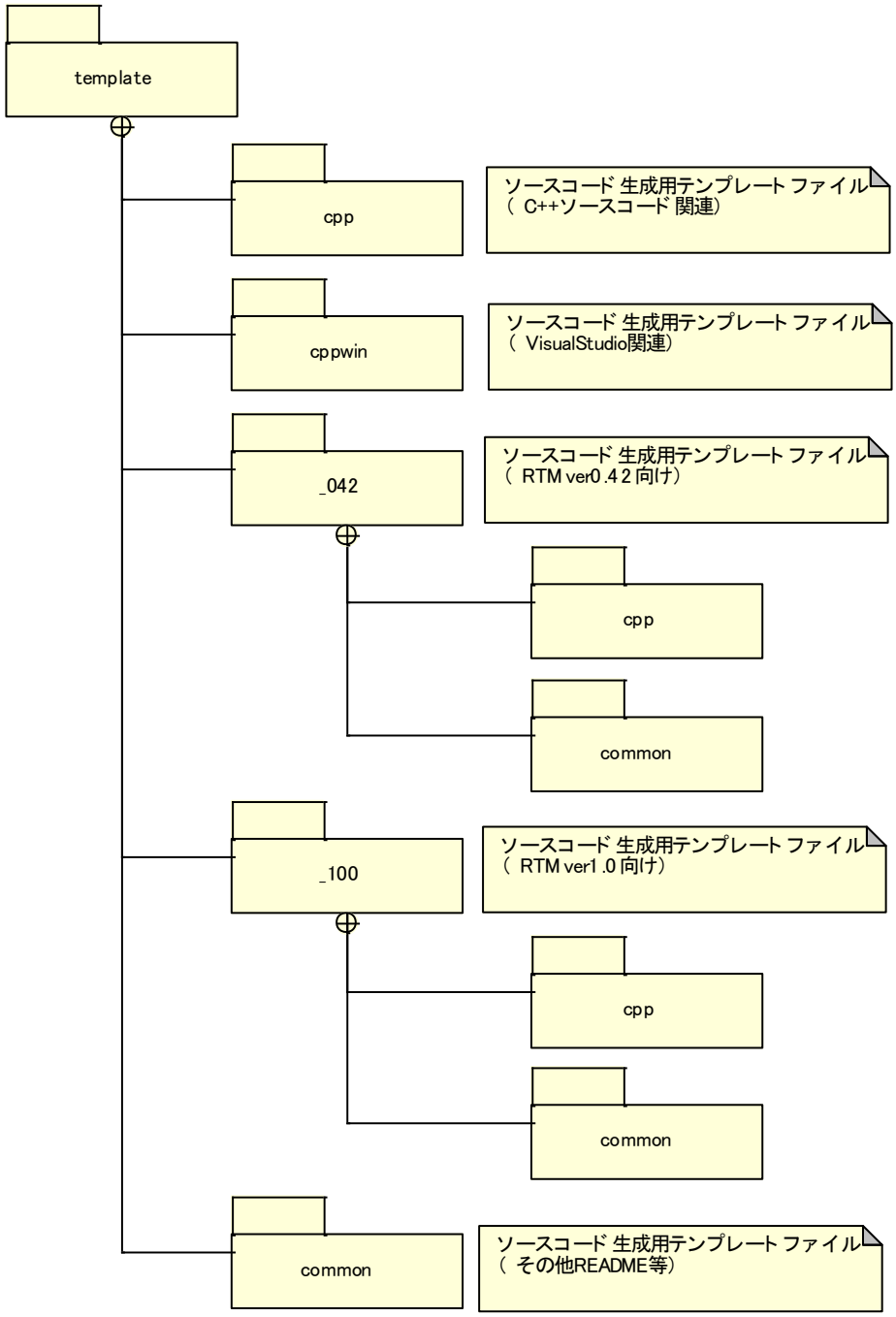


図 a-3-20 RT 要素部品（RT コンポーネント）化支援ツール template パッケージの構成

○RT 要素部品情報編集機能

RT 要素部品情報編集機能を実現するための基本的なクラス構造を示す.

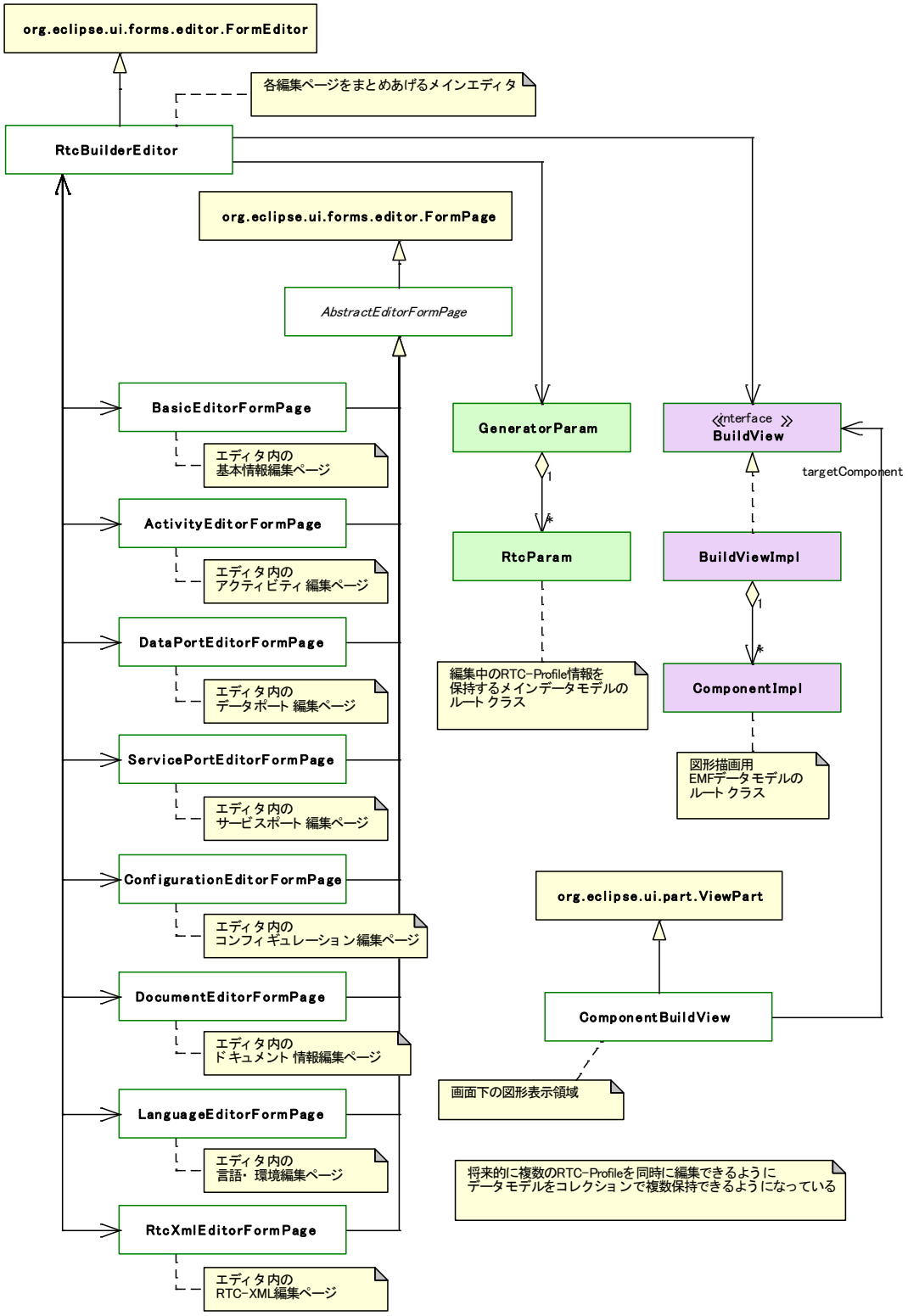


図 a-3-21 RT 要素部品情報編集機能 クラス図

RT 要素部品情報を編集するための各種ページのクラスが1つのエディタクラス（RtcBuilderEditor）によってまとめられる. このエディタクラスは編集内容を2種類のデータモデルに反映する. 1つは編集中のすべての RT 要素部品情報を保持するデータモデル「RtcParam」, もう1つはコンポーネントビルドビューにおける図形描画用モデル

「BuildView」である(このクラスは EMF によって生成されたクラスである).
以下に RT 要素部品情報編集時の処理フローを示す.

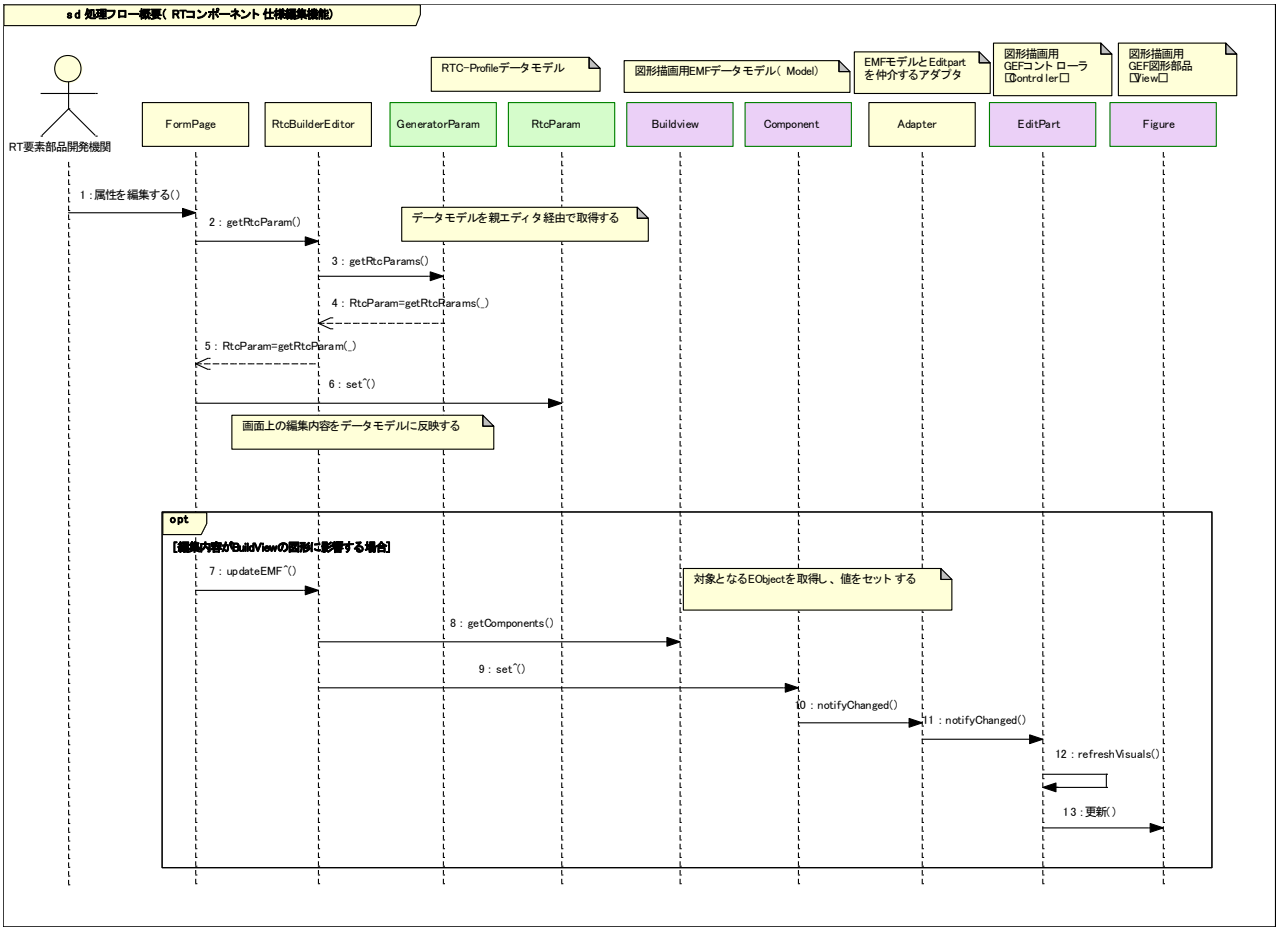


図 a-3-22 RT 要素部品情報編集機能　シーケンス図

ツールのユーザによって各編集ページの情報が変更されると，各編集ページクラスはエディタクラスを経由してデータモデルである「RtcParam」の値を書き換える．また，その値がコンポーネントビルドビューの図形に影響する場合は(コンポーネントの名称やデータポートの編集等を行った場合)，「BuildView」のデータモデルの値を変更し，図形を再描画する．図形の描画は EMF と GEF を連携させて実現している．

〇ソースコード生成機能

ソースコード生成機能を実現するための基本的なクラス構造を示す.

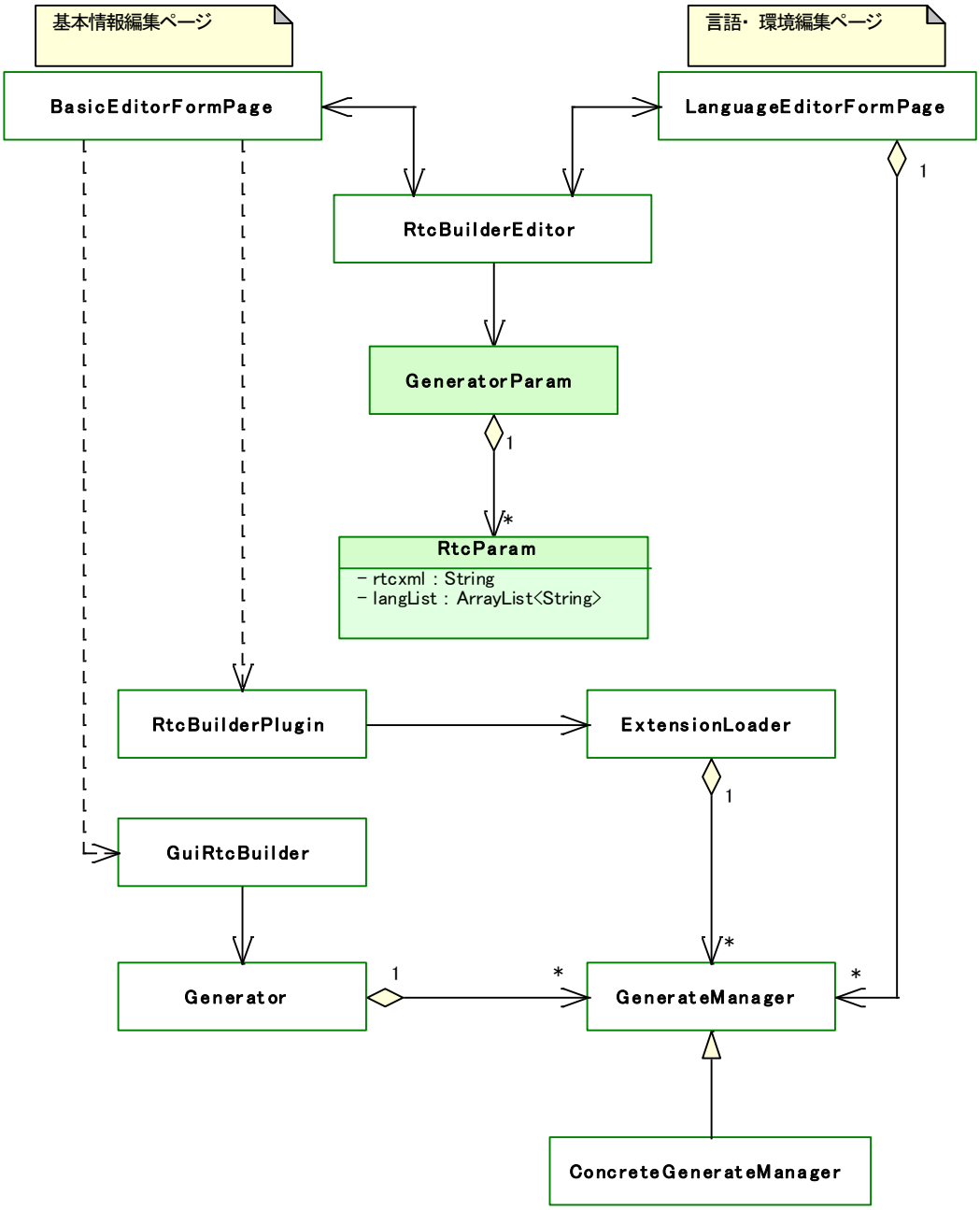


図 a-3-23 ソースコード生成機能 クラス図

ソースコード生成機能は，基本情報編集ページ（BasicEditorFormPage）上の「コード生成」ボタンによって起動する. ソースコード生成の具体的な機能が実装されているのは「GenerateManager」であり，データモデル「RtcParam」を参照しながら処理が実行される（GenerateManager クラスはプラットフォームごとに用意される言語プラグイン側で定義されるクラスである）. また，言語・環境編集ページ（LanguageEditorFormPage）では，あらかじめソースコード生成のターゲットとなるプラットフォーム情報（プログラミング言語，OS，CPU 等）の編集を行っておく.

以下にソースコード生成時の処理フローを示す.

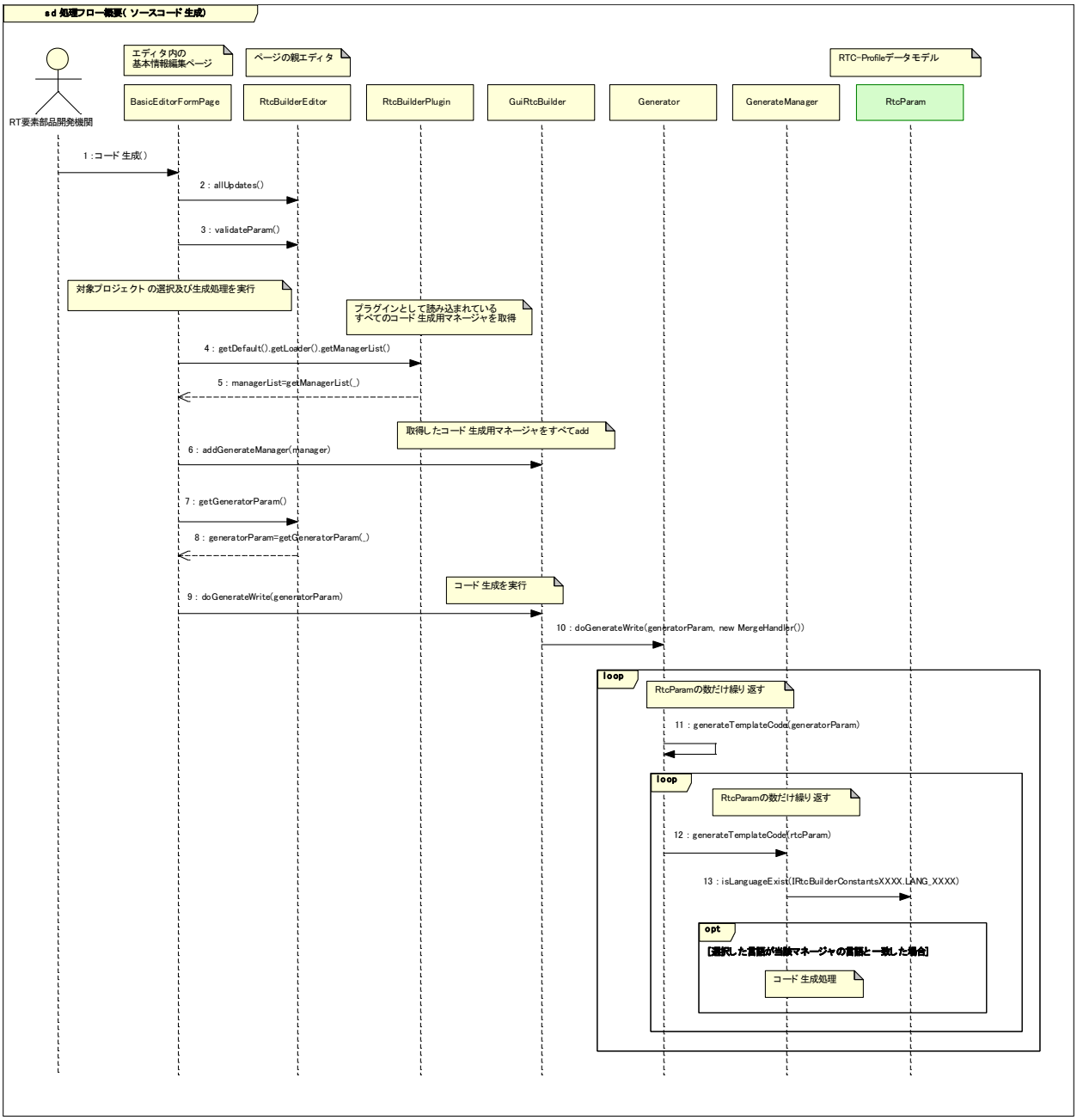


図 a-3-24 ソースコード生成機能　シーケンス図

ツールのユーザによってソースコード生成機能が起動されると，まずツール本体側でソースコードの出力先となる Eclipse プロジェクトの選択および生成処理を行う．次にあらかじめ読み込まれた言語プラグイン側の「GenerateManager」を起動する．各言語プラグインの「GenerateManager」では，言語・環境編集ページ（LanguageEditorFormPage）で選択された言語が自分かをどうか判断し，一致する場合にソースコード生成処理を継続する．

○インポート／エクスポート機能

インポート／エクスポート機能を実現するための基本的なクラス構造を示す．

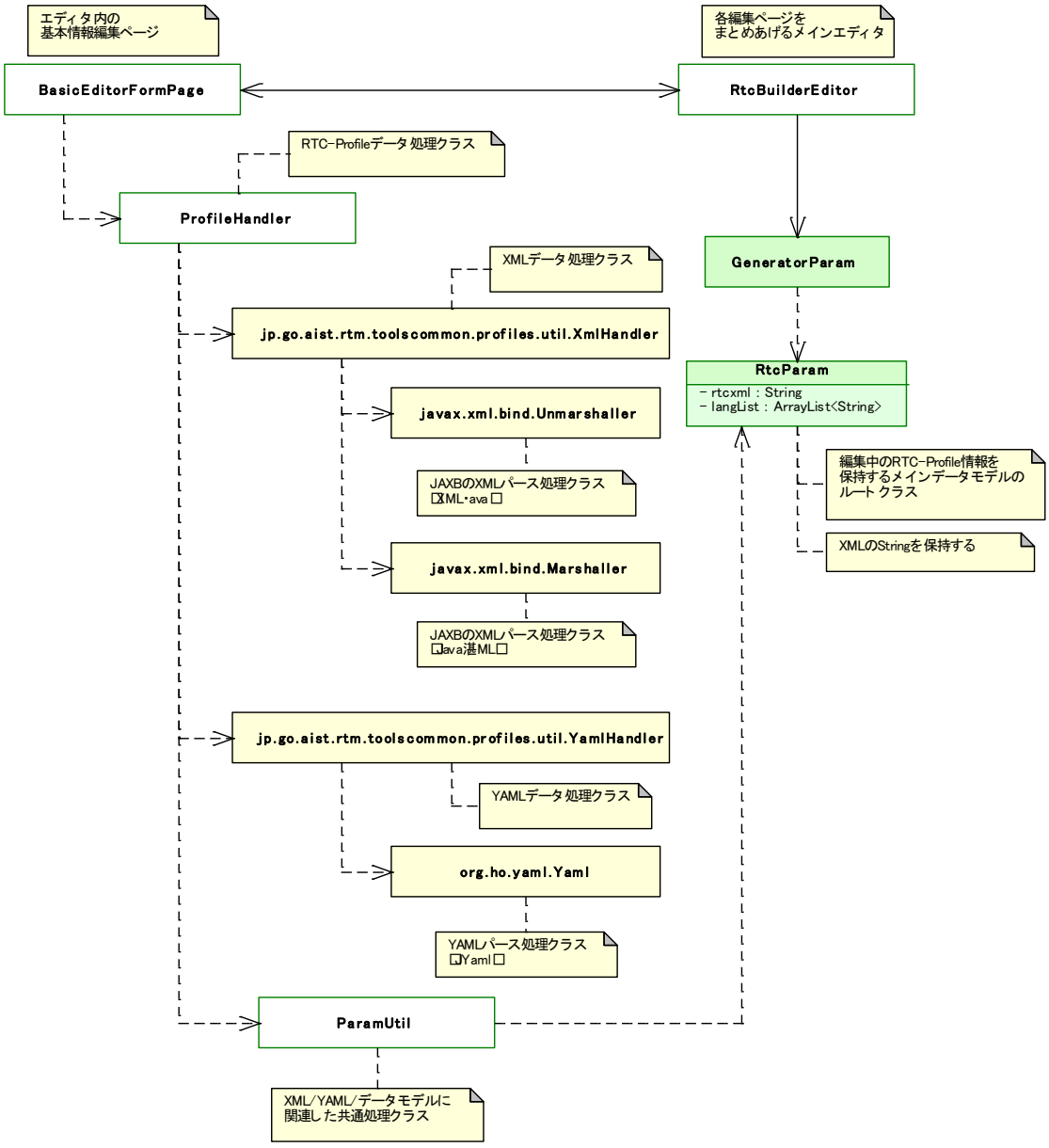


図 a-3-25 インポート／エクスポート機能 クラス図

インポート／エクスポート機能は，基本情報編集ページ（BasicEditorFormPage）上の「インポート」「エクスポート」ボタンによって起動する．インポート／エクスポート機能は「ProfileHandler」によって制御され，XML のインポート／エクスポートは「XmlHandler」に，YAML のインポート／エクスポートは「YamlHandler」に処理を委譲して実行する（この2つのハンドラークラスは共通ライブラリ側に存在する）．XML の String イメージはデータモデル「RtcParam」によって常に保持される．

以下に XML インポート時の処理フローを示す.

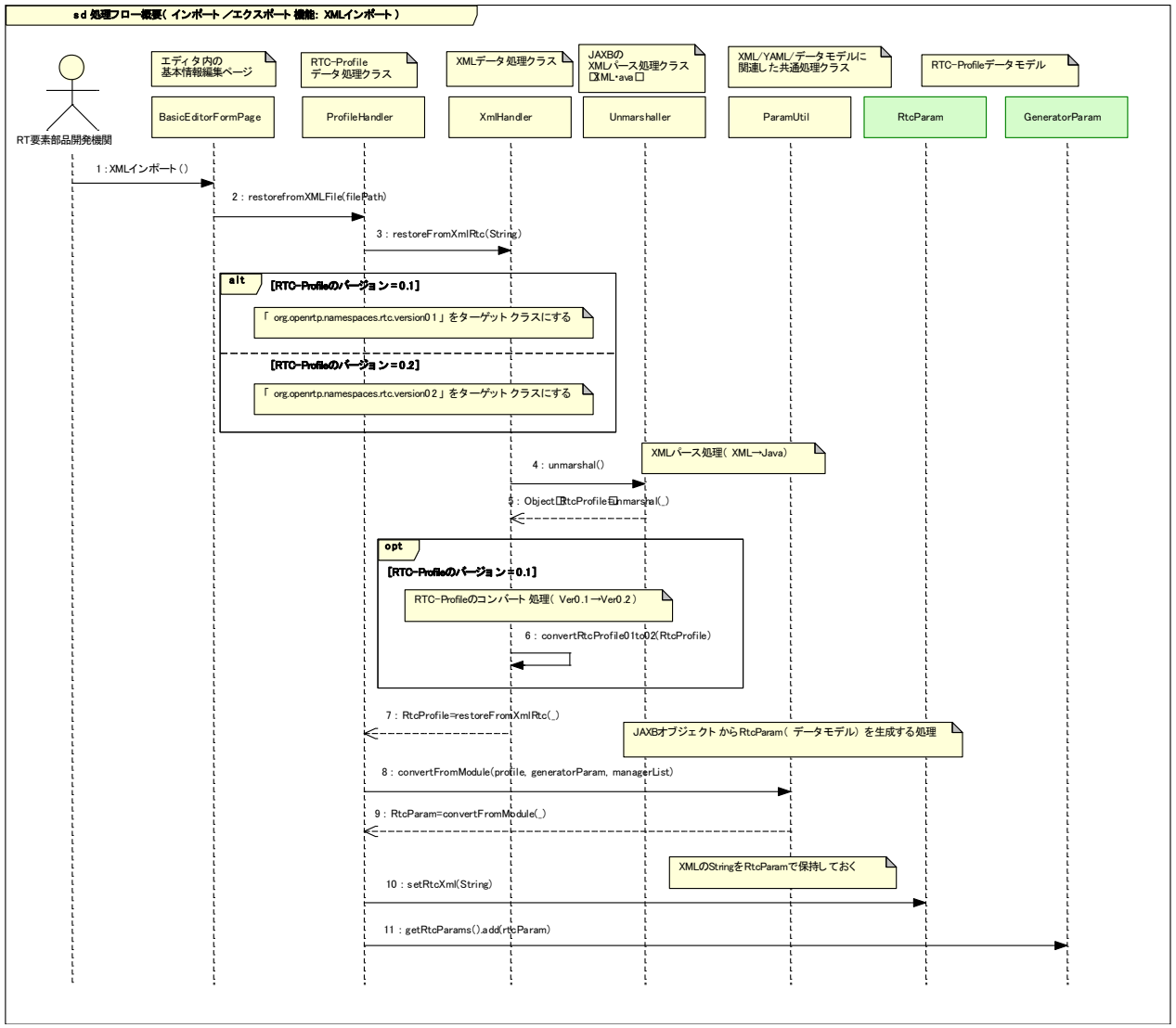


図 a-3-26 インポート／エクスポート機能 シーケンス図(XML インポート)

ツールのユーザによって XML インポート機能が起動されると，まず「ProfileHandler」が XML インポート処理を共通ライブラリ側のクラス「XmlHandler」に委譲する．次に「XmlHandler」は XML ファイルの一部をパースし，RT コンポーネント仕様記述方式のバージョン（ver0.1／ver0.2）を判断する．その後，バージョンに応じた XML ファイル全体のパース処理を実行し，Ver0.1 の場合は ver0.2 の形式にコンバートする処理を実行した上で「ParamUtil」を使用し，JAXB のオブジェクトの内容をデータモデル「RtcParam」に展開する．さらにこのとき，XML の String イメージをデータモデル「RtcParam」にセットしておく．

以下に YAML インポート時の処理フローを示す.

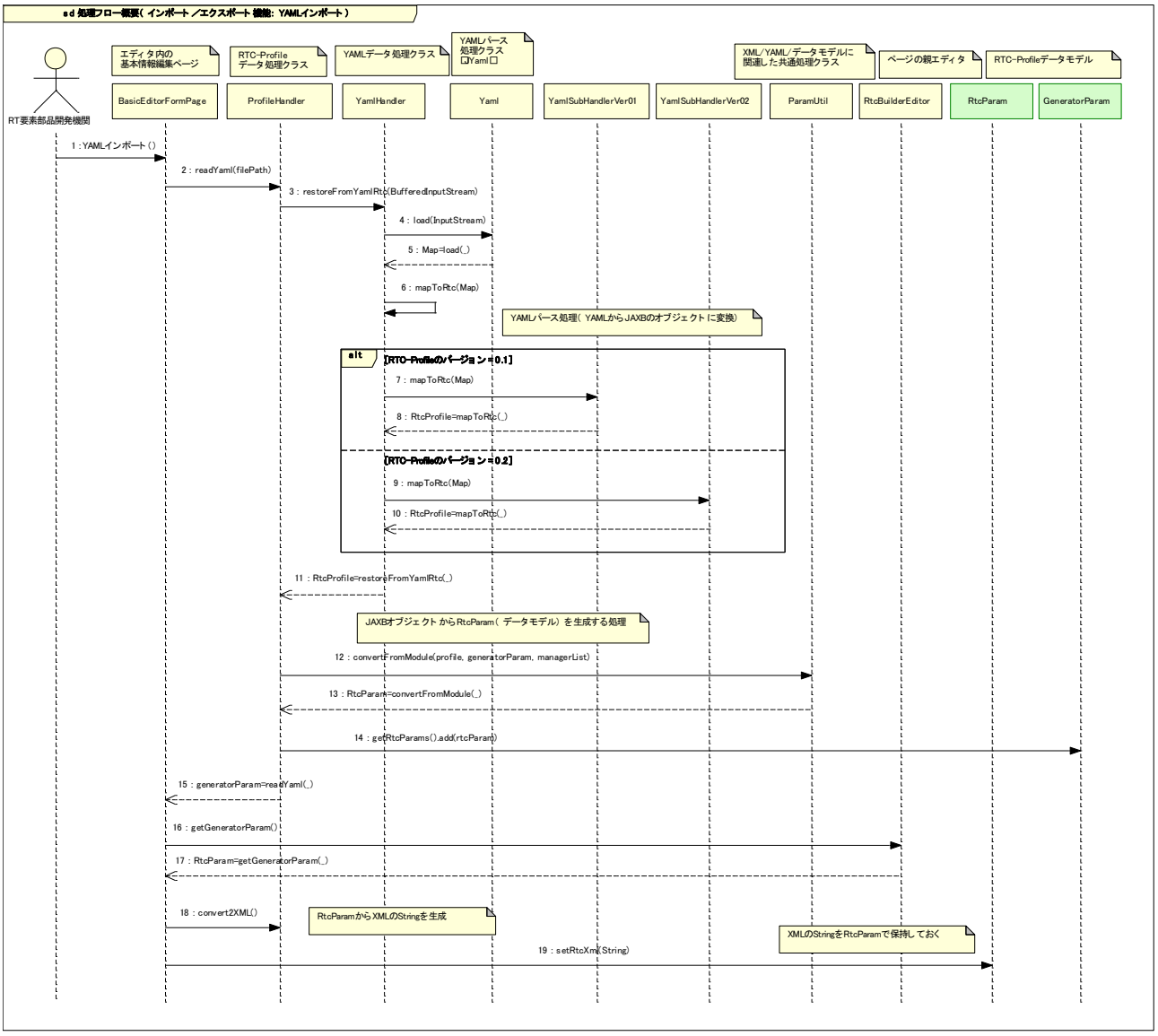


図 a-3-27 インポート／エクスポート機能 シーケンス図(YAML インポート)

ツールのユーザによって YAML インポート機能が起動されると，まず「ProfileHandler」が YAML インポート処理を共通ライブラリ側のクラス「YamlHandler」に委譲する．次に「YamlHandler」は RT コンポーネント仕様記述方式のバージョン（ver0.1／ver0.2）に応じたパース処理を実行し，JAXB のオブジェクトイメージに変換する．その後「ParamUtil」を使用し，JAXB のオブジェクトの内容をデータモデル「RtcParam」に展開し，XML の String イメージを生成し，データモデル「RtcParam」にセットしておく．

以下に XML エクスポート時の処理フローを示す.

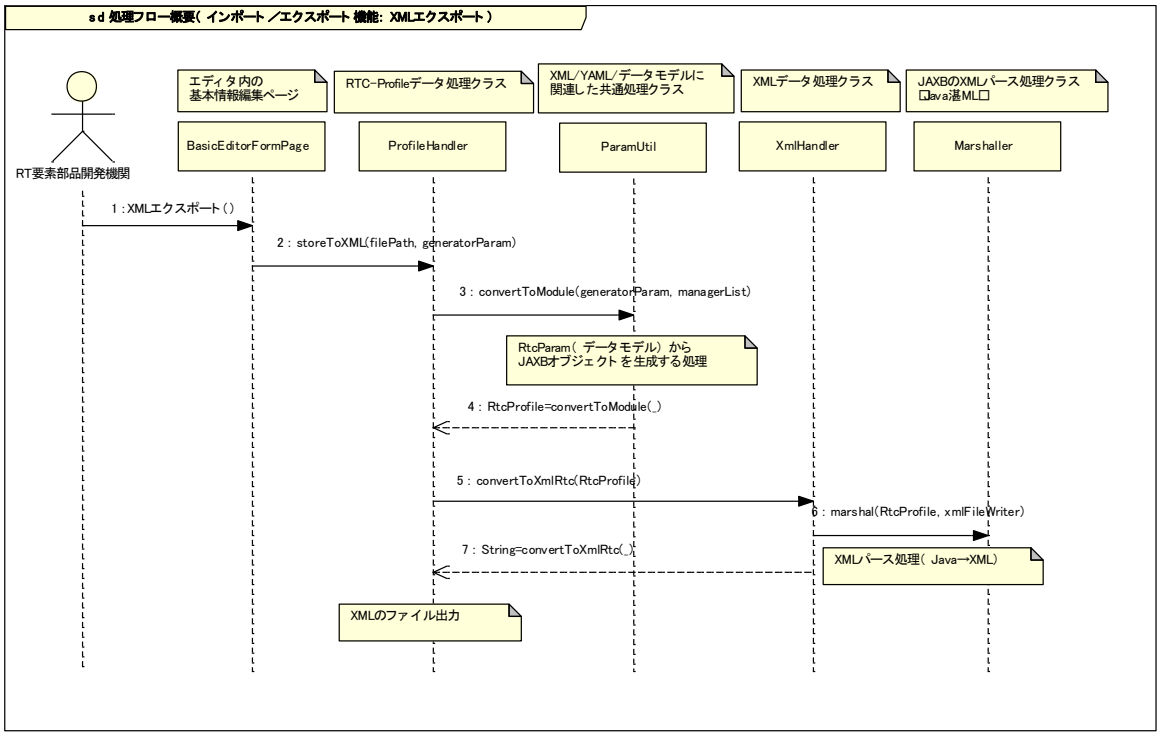


図 a-3-28 インポート／エクスポート機能 シーケンス図(XML エクスポート)

ツールのユーザによって XML エクスポート機能が起動されると，まず「ParamUtil」を使用し，データモデル「RtcParam」の内容を JAXB のオブジェクトイメージに変換する．次に、共通ライブラリ側のクラス「XmlHandler」によって XML ファイルの生成を行う.

以下に YAML エクスポート時の処理フローを示す.

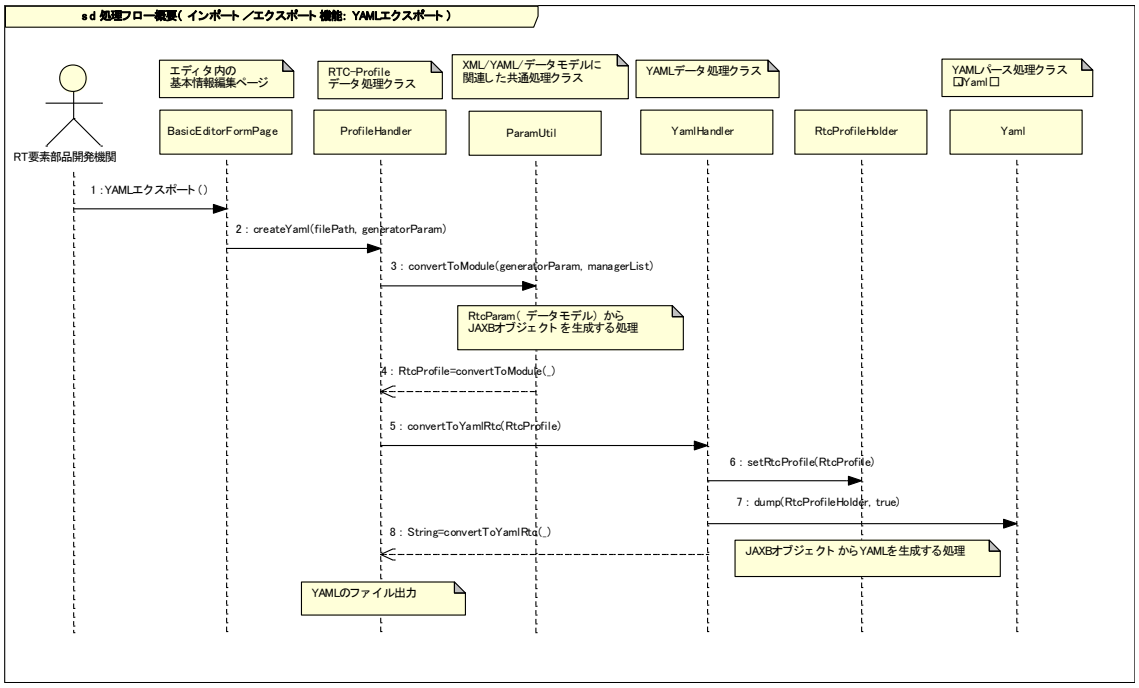


図 a-3-29 インポート／エクスポート機能 シーケンス図(YAML エクスポート)

ツールのユーザによって YAML エクスポート機能が起動されると，まず「ParamUtil」を使用し，データモデル「RtcParam」の内容を JAXB のオブジェクトイメージに変換する．次に，共通ライブラリ側のクラス「YamlHandler」によって YAML ファイルの生成を行う.

a-3-3　RT 要素部品(RT コンポーネント)コンフィギュレーション支援ツールの開発

◎目的

本ツールは，「構成機器メーカー」「設備機器メーカー」の開発者が，構成機器(設備機器)向けアプリケーションを開発する際の各種コンフィギュレーション作業を支援することを目的としている．また，「施工業者」が現地で各種設定，調整，テストを行う際の作業を支援することを目的としている．本プロジェクトの対象である「住宅」など，各種 RT システムを構築する場合，前述のように必要な機器を階層化する場合が多い．この場合，機器メーカーはより汎用的な視点から，自社の機器の用途に応じたコンフィギュレーションを規定する．一方，これらの機器を使用する設備機器メーカーは，より限られたドメインに対応するためのコンフィギュレーションを必要とする．このため本ツールでは，このような視点のギャップ，ドメインのギャップを埋め，組み込み RT コンポーネントのコンフィギュレーション情報の設定支援を行うための機能を実現する．また，一度入力/設定した各種情報を類似デバイスの設定に再利用できる仕組みを用意する．そして，ツール自身の実装形態としては，プラグイン方式を採用し，ツール群全体として各種機能追加，カスタマイズを容易に実行できる構成とする．本ツールを利用することで，「構成(設備)機器メーカー」の開発者は，慣れ親しんだ各ドメイン固有の情報により各種設定を行うことができるようになる．全体システムの中において，本ツールが支援を行う範囲を以下に示す．

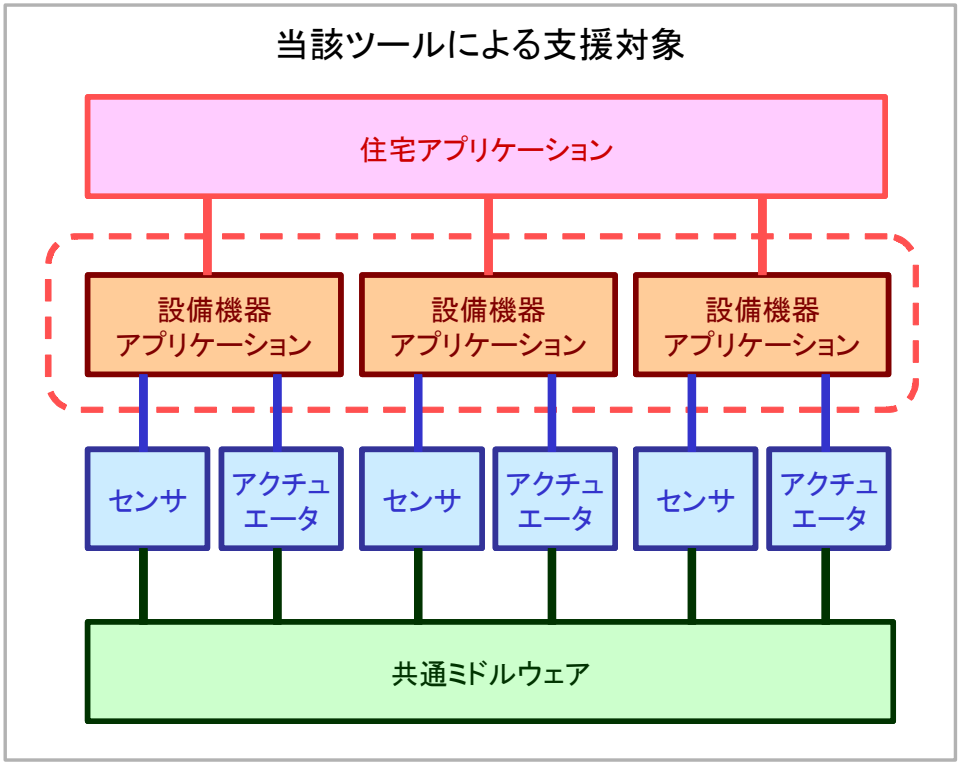


図 a-3-30 RT 要素部品(RT コンポーネント)コンフィギュレーション支援ツールの適用範囲

◎全体概要

本ツールは，RT コンポーネント(組み込み RT コンポーネント含む)にて必要となる各種コンフィギュレーション情報の入力項目の名称や単位変換などを行い，設備機器メーカー視点からの各種項目設定を実現した．また，各種パラメータに対して設定された制限値を用いることで，利用者が誤って不正な値を設定することを防止する．

本ツールは，既存ツールとの操作性の統一，連携の容易化を図るため，独立行政法人　産業技術総合研究所が開発を行っている RTSystemEditor を拡張する形式で Eclipse プラグインとして実現している．このため，必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで，利用者自身が使用する開発環境全体をカスタマイズする事ができるようになっている．

本ツールの画面例を以下に示す.

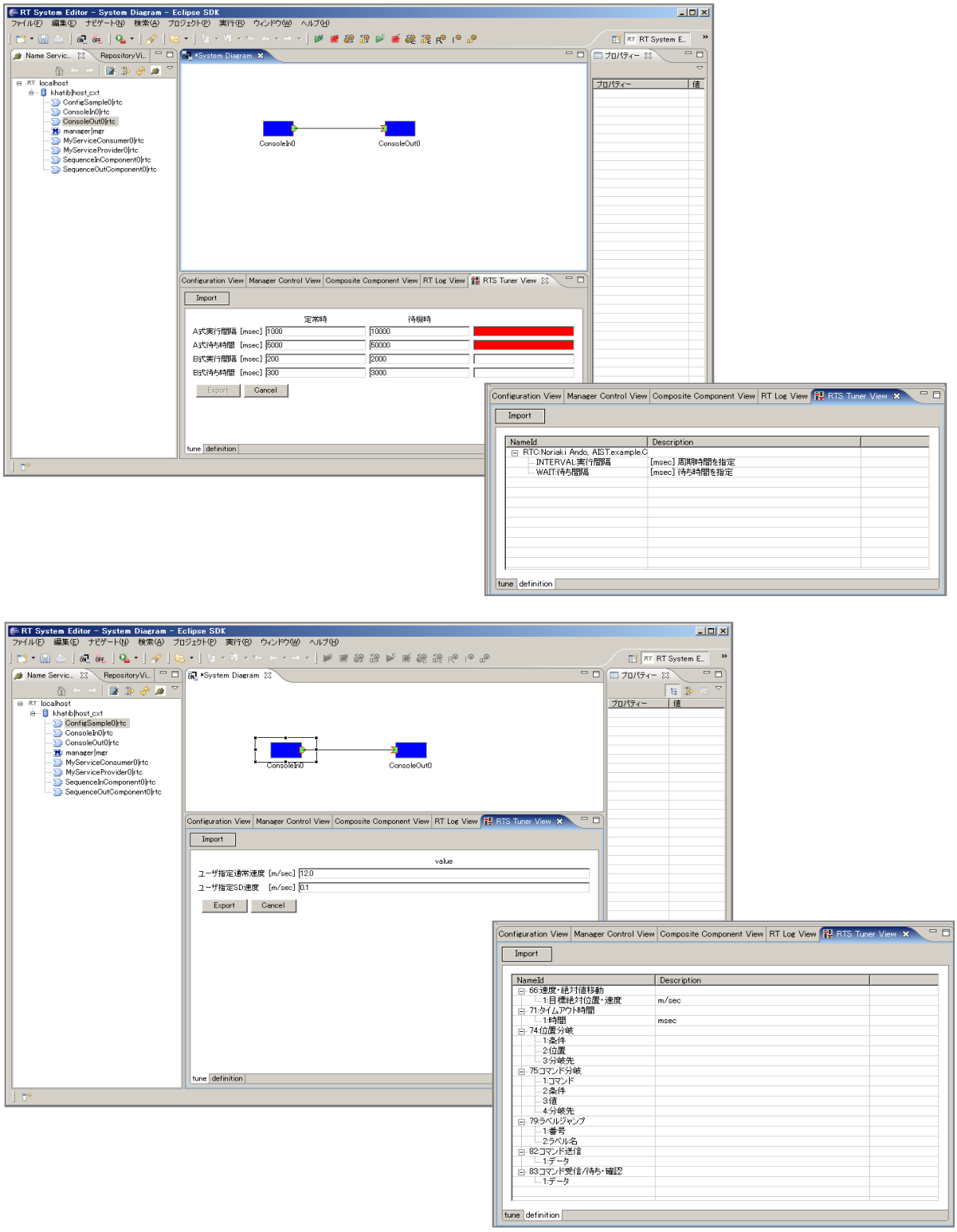


図 a-3-31 RT 要素部品コンフィギュレーション支援ツール

本ツールでは，各種設定に用いる情報をチューニング定義ファイルとして，外部ファイルに保持する形式を採用している．そして，必要に応じてこの情報のみのインポート/エクスポートが行えるようになっている．このため，類似の RT コンポーネントを使用してシステムを構築する場合は，チューニング定義ファイルを再利用することで，各種詳細設定を行う負担を削減でき，開発効率を向上させることができる．

以下に，チューニング定義ファイルの例を示す．

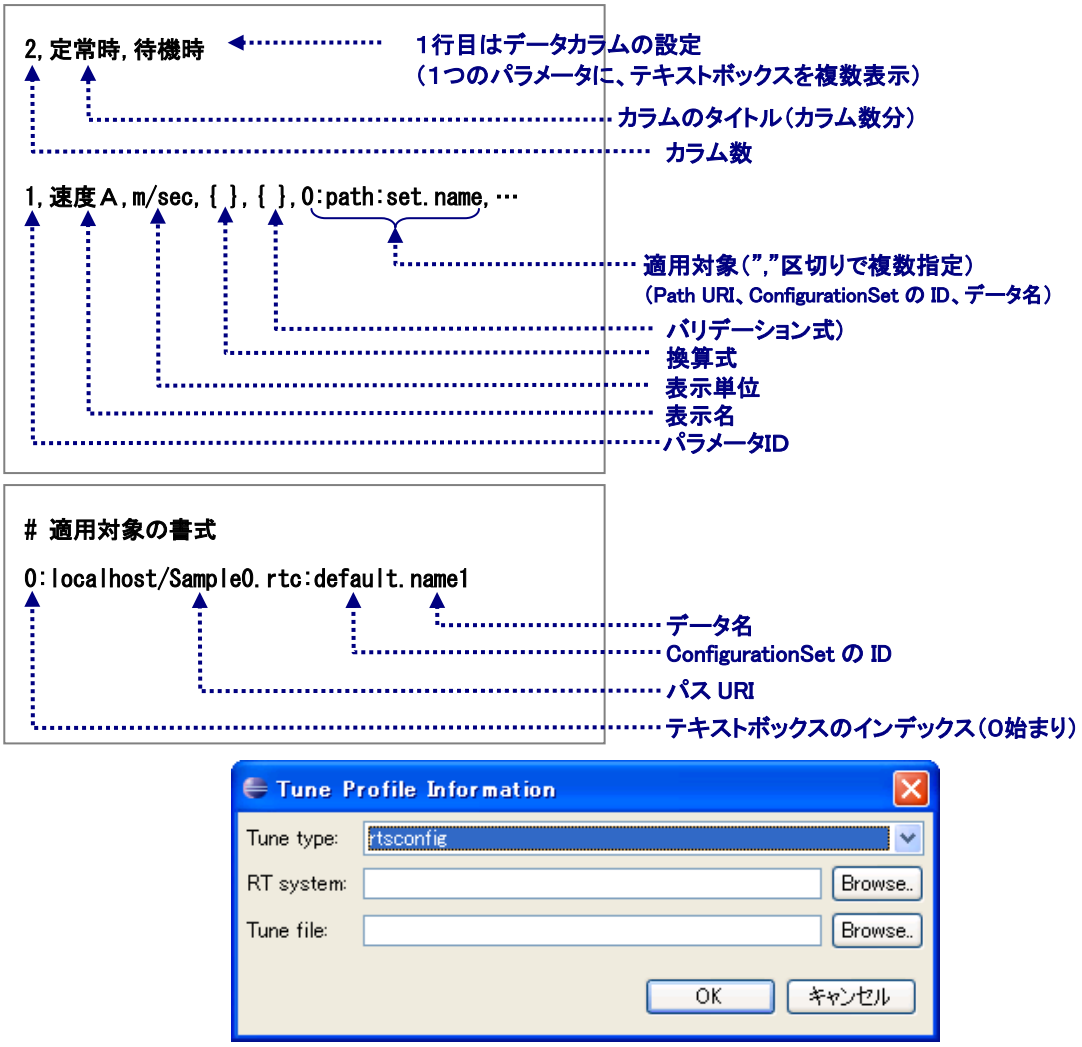


図 a-3-32 チューニング定義ファイルとインポート画面

◎詳細内容

〇機能概要

本ツールは，RT システム(RtsProfile)に含まれる RT コンポーネントのコンフィギュレーションの設定を支援する「RTS コンフィギュレーションのチューニング支援機能」とアクチュエータ制御などのコマンドスクリプトの設定を支援する「コマンドスクリプト向けチューニング機能」を持つ．本ツールの機能概要図を以下に示す．

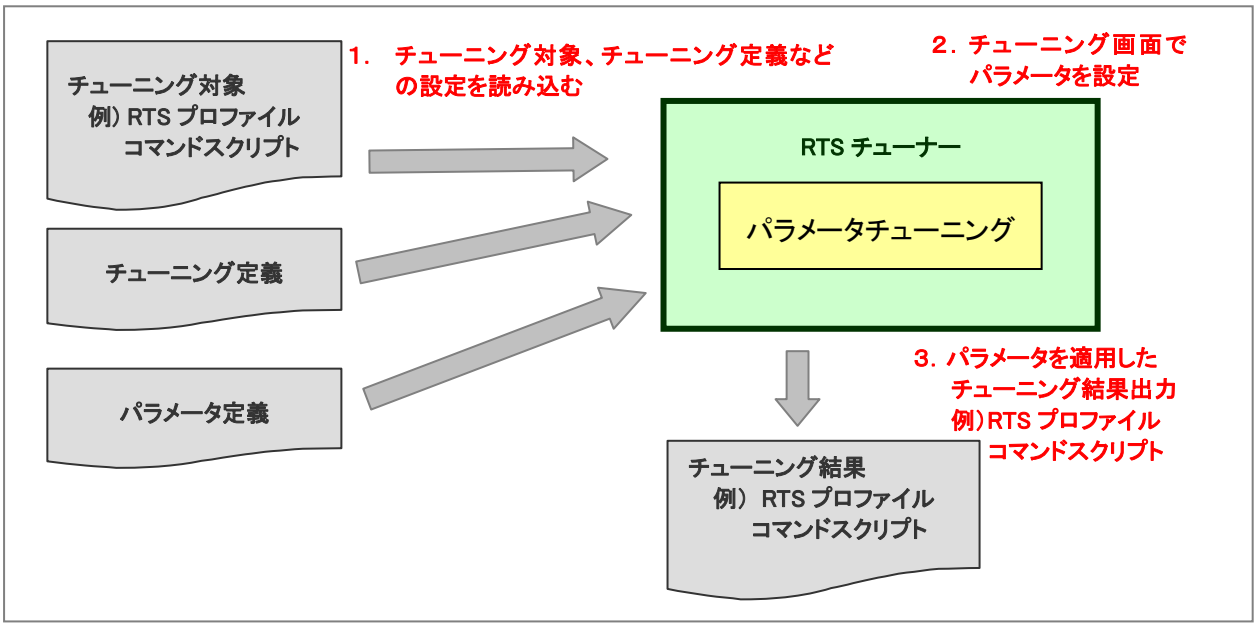


図 a-3-33 RT 要素部品コンフィギュレーション支援ツール　機能概要

「RTS コンフィギュレーションのチューニング支援機能」においては，対象となる RT コンポーネントが持つコンフィギュレーション(ConfigurationSet)のうち，ユーザがチューニング可能な項目のみを抜き出し，画面のフォームにて設定を行えるようにする．

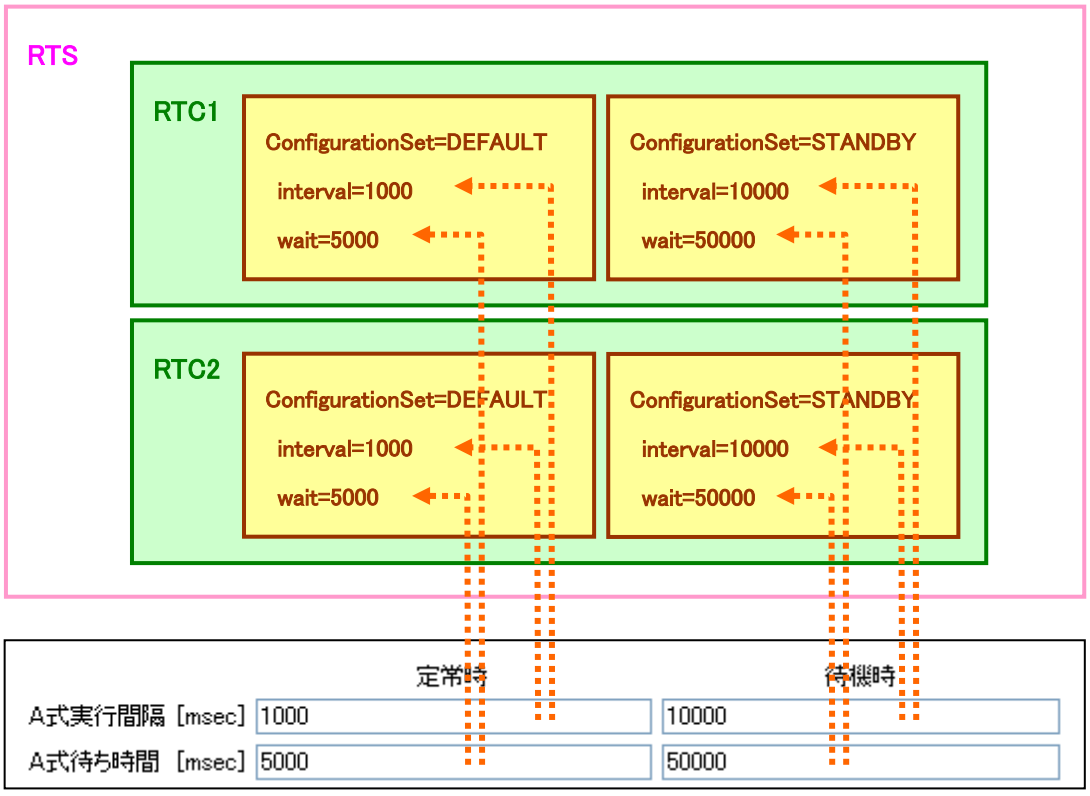


図 a-3-34 チューニング支援機能

「コマンドスクリプト向けチューニング機能」」においては，対象となる CSV 形式のコマンドスクリプトから，ユーザがチューニング可能な項目のみを抜き出し，画面のフォームにて設定を行えるようにする．

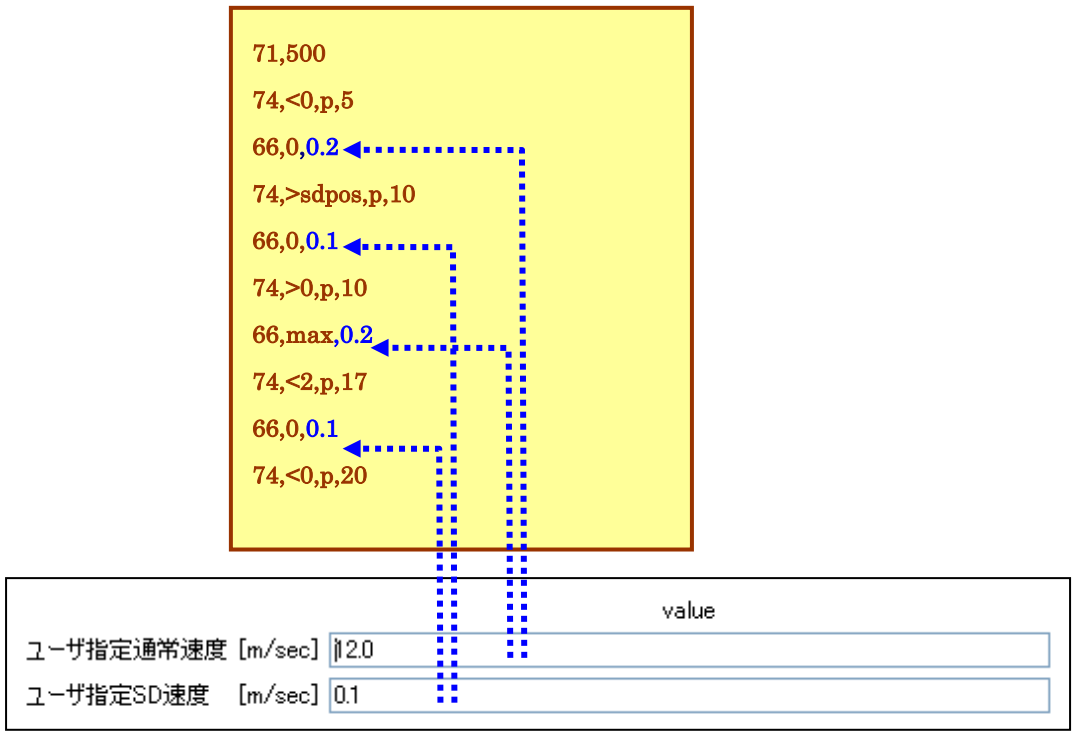


図 a-3-35 コマンドスクリプト向けチューニング機能

○外部仕様

以下に，本ツールの画面構成を示す．

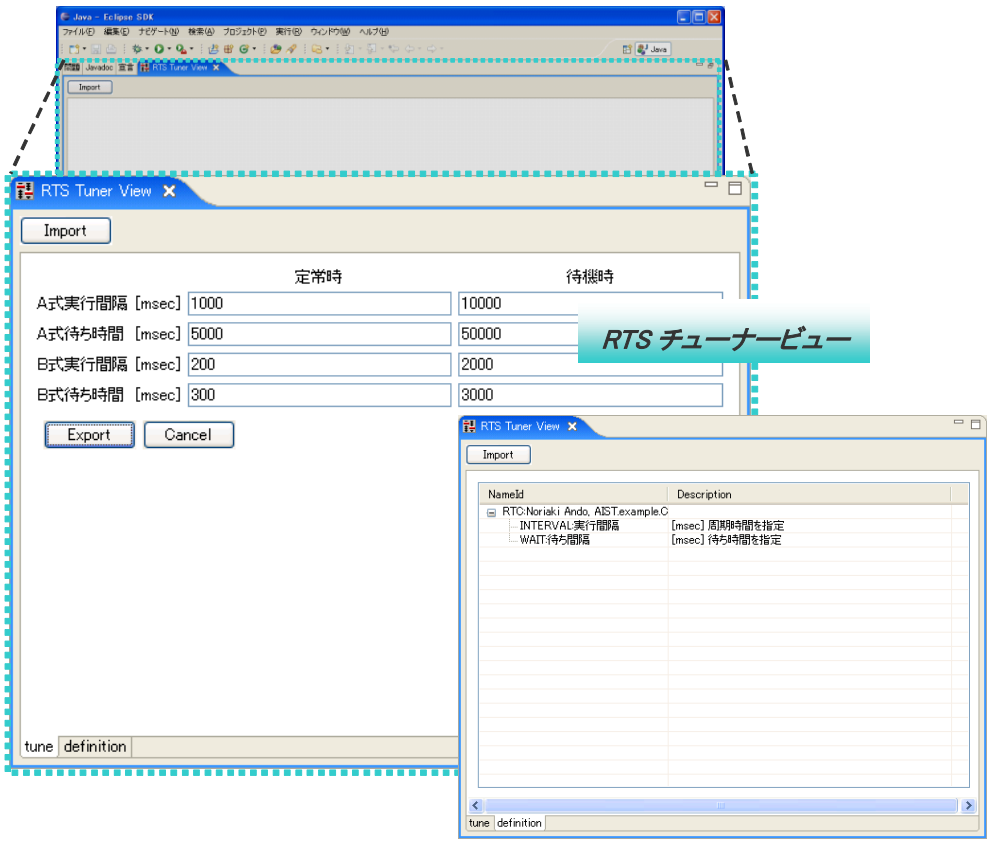
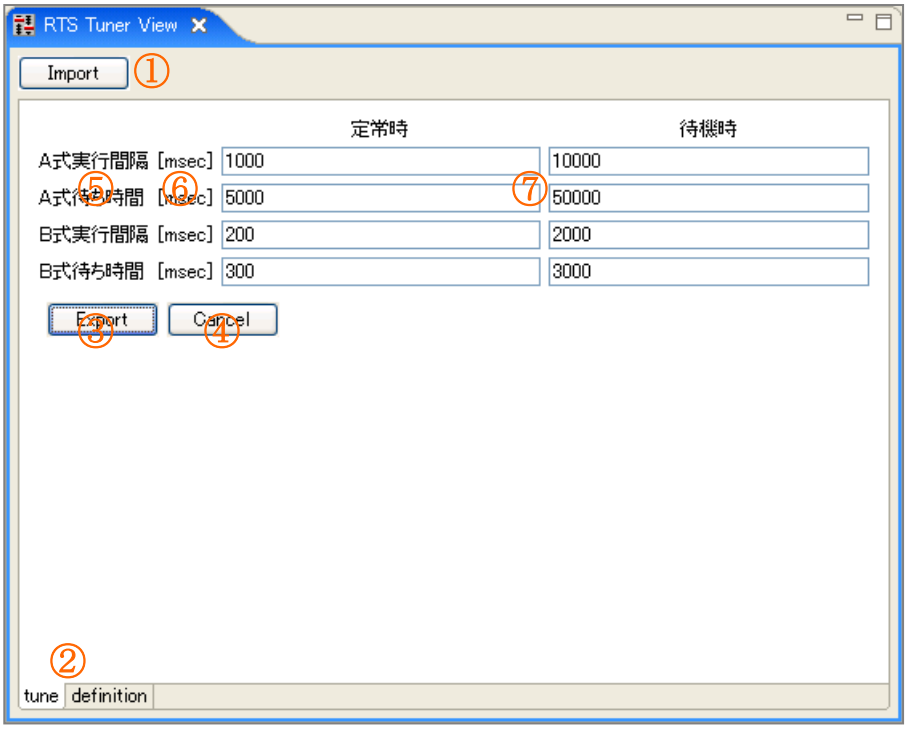
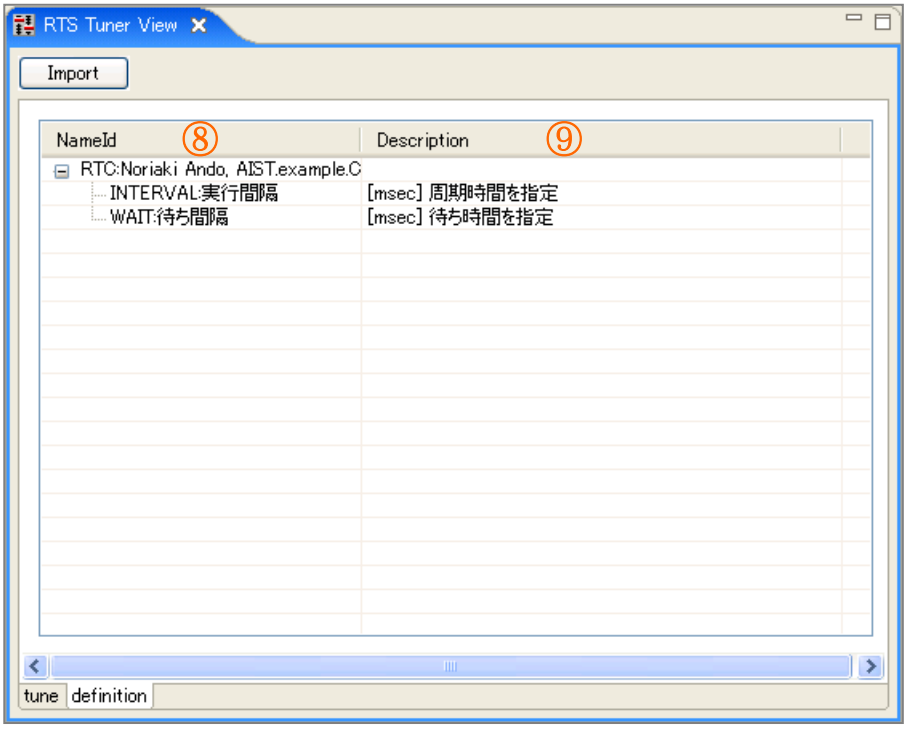


図 a-3-36 RT 要素部品コンフィギュレーション支援ツール　画面構成

本ツールでは，チューニング定義に応じたパラメータの設定画面を表示する．画面内はタブ切り替えによる複数のシートで構成されており，チューニングを行う「tune」シート，コマンドやコンフィグレーション定義を一覧する「definition」シートが存在する．



「tune」シート



「definition」シート

図 a-3-37 RT 要素部品コンフィギュレーション支援ツール　詳細仕様

①はチューニング定義などの各種設定ファイルを読み込むためのインポートボタンであり，②はシートを切り替えるためのタブである．

「tune」シート中の③が指定したターゲットへ設定したパラメータの内容を転送するエクスポートボタンであり，④が変更した内容を元の状態に戻すためのクリアボタンである．⑤は各パラメータの名称，⑥はパラメータの単位をそれぞれ示しており，⑦がパラメータ情報を設定するためのテキストボックスである．なお，設定ファイルを用いることで，各パラメータには複数の値を設定することが可能である．

「definition」シート中の⑨は設定した各パラメータに対する説明である．

a-3-4 RT システム構築支援ツールの開発

○目的

本ツールは，「住宅メーカー」「システムインテグレーター」の RT システム構築担当者が，ドメイン固有のアプリケーションを開発する際の開発負荷や，「構成機器メーカー」「設備機器メーカー」の開発者が，構成機器(設備機器)向けアプリケーションを開発する際の開発負荷を低減することを目的としている．また，「施工業者」が現地で各種設定，調整，テストを行う際の作業を支援することを目的としている．本ツールを利用することで，各 RT コンポーネントが実際に使用しているプロトコルの違いを意識することなく，通常の RT コンポーネントのみを用いている場合と類似の操作で RT システムの構築を行うことが可能となる．更に，既存のシステムに対して，利用者が容易に機器の追加，拡張を行うためのプラグアンドプレイ機能を支援する．このため，プラグアンドプレイ機能を実現する際に必要な各種情報の設定を行うとともに，設定したプラグアンドプレイ情報に則って，各 RT コンポーネントの実際の制御を行う．また，部分システムの構築をサポートするために，オフライン状態(実際の RT コンポーネントは存在せず，コンポーネントの仕様のみから RT システムを構築した状態)にて構築した RT システムから実際に動作するシステムを構築・起動を行う．

全体システムの中において，本ツールが支援を行う範囲を以下に示す．

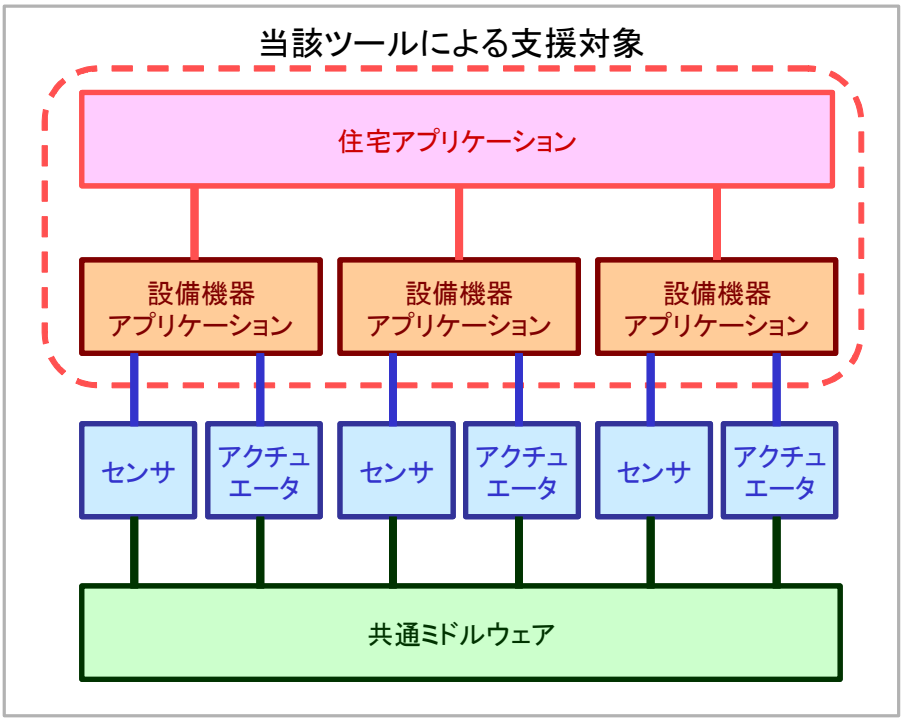


図 a-3-38 RT システム構築支援ツールの適用範囲

○全体概要

・RTCHub/miniRTC/microRTC 用 RTSystemEditor

今回開発したシステムでは，システムの可用性・信頼性を高めるため，コントローラが複数箇所に分散配置されている．そして，高速制御用 RTC-Lite(miniRTC)，低速制御用 RTC-Lite(microRTC)については，実際の制御/センサ用基盤の上で各コンポーネントが動作しているが，これらのデバイスを設備機器アプリケーション単位で管理するためのモジュール(RTC-Lite Manager)が，基盤通信モジュール上で動作している．本ツールではこれらの仕組みを隠蔽し，通常の RT コンポーネントと同様に各種デバイス向けコンポーネントを操作するための機能を実現した．そして，各デバイス単位での操作，情報取得を実現するとともに，デバイス間のポートの接続/切断を実行する機能を実現した．

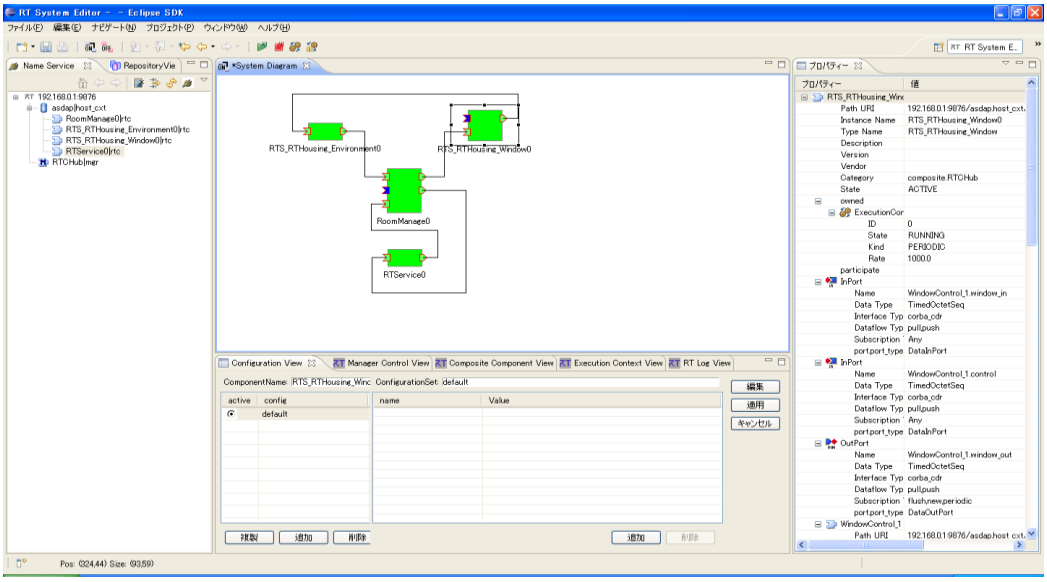


図 a-3-39 minRTC/microRTC 用 RTSystemEditor

一方，設備機器メーカーが提供する各種機器では，ドメインに応じた機能を実現するため，miniRTC と microRT，また通常の RT コンポーネントは区別されることなく，混在した形で使用される．このため，今回開発を行ったシステムでは，要素部品管理モジュール上の RTCHub が，これら機器単位で管理，制御するため、RTC-Lite Manager 毎に疑似複合コンポーネントを作る形で動作している．本ツールでは，RTCHub を構築する際に必要となるプロファイルテーブル情報を生成する機能を開発した．これらの機能を利用することで，システムインテグレーターやある程度のスキルを持ったエンドユーザーが，通常の RT コンポーネントのみを用いて RT システムを構築する場合と類似の操作で，組み込み RT コンポーネントを含んだ RT システムを容易に構築することが可能となる．

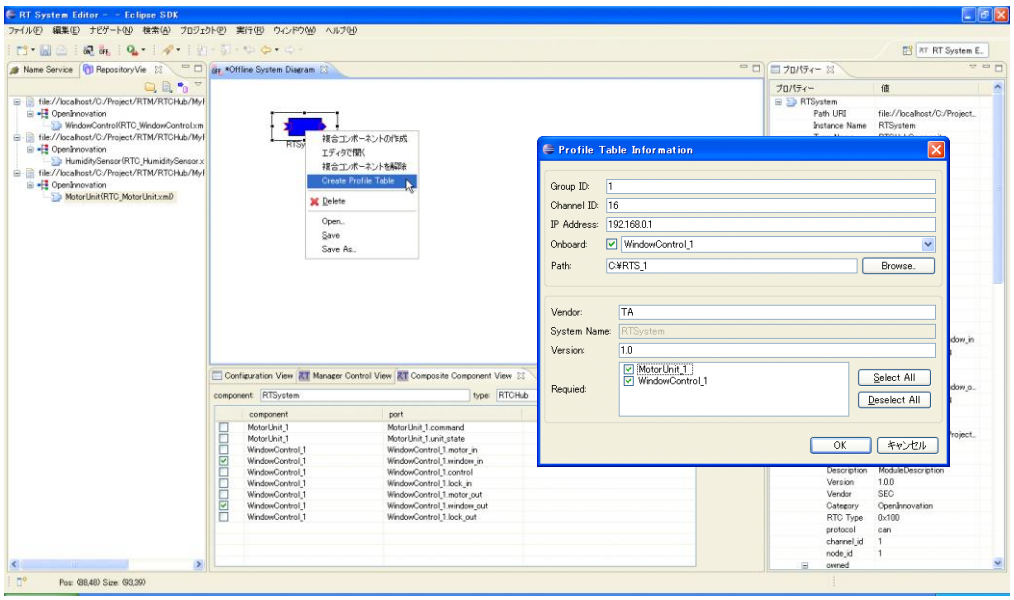


図 a-3-40 RTCHub 用 RTSystemEditor

なお，他の RT ミドルウェア向けツールとの操作性の統一，連携の容易化を図るため，本ツールは独立行政法人産業技術総合研究所が開発を行っているRTSystemEditorを拡張する形で，Eclipseプラグインとして実現しており，通常のサブシステムを構築する場合と同様な手順で，機器の構成を定義することが可能である．また，本ツールを用いて設定を行った内容は，独立行政法人　産業技術総合研究所が規定している「RT　システム仕様記述方式(RTSProfile)」を利用して保存/読み込みが可能である．

・プラグアンドプレイ設定ツール

本ツールは，プラグアンドプレイ機能を実現するために必要となる各種情報を RT システム構築時に設定するとともに，設定したプラグアンドプレイ情報を用いて，実際の各 RT コンポーネントを制御する機能を実現している．プラグアンドプレイ機能を実現するために，対象システムを構成する各コンポーネントのアクション実行順序，アクション実行条件を設定することができるようになっている．また，設定したプラグアンドプレイ情報に基づいて，指定された実行順序でアクションを実行するとともに，制約条件が成立していない場合のアクションの実行待機などを制御する．

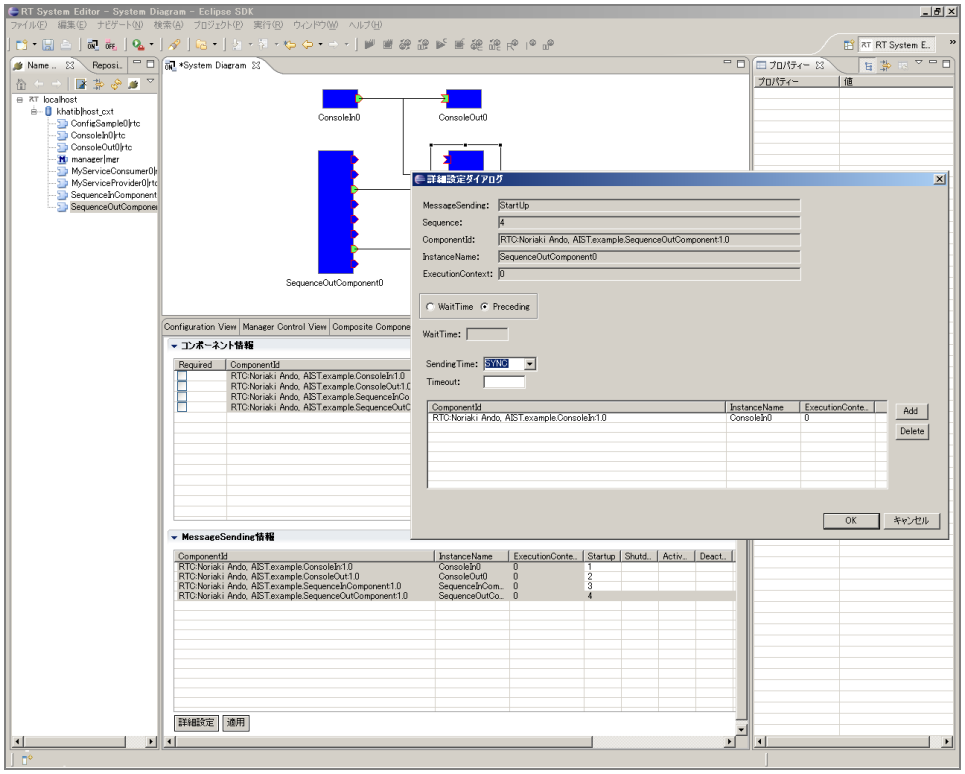


図 a-3-41　プラグアンドプレイ設定ツール

本ツールも他の　RT　ミドルウェア向けツールとの操作性の統一，連携の容易化を図るため，独立行政法人　産業技術総合研究所が開発を行っている RTSystemEditor を拡張する形で，Eclipse プラグインとして実現している．

・RT システムローダー

本ツールは，対象 RT コンポーネントが実際には存在せず，RT コンポーネントの仕様(RtcProfile)だけのオフライン状態で構築した RT システム情報から，実際に動作するシステムを起動するためのツールである．

RT システムの開発では，各要素の開発を並行して行う場合が多いため，各フェーズ初期段階では全ての要素部品が揃わないことが多いと思われる．このような場合，使用する RT コンポーネントの仕様(RtcProfile)から，オフライン状態で対象となる RT システムを仮想的に構築することにより，必要となる RT コンポーネントの整合性，RT コンポーネント間の接続関係を確認することができる．本ツールは，このようにオフライン状態で構築した　RT　システムの情報から，実際のシステムを起動するためのツールである．また本ツールでは，実際のシステムで使用する構成要素を決定する際に，既に動作中の RTC 群から ID 情報をキーとして，マッピング候補 RTC を検索し，該当システムの構成要素として使用する方法と，OpenRTM-aist の RT コンポーネント管理機能(RTManager)を利用して新しい

RT コンポーネントのインスタンスを生成し，該当システムの構成要素として使用する方法を用意している．
　また，本ツールも他の RT ミドルウェア向けツールとの操作性の統一，連携の容易化を図るため，独立行政法人産業技術総合研究所が開発を行っている RTSystemEditor を拡張する形で，Eclipse プラグインとして実現している．

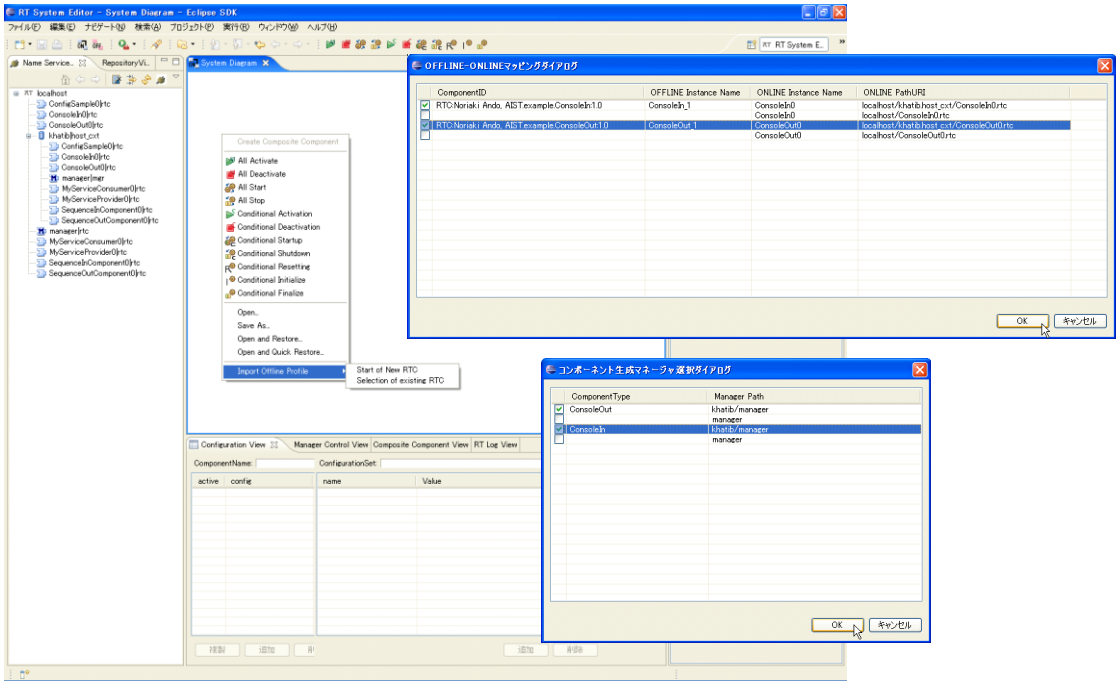


図 a-3-42 RT システムローダー

◎詳細内容
・RTCHub/miniRTC/microRTC 用 RTSystemEditor
〇機能概要
　本ツールは要素部品管理モジュール上で動作する RTCHub を構築する機能と，各組み込み MPU 上で動作する miniRTC/microRTC を管理・制御する機能を持つ．
　RTCHub を構築するためには，予め RTCBuilder により，使用する RTC-Lite(miniRTC，microRTC)を設計し，RTC プロファイルを作成する必要がある．そして，設計した RTC プロファイルをリポジトリビューに登録し，オフラインエディタで RTC-Lite をまとめた複合 RTC(RTCHub 複合 RTC)を作成する．その後，RTCHub 複合 RTC の設計情報(RTS プロファイル)を基に，RTCHub，RTC-LiteManager を構築する．RTC-LiteManager はひとつの RTCHub 複合 RTC を構成し，配下のデバイスをカプセル化するとともに，RTC-LiteManager を RTCHub 管理下に置くため，それぞれの RTCHub 複合 RTC の RTS プロファイルを RTCHub に登録する．

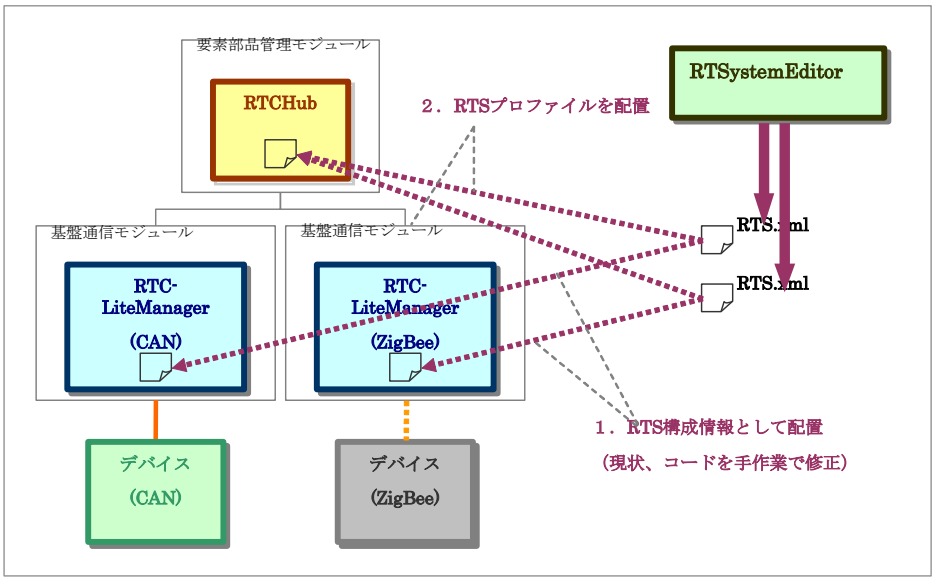


図 a-3-43 RTCHub 構築機能

通常，miniRTC/microRTC は，RTC-LiteManager および RTCHub の配下で管理されるため，通常の RT コンポーネントのようにネームサービス上などで直接参照することはできず，ポート間の接続や，状態の変更などの制御も直接は行うことができない．しかし本ツールでは，RTCHub 複合 RTC を介して miniRTC/microRTC を参照，制御する仕組みを用意している．

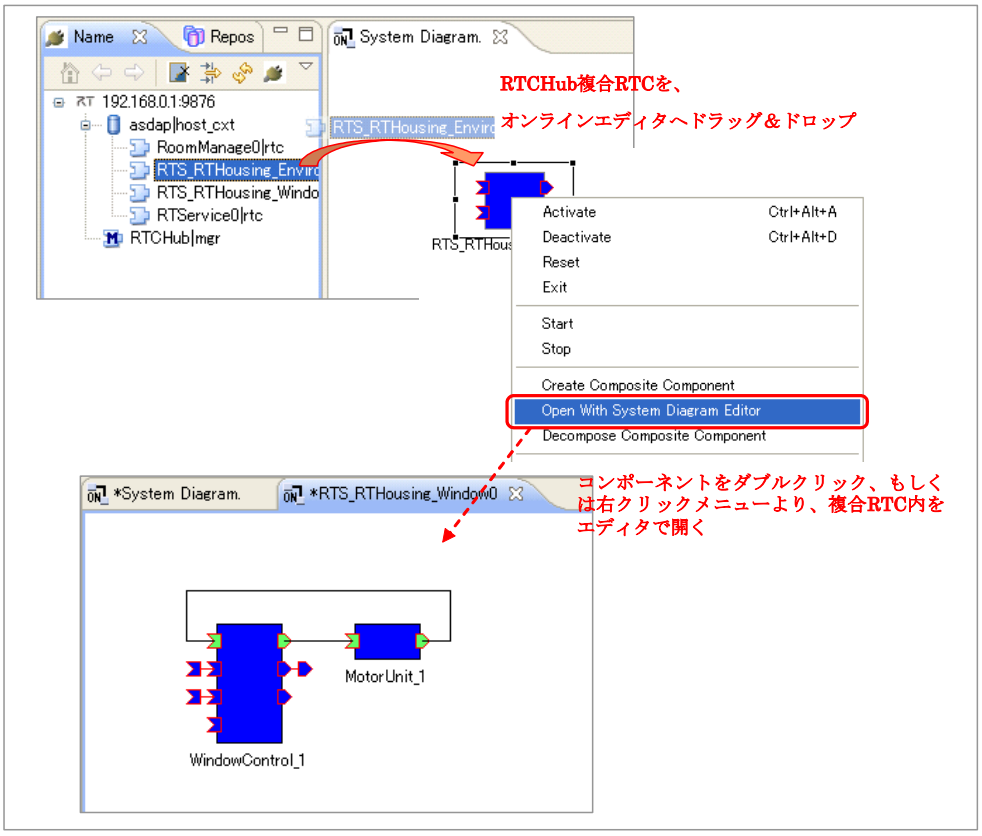


図 a-3-44 miniRTC/microRTC 管理･制御機能

〇外部仕様

以下に，本ツールを適用した画面構成を示す．基本的な画面構成は通常の RTSystemEditor と同様である．

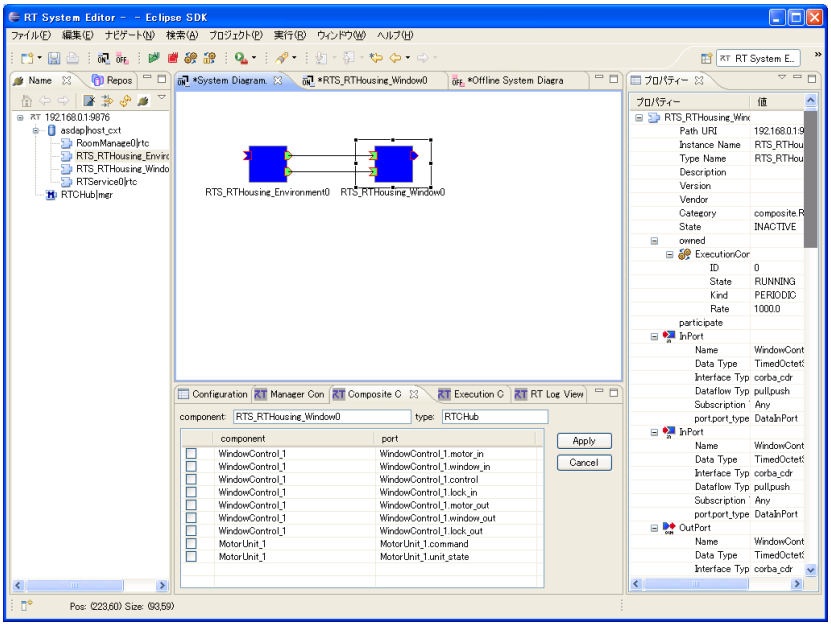


図 a-3-45 RTCHub/miniRTC/microRTC 用 RTSystemEditor

・プラグアンドプレイ設定ツール

〇機能概要

本ツールは大きく3つの機能を提供する．1つ目は， RT システム構築支援ツールから出力された RT システム仕様記述形式の XML ファイル(RTSProfile)をインポートし，プラグアンドプレイ情報を設定した上で，RTSProfile（プラグ＆プレイ情報が付加されたもの）をエクスポートする「プラグアンドプレイ情報設定機能」である．2つ目の機能は，プラグアンドプレイ情報を付加した RTSProfile を保存する際に，プラグアンドプレイ情報とシステムの整合性をチェックする「プラグアンドプレイ情報バリデーション機能」である．そして3つ目の機能は，設定したプラグアンドプレイ情報に応じて，対象となる RT コンポーネントの状態を制御する「RT コンポーネント制御機能」である．

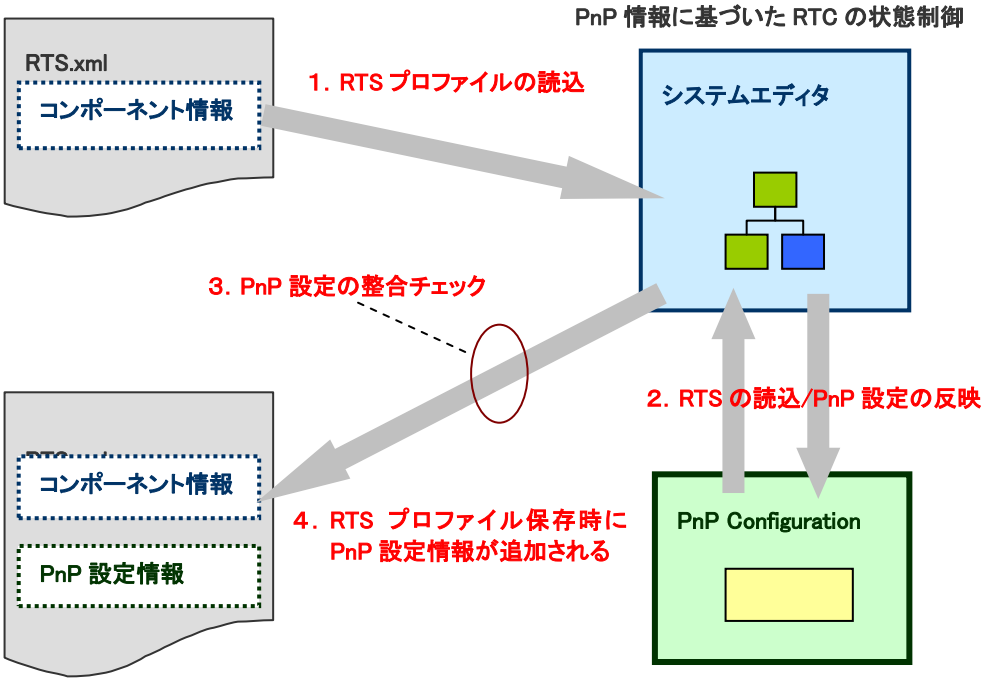


図 a-3-46 プラグアンドプレイ設定ツール 機能概要

○RT システム仕様記述方式の概要

本ツールで編集する RT システムの仕様情報は，独立行政法事 産業技術総合研究所（AIST）の「RT システム仕様記述方式」に基づいて管理される．このうち，本ツールで編集する「プラグ＆プレイ情報」について概要を示す．以下は RT システム仕様記述方式における基本要素のうち，プラグアンドプレイ情報の範囲を示したものである．

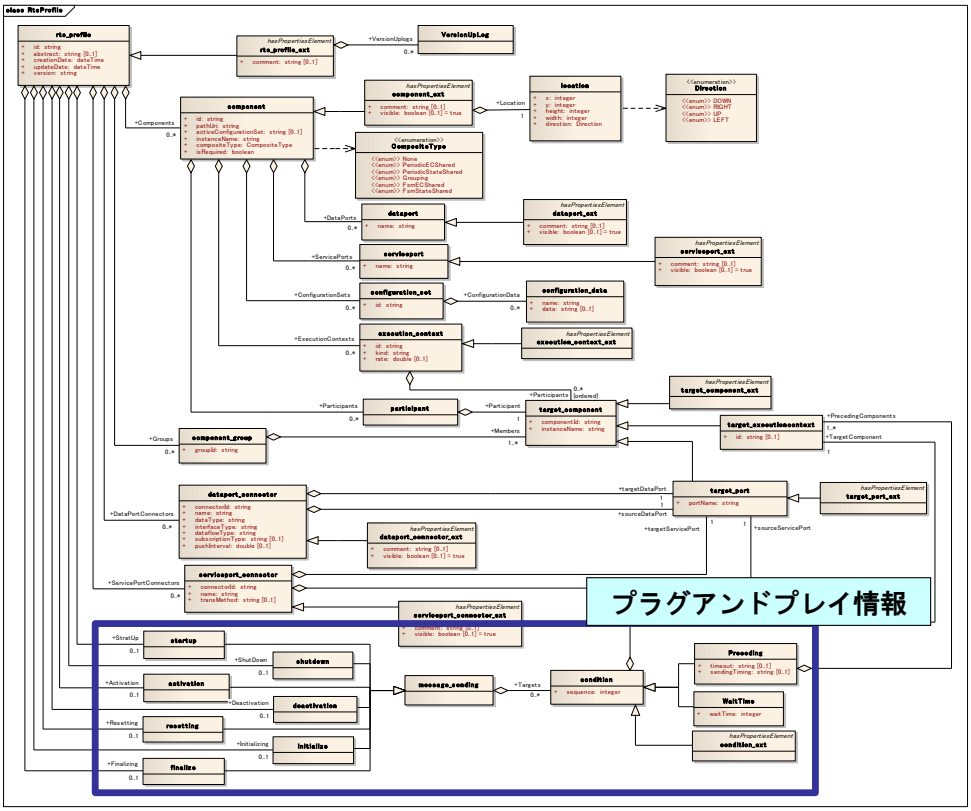


図 a-3-46 プラグアンドプレイ情報

プラグアンドプレイ情報は，当該 RT システムに参加する RT 要素部品（RT コンポーネント）が「起動」「終了」「活性化」「非活性化」といったイベントごとに，どのような順番や条件で実行されるかを保持するものである．以下は RT システム仕様記述方式で定義された要素群のうち，プラグアンドプレイ情報を抜粋したものである．

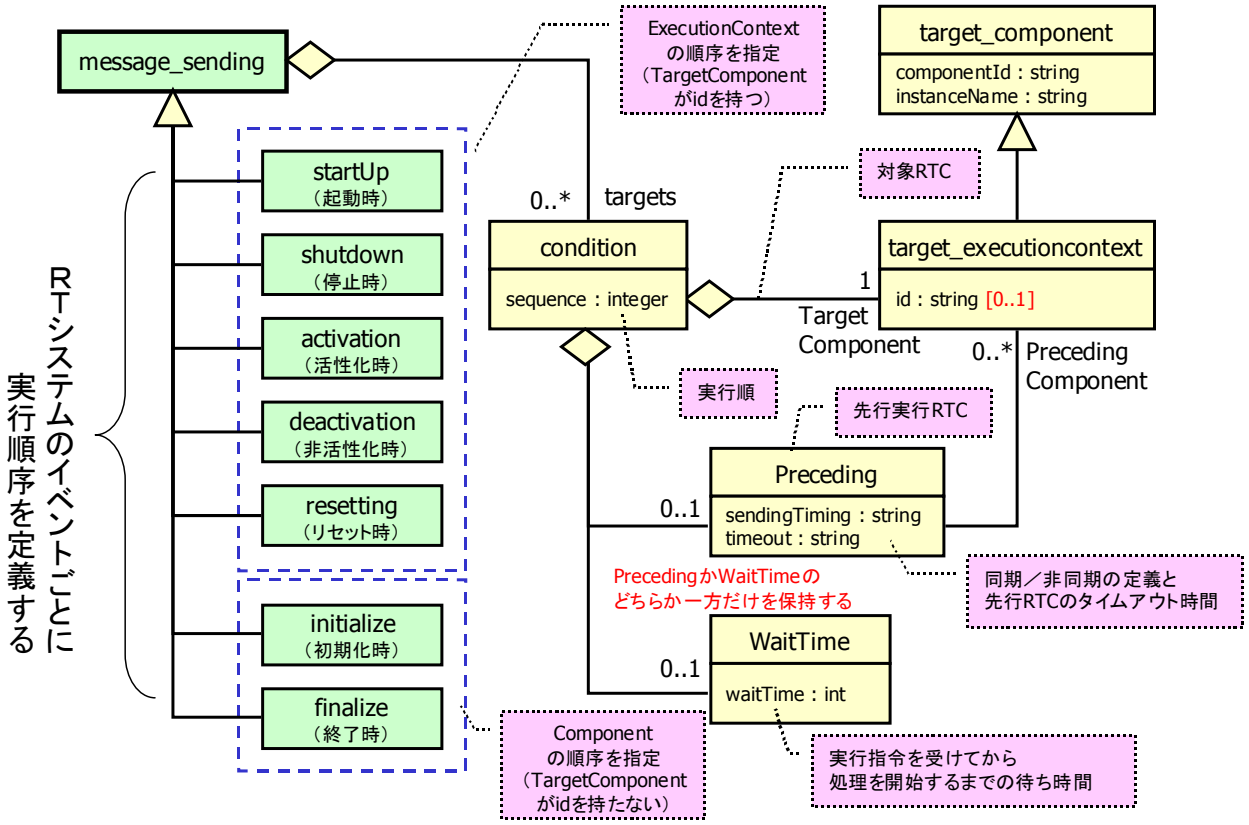


図 a-3-47 プラグアンドプレイ情報(詳細)

プラグアンドプレイ情報の設定対象となるイベントは，「message_sending」を継承した要素として定義されている．各イベントの実行順序を保持する要素として「condition」がある．この要素の属性「sequence」には整数値で順番が定義され，実行の対象となる RT 要素部品（RT コンポーネント）が「target_executioncontext」に定義される．また，各「condition」ごとに制約条件として「Preceding」または「WaitTime」が定義される．「Preceding」は先行実行 RT 要素部品のことであり，当該 RT 要素部品を実行する以前に実行されていなければならない他の RT 要素部品が定義されるものである．この要素の属性「sendingTiming」には同期，非同期を表す文字列が定義される．一方「WaitTime」は待機時間のことであり，当該 RT 要素部品の実行指示を受け取った後，実際に実行するまで待機する時間が定義される（単位はミリ秒）．

〇インスタンスイメージ

プラグアンドプレイ情報の具体的なインスタンスイメージを示す．以下はイベント「startUp」（起動時）の実行順序と制約条件を定義した例である．このイベントでは3つの「condition」が存在している．この中で最初に実行されるのは sequence に1が設定された RT 部品要素「Comp1Ec1」である．この condition には「WaitTime」に「1000」が設定されているため，1秒待機した後に実行される．次に実行されるのは RT 部品要素「Comp2Ec1」である．「WaitTime」に「500」が定義されているため，「Comp1Ec1」の実行が完了した後，0.5 秒待機した後にこの RT 要素部品が実行される．そして，最後に実行されるのは RT 部品要素「Comp3Ec1」である．「WaitTime」に「750」が定義されているため，「Comp2Ec1」の実行が完了した後，0.75 秒待機した後にこの RT 要素部品が実行される．

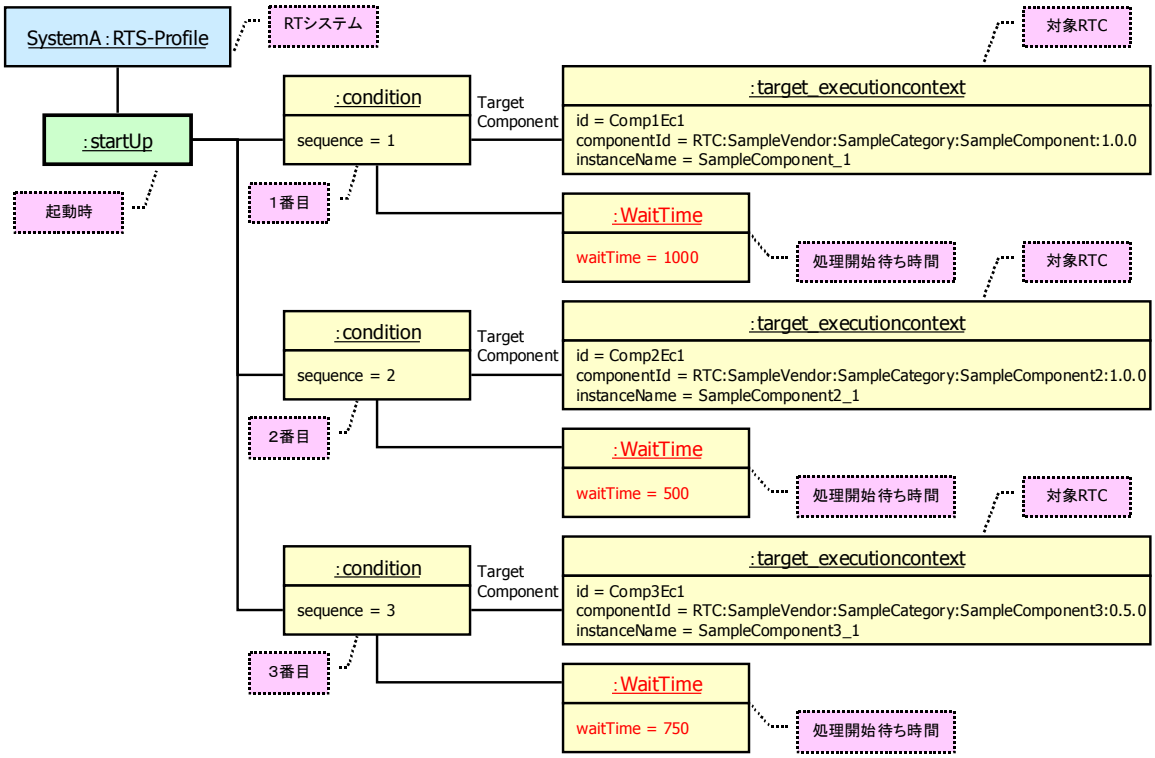


図 a-3-48 インスタンスイメージ 1

次に，Preceding を利用したインスタンスイメージを示す．以下はイベント「shutDown」（停止時）の実行順序と制約条件を定義した例である．このイベントでは3つの「condition」が存在しているが，2つ目に実行される RT 部品要素「Comp2Ec1」には Preceding（先行実行 RTC）として「Comp3Ec1」が指定されている．さらに属性「sendingTiming」に「SYNC（同期）」が指定されているため，「Comp3Ec1」の実行が完了するのを待って「Comp2Ec1」は実行することになる（「timeout」に「100」が指定されているため，0.1 秒待っても「Comp3Ec1」の実行が完了しない場合は，一連の処理がエラーとなる）．

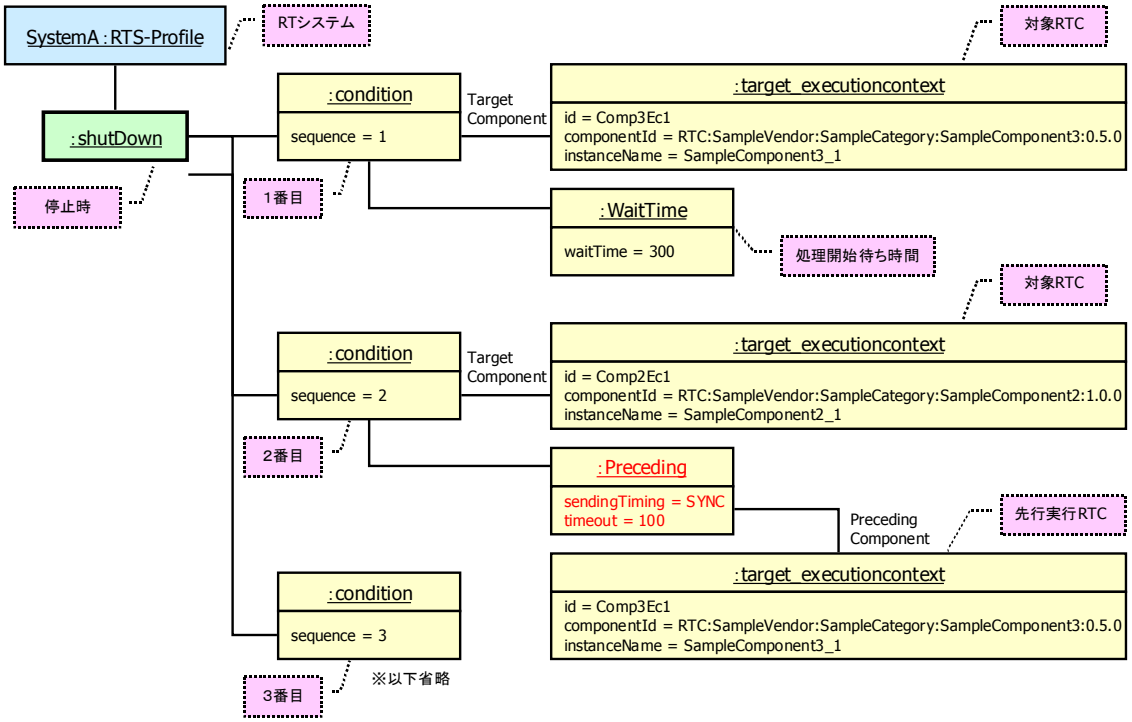
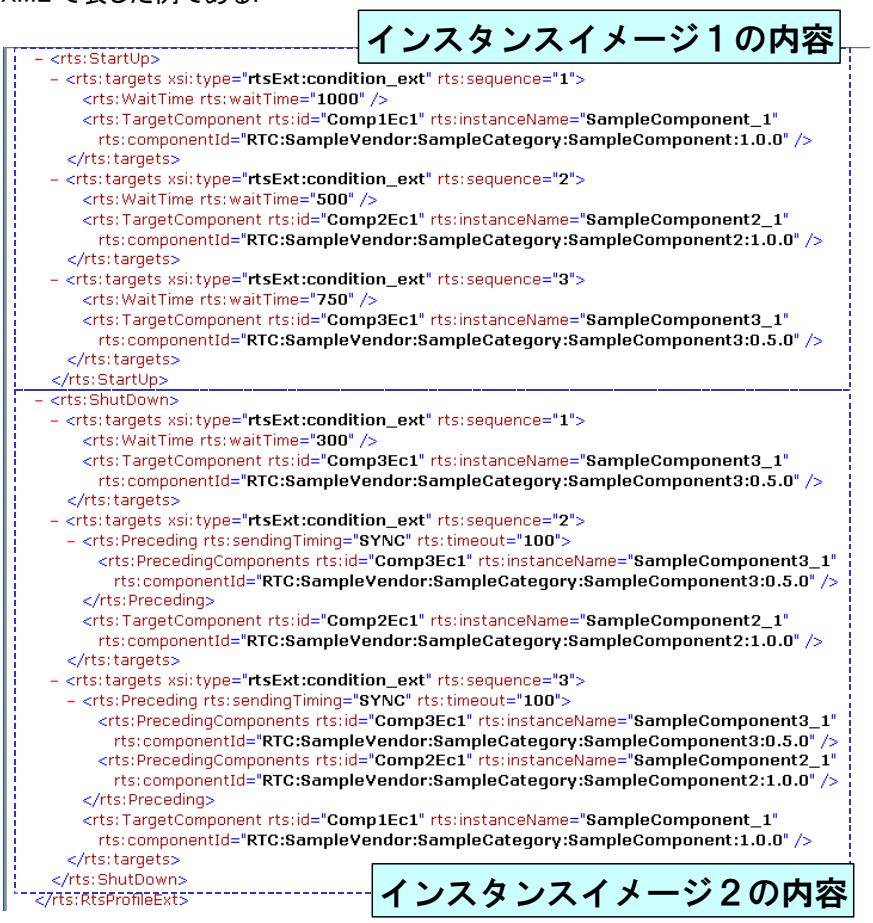


図 a-3-49 インスタンスイメージ 2

○XML イメージ

プラグアンドプレイ情報を RT システム仕様記述方式の XML 形式で表したイメージを示す．以下は，前述のインスタンスイメージを XML で表した例である．




```
- <rts:StartUp>
  - <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="1">
      <rts:WaitTime rts:waitTime="1000" />
      <rts:TargetComponent rts:id="Comp1Ec1" rts:instanceName="SampleComponent_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0" />
    </rts:targets>
  - <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="2">
      <rts:WaitTime rts:waitTime="500" />
      <rts:TargetComponent rts:id="Comp2Ec1" rts:instanceName="SampleComponent2_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0" />
    </rts:targets>
  - <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="3">
      <rts:WaitTime rts:waitTime="750" />
      <rts:TargetComponent rts:id="Comp3Ec1" rts:instanceName="SampleComponent3_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0" />
    </rts:targets>
  </rts:StartUp>
- <rts:ShutDown>
  - <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="1">
      <rts:WaitTime rts:waitTime="300" />
      <rts:TargetComponent rts:id="Comp3Ec1" rts:instanceName="SampleComponent3_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0" />
    </rts:targets>
  - <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="2">
    - <rts:Preceding rts:sendingTiming="SYNC" rts:timeout="100">
        <rts:PrecedingComponents rts:id="Comp3Ec1" rts:instanceName="SampleComponent3_1"
          rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0" />
      </rts:Preceding>
      <rts:TargetComponent rts:id="Comp2Ec1" rts:instanceName="SampleComponent2_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0" />
    </rts:targets>
  - <rts:targets xsi:type="rtsExt:condition_ext" rts:sequence="3">
    - <rts:Preceding rts:sendingTiming="SYNC" rts:timeout="100">
        <rts:PrecedingComponents rts:id="Comp3Ec1" rts:instanceName="SampleComponent3_1"
          rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent3:0.5.0" />
        <rts:PrecedingComponents rts:id="Comp2Ec1" rts:instanceName="SampleComponent2_1"
          rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent2:1.0.0" />
      </rts:Preceding>
      <rts:TargetComponent rts:id="Comp1Ec1" rts:instanceName="SampleComponent_1"
        rts:componentId="RTC:SampleVendor:SampleCategory:SampleComponent:1.0.0" />
    </rts:targets>
  </rts:ShutDown>
</rts:RtsProfileExt>
```

図 a-3-50 XML イメージ

○全体アーキテクチャ概要

本ツールは統合開発環境（IDE）Eclipse プラットフォームのプラグインとして動作することを前提とする．また，本ツールの開発では共通ライブラリを介して，XML 入出力の機能で JAXB（Java Architecture for XML Binding）を使用する．以下に UML の配置図で表した本ツールの全体アーキテクチャ概要を示す．

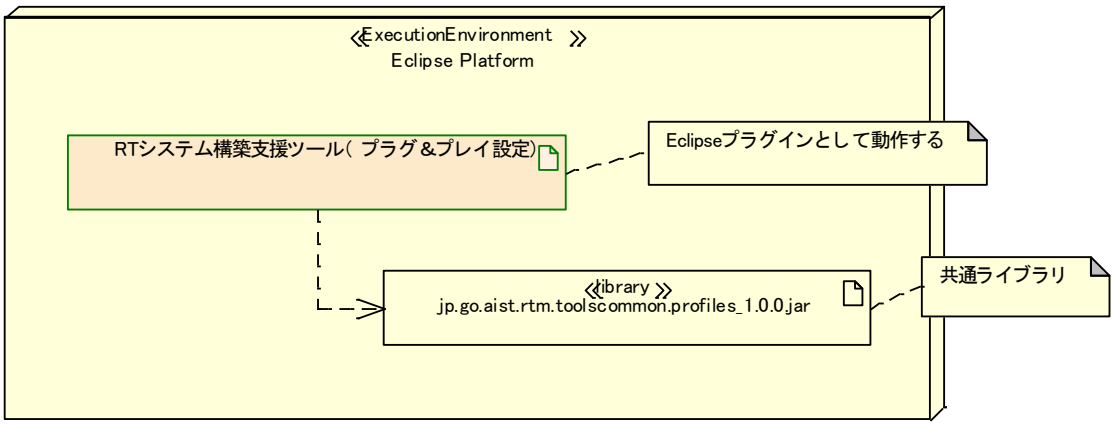


図 a-3-51 全体アーキテクチャ概要

○外部仕様

以下に本ツールの基本的な画面構成を示す．

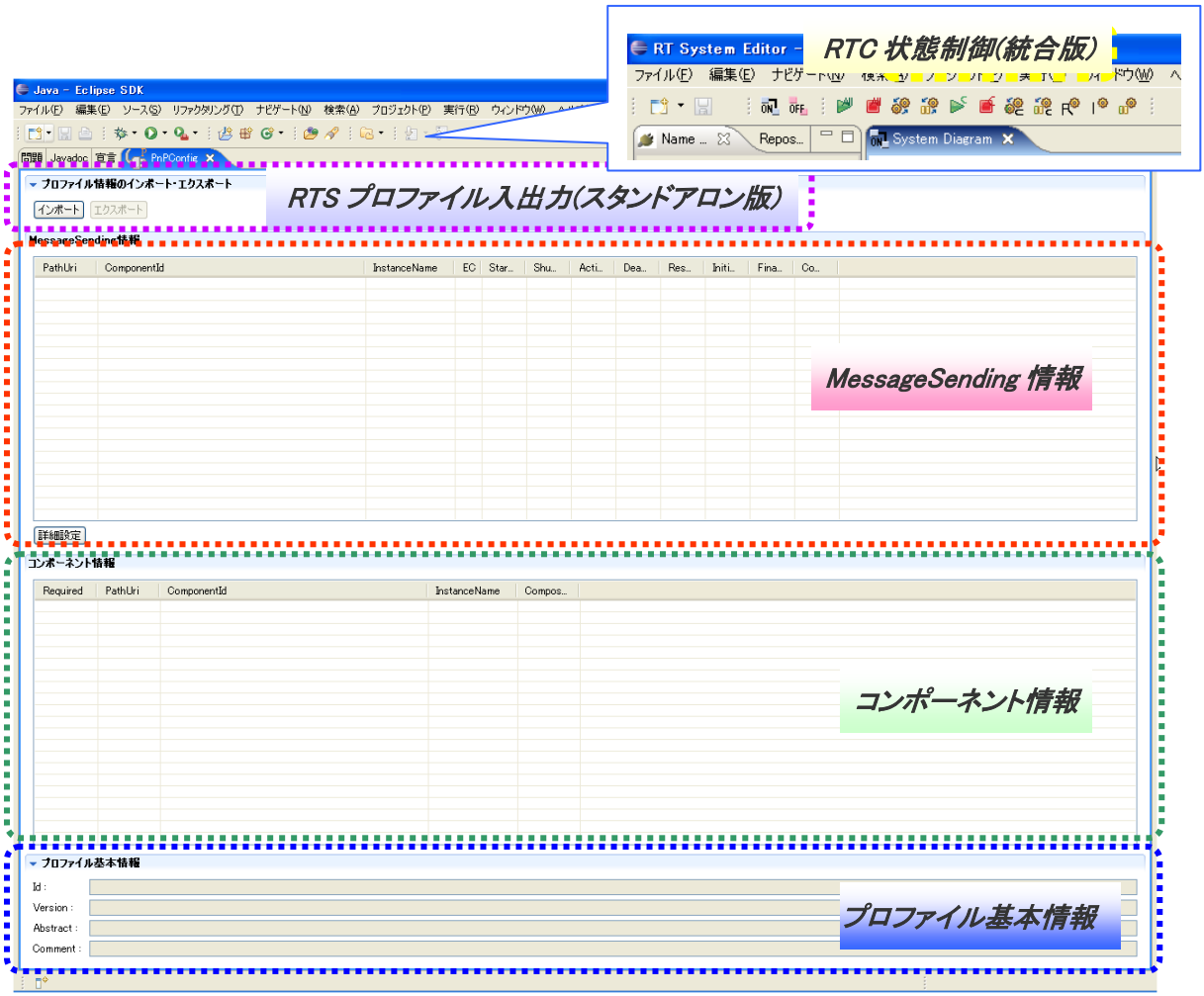


図 a-3-52 全体画面構成

本ツールの画面は大きく3つのパートに分けられる．画面下部の「プロファイル基本情報」は RT システム仕様の基本情報が表示される（参照のみで内容の編集はできない）．画面中段の「コンポーネント情報」には RT システムに参加する RT 要素部品の実行コンテキスト（ExecutionContext）の一覧がテーブルに表示される．ここでは必須の要素部品かどうかをチェックボックスで指定することができる．画面上部のパート「MessageSending 情報」にはイベントごとの RT 要素部品の実行順序が表示される．このテーブルはマトリクス形式になっており，行には RT システムに参加する RT 要素部品の一覧が，列にはイベントが表示され，各イベントにおいて実行される RT 要素部品に対し，整数値で順序が定義される．整数値を入力した箇所を選択して「詳細設定」ボタンを押下すると，RT 要素部品実行時の制約を入力する「詳細設定ダイアログ」が表示される．

詳細設定ダイアログは，RT 要素部品の実行時制約を入力するダイアログである．実行時制約は「WaitTime」または「Preceding」の 2 種類があり，どちらか一方だけを指定することになる．「WaitTime」は待機時間のことであり，当該 RT 要素部品の実行指示を受け取った後，実際に実行するまで待機する時間を指定する場合に使用される．一方、「Preceding」は先行実行 RT 要素部品のことであり，当該 RT 要素部品を実行する以前に実行されていなければならない他の RT 要素部品を指定する場合に使用される．「WaitTime」を指定する場合は，「WaitTime」の欄にミリ秒で待ち時間を入力する．

「Preceding」では，まず「SendingTiming」を指定する．通常は「SYNC（同期）」または「ASYNC（非同期）」のどちらかが選択されるが，それ以外についても任意の文字列を指定することができる．さらに，テーブルの横にある「Add」ボタンを押下し，「先行実行 RTC 選択ダイアログ」で先行実行 RT 要素部品を指定する．

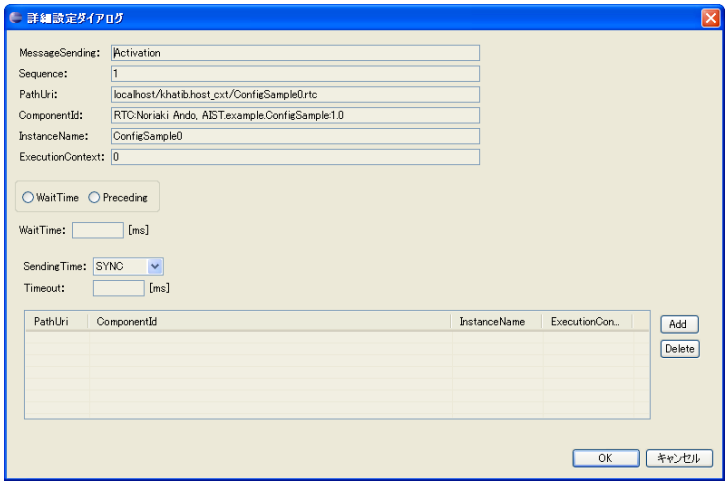


図 a-3-53 詳細設定ダイアログ

先行実行 RTC 選択ダイアログは，RT 要素部品の実行時制約のうち，先行実行 RT 要素部品を指定するためのダイアログである．当該 RT 要素部品を実行する前にあらかじめ実行しておくべき RT 要素部品を選択する（複数選択可）．

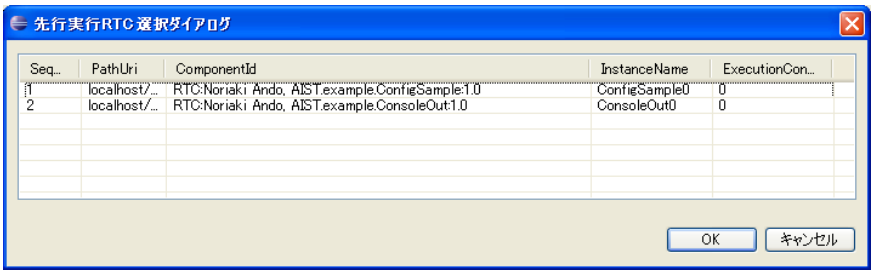


図 a-3-54 先行実行 RTC 選択ダイアログ

○パッケージ構成
本ツールは以下のような役割ごとにパッケージを構成する。
・ユーザインタフェース:ダイアログ(dialog)
・ユーザインタフェース:ビュー(views)
・RT システム情報保持(dto)
・共通処理(util)

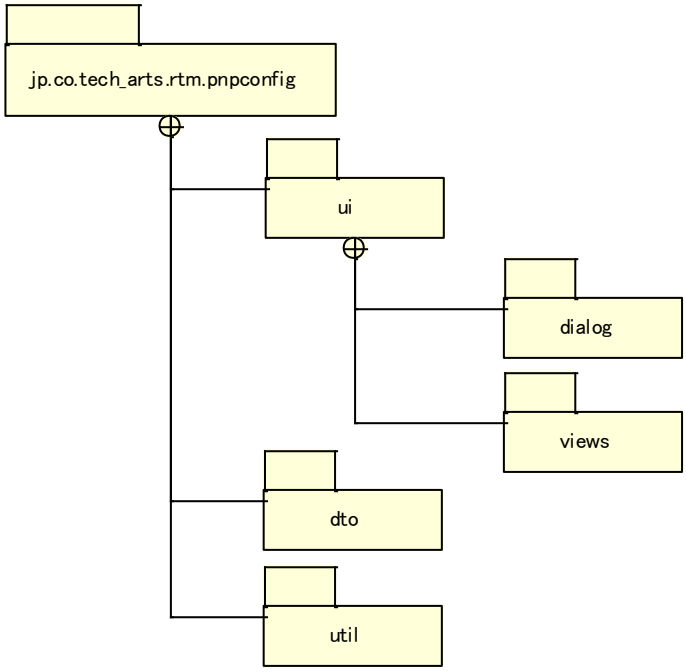


図 a-3-55 パッケージ構成

これらパッケージの依存関係を以下に示す. ユーザインタフェースから各種機能を呼び出す形になるため, 以下のような依存関係となる.

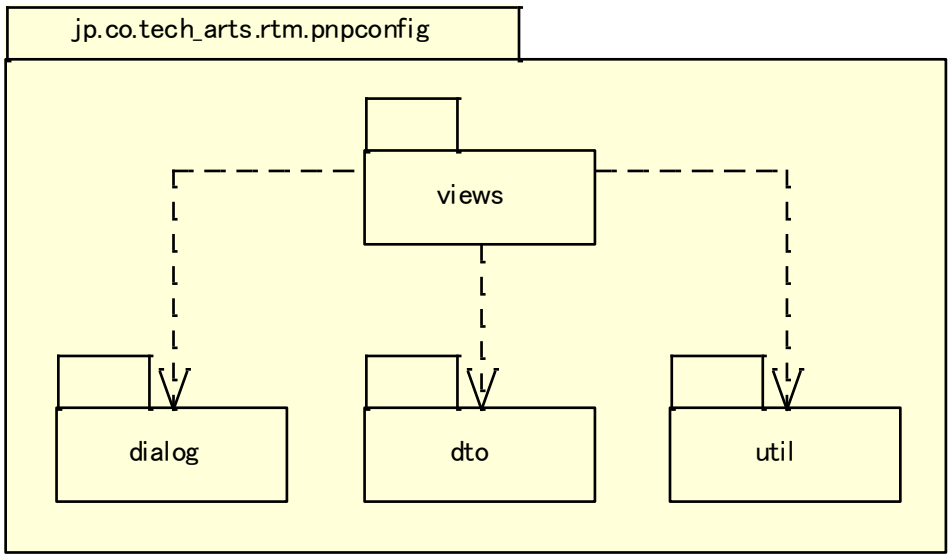


図 a-3-56 パッケージ間の依存関係

○内部構造
RT 要素部品情報編集機能を実現するための基本的なクラス構造を示す.

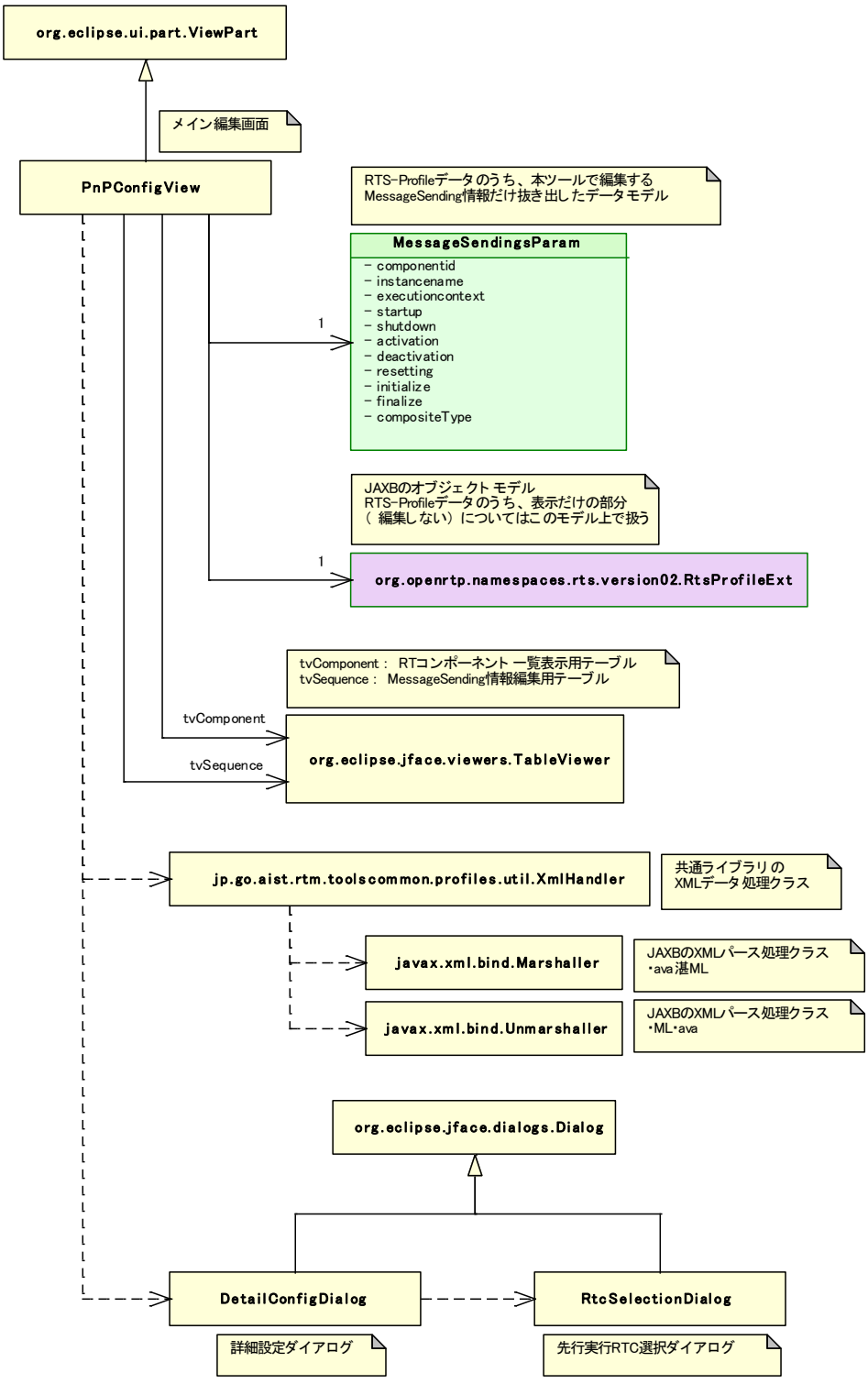


図 a-3-57 クラス構造

本ツールのメイン画面は Eclipse の「org.eclipse.ui.part.ViewPart」を継承した「PnPConfigView」である．この画面上に配置された2つのテーブルに対してプラグアンドプレイ情報を設定していく(JFace の TableViewer を利用している)．テーブル「tvComponent」は画面中段に位置する「コンポーネント情報」のテーブルであり，RT システムに参加する RT 要素部品（ExecutionContext）の一覧情報を扱う．テーブル「tvSequence」は画面下に位置する「MessageSending 情報」のテーブルであり，各イベントの実行順序と制約情報を扱う．「MessageSending 情報」の編集では，詳細設定ダイアログ(DetailConfigDialog)と先行実行 RTC 選択ダイアログの2つのダイアログを使用する

(JFace の Dialog を利用している). 編集したプラグ &プレイ情報は「RT システム仕様記述方式」の XML 形式でエクスポートされる. また, 過去に編集したプラグアンドプレイ情報を「RT システム仕様記述方式」の XML 形式でインポートする. このインポート・エクスポートでは共通コンポーネントの「XmlHandler」を利用する.

○処理フロー(インポート)

以下にインポート時の処理フローを示す.

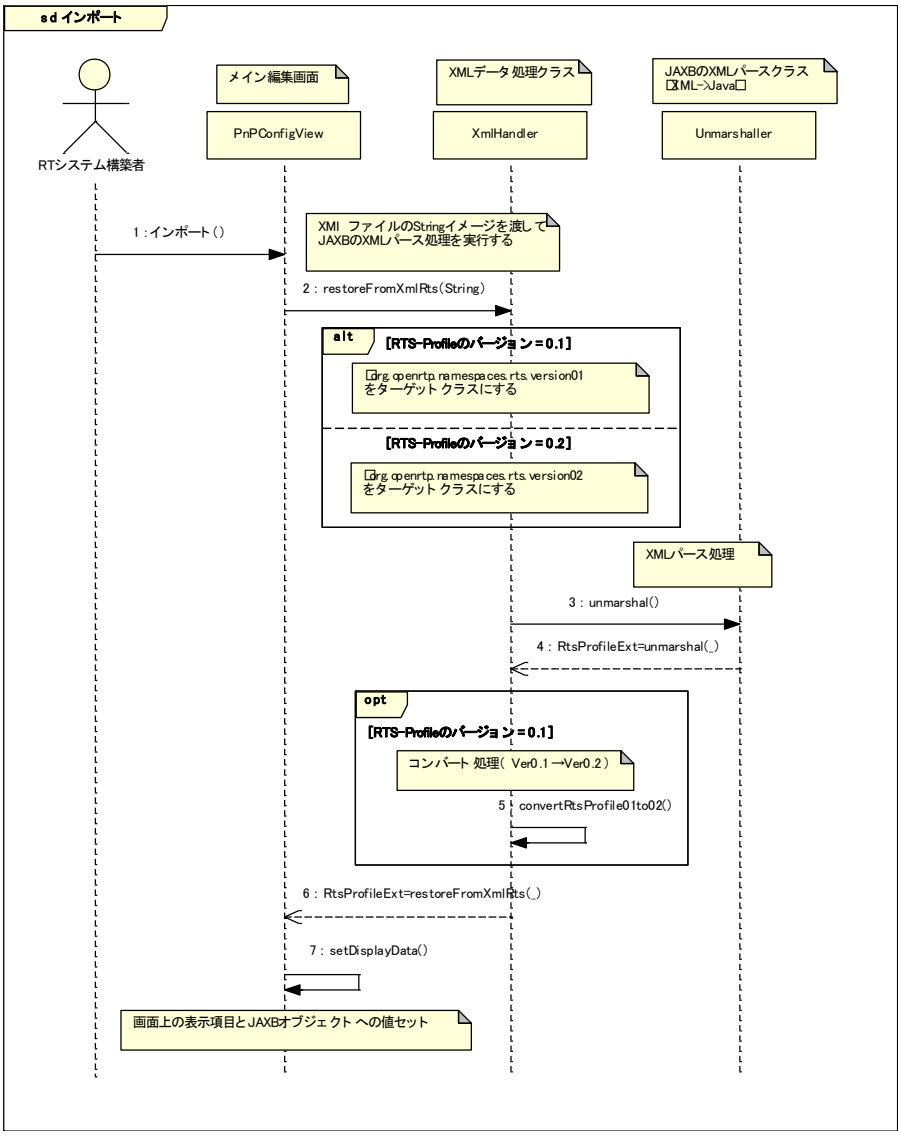


図 a-3-58 インポート時シーケンス図

ツールのユーザによってインポート機能が起動されると, まずメイン編集画面である「PnPConfigView」が XML インポート処理を共通ライブラリ側のクラス「XmlHandler」に処理を委譲する. 次に「XmlHandler」は XML ファイルの一部をパースし, RT システム仕様記述方式のバージョン(ver0.1／ver0.2)を判断する. その後, バージョンに応じた XML ファイル全体のパース処理を実行し, Ver0.1 の場合は ver0.2 の形式にコンバートする処理を実行した上で画面上に読み込んだデータをセットする.

○処理フロー(エクスポート)

以下にエクスポート時の処理フローを示す.

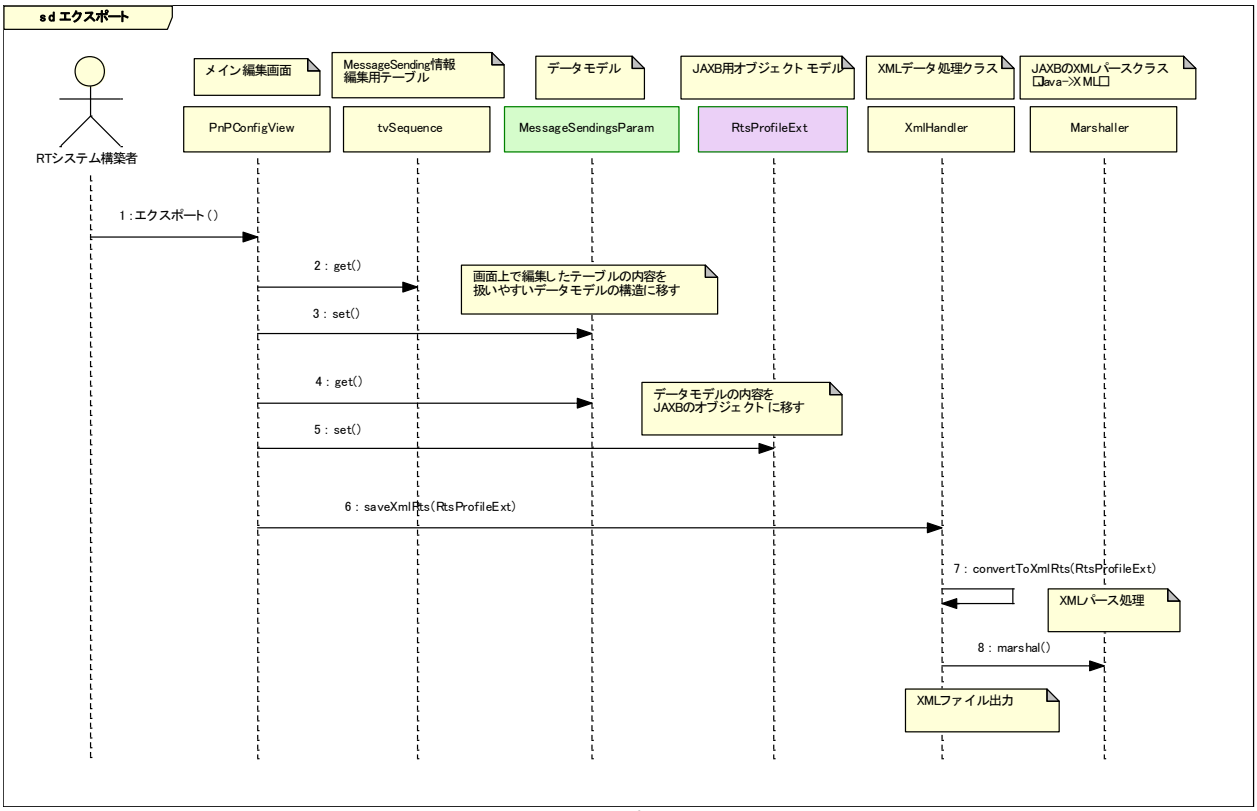


図 a-3-59 エクスポート時シーケンス図

ツールのユーザによってエクスポート機能が起動されると，まずテーブル「tvSequence」からデータモデル「MessageSendingsParam」に「MessageSending 情報」の内容を移し変える．次に「MessageSendingsParam」から JAXB のオブジェクトである「RtsProfileExt」に値を移し変え共通ライブラリ側のクラス「XmlHandler」によって XML ファイルの生成を行う．

・RT システムローダー

〇機能概要

本ツールは，オフラインにて作成した RT システム情報(RtsProfile)から，実際に動作するシステムを構築する機能を提供する．実システムを構築するための手法としては，NameService に登録されている起動済み RTC を用いる方法と，OpenRTM-aist の RT コンポーネント管理機能(RTManager)を利用し，新しいインスタンスを生成する方法を提供する．

○外部仕様

以下に，本ツールの画面構成を示す．

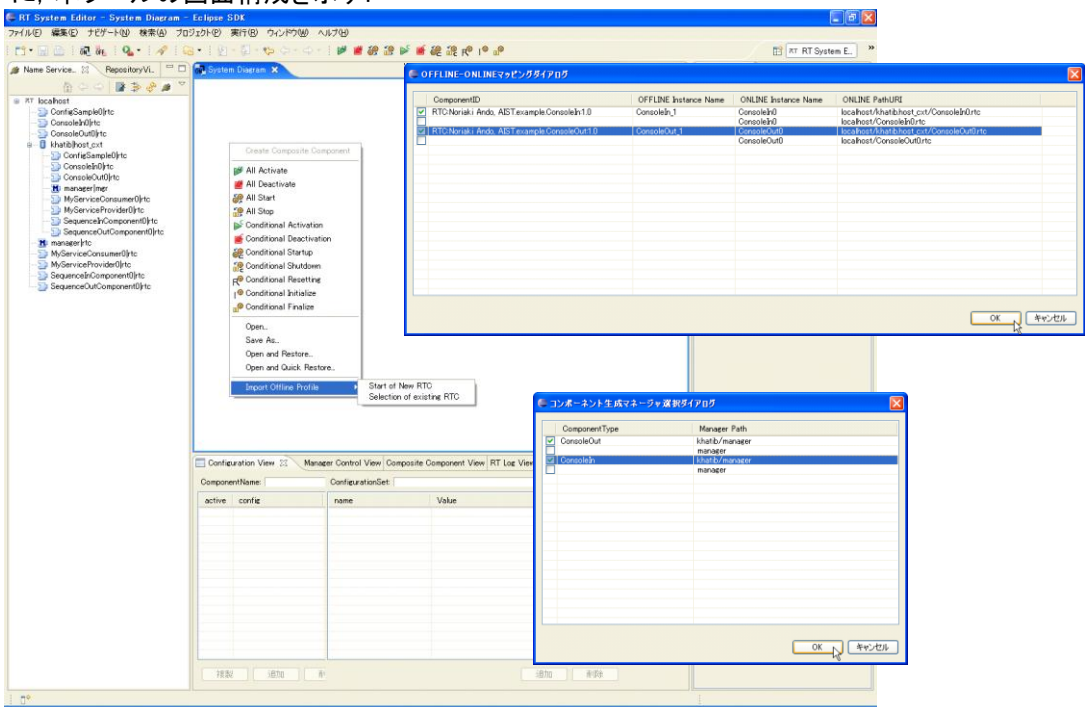


図 a-3-60 画面構成

起動済み RTC を用いてシステムを構築する場合，指定されたオフライン RTSProfile の情報を基に，NameServiceView の中からマッピング先の候補となる RTC を検索し，ComponentID(オフライン RtsProfile および NameService に登録された RTC の ID 情報．検索のキーであるため共通となる)，OFFLINE Instance Name(オフライン RtsProfile に保存されている RTC のインスタンス名)，ONLINE Instance Name(NameServiceView に登録されている RTC のインスタンス名)，ONLINE PathUri(NameServiceView に登録されている RTC の PathUri)をマッピング画面に一覧表示する．なお，指定されたオフライン RTSProfile に含まれている RT コンポーネントと同じ ID を持つ RT コンポーネントが NameServiceView 内に存在しない場合には警告画面を表示する．また，実際のシステムを構築する際には，オフライン RTSProfile に保存されたインスタンス名を使用する．このため，起動済み RTC のインスタンス名がオフライン RTSProfile に保存されたインスタンス名と異なっていた場合は，インスタンス名が変更されることとなる．

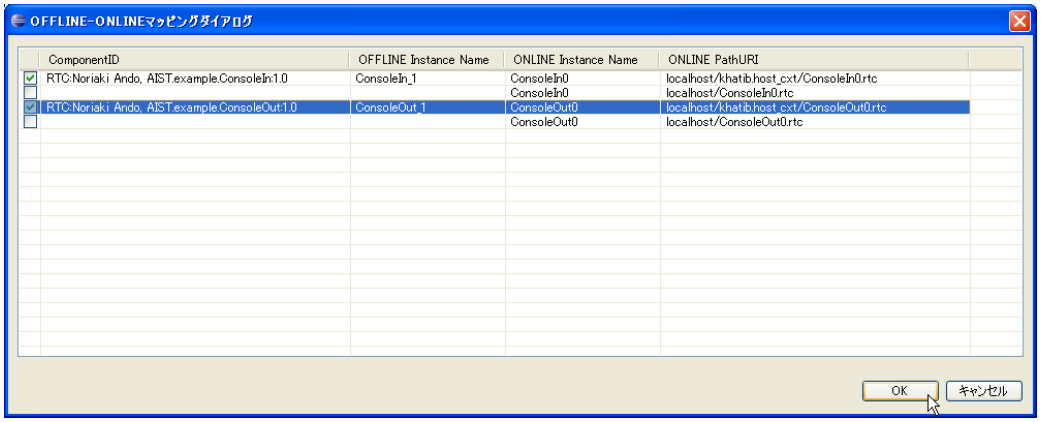


図 a-3-61 起動済み RTC マッピング画面

RTManager を用いてシステムを構築する場合，指定されたオフライン RTSProfile の情報を基に，NameServiceView に登録されている RTManager の中から必要な RTC を生成可能な RTManager を検索し，ComponentType(フライン RtsProfile に登録された RTC の型情報)，Manager Path(指定された型の RTC を生成可能な RTManager)を一覧画面に表示する．なお，RTManager を検索する際には，オフライン RtsProfile に保存された

RTC の型情報のみを用いる．そして，指定された型の RTC を生成可能な RTManager が NameServiceView 内に存在しない場合には，警告画面を表示する。更に，選択した RTManager 配下に同一インスタンス名の RTC が既に存在している場合には，その起動済み RTC を用いてシステムを構築する．

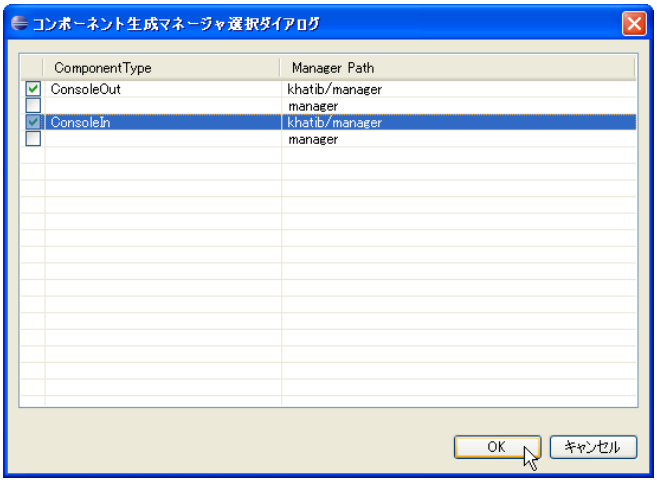


図 a-3-62 RTManager 一覧画面

a-3-4 RT システム開発支援ツールの開発

◎目的

本ツールは，「住宅メーカー」や「システムインテグレーター」が，RT システムとして実現すべき振る舞い，サービスを定義する際の開発負荷を低減することを目的としている．

「住宅システム」では最終的なエンドユーザーである「顧客」から出てくる多彩な要望を基に，システム全体の振る舞いや，提供していくサービスを構築する必要がある．そして，システムが完成した後は，実際の現場にて構築した各種サービスの説明を行うとともに，要望に応じた調整などを行う必要がある．この場合，最終的なシステムの説明，構築を行うのは，「住宅メーカー」の営業担当者など，よりエンドユーザーである「顧客」に近い立場の人である事が想定される．本ツールでは，RT システム内に組み込んだ RT コンポーネントの仕様から，プログラミングを行うことなく RT システム全体の振る舞い，提供するサービスを構築する機能を実現する．本ツールを利用することで，各種プログラミングに関する知識が少ないユーザーが，RT システム全体の振る舞い，提供するサービスを構築，修正，調整することができるようになる．全体システムの中において，本ツールが支援を行う範囲を以下に示す．

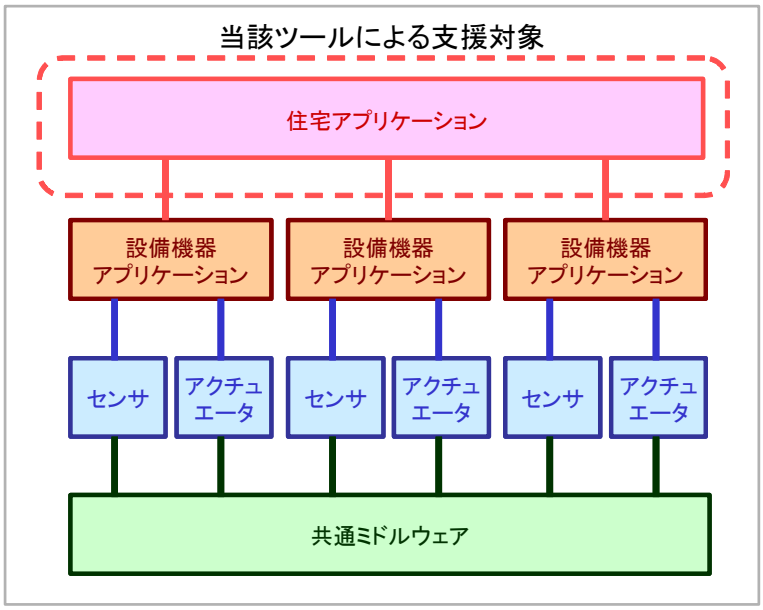


図 a-3-63 RT システム開発支援ツールの適用範囲

◎全体概要

本ツールは，対象となる RT システム内に配置された RT コンポーネントの仕様情報(RTCProfile)から，システム全体の振る舞いを定義する機能を実現している．また，作成した振る舞いに従って，システム全体を制御する機能も実現している．更に，システムの振る舞い定義は，グラフィカルなエディタを用いて，プログラミングレスで行うことが可能である．

本ツールでは，RTCProfile の情報を基に，システム全体の状態遷移を定義するために必要な要素を定義することが可能であり，各データ出力ポートのデータ出力値やサービスポートとの出力，時間条件（一定周期，年月日時間指定）に基づく，トリガイベントやガード条件の定義，状態内で実行するアクションの定義を行うことができる．そして，定義した構成要素を仕様して，グラフィカルエディタ上で対象システムが持つ状態や，状態間を遷移するトリガイベントの指定，各状態内で実行するアクションを指定することで，システム全体の振る舞い（状態遷移）を定義することができる．

更に，既存の状態遷移定義を再利用したり，粒度が細かくより詳細な状態遷移定義から，より広い視点での状態遷移を定義するために，複数の状態遷移をマージする機能も実現している．この機能を利用することで，ユーザーは類似のサービスを構築する際に，過去の資産を再利用することができ，開発効率を向上させることができる．また，定義した振る舞い，サービスを実現するために必要となる各 RT コンポーネント間の接続情報を RT システム仕様記述方式（RTSProfile）の形式で自動生成することができるため，対象システムを容易に構築することが可能となる．

また，実際の RT システムにおいて使用する RT コンポーネントが手元に揃っていない場合でも，各 RTC の仕様定義(RTCProfile)のみを利用してシステム全体の振る舞いを定義することができる機能も実現している．この機能を利用することで，システムの構成要素である RT コンポーネントと，最終的な RT システムの並行開発を実現するこ

とが可能となる.
本ツールの画面例を以下に示す.

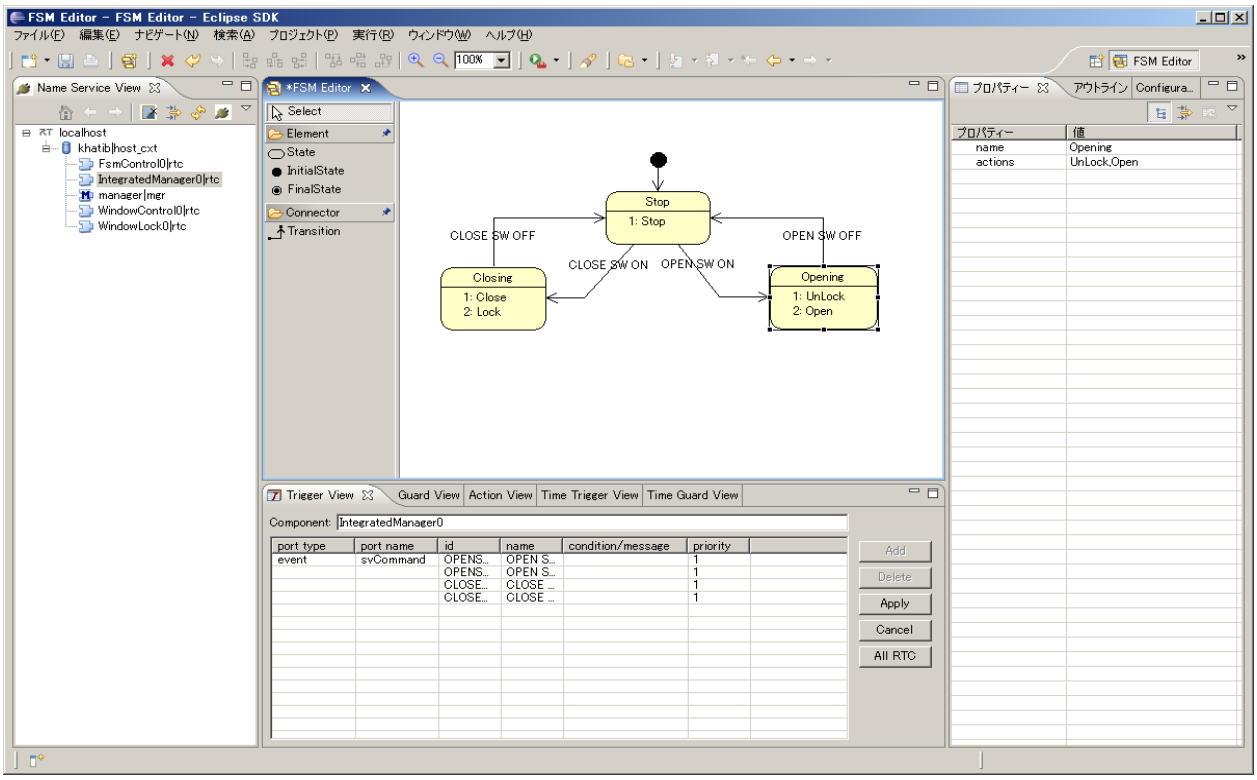


図 a-3-64 RT システム開発支援ツール

なお，本ツールは， Eclipse プラグインとして実現しているため，必要に応じて他の Eclipse プラグイン形式の開発環境を選択することで，利用者自身が使用する開発環境全体をカスタマイズする事ができるようになっている.

◎詳細内容
〇機能概要
本ツールは大きく5つの機能を提供する．1つ目は，使用する RT コンポーネントのポートに対して，イベントトリガ，ガード条件，アクションなどの状態遷移の構成要素を定義する「状態遷移構成要素定義機能」である．2つ目の機能は，グラフィカルエディタを用いて状態遷移を構築するとともに，属性設定ビューを用いて，状態，遷移などの要素に対して，状態遷移構成要素定義機能で設定したイベントトリガ，ガード条件，アクションを適用する「状態遷移構築機能」である．3つ目の機能は，編集中の状態遷移に別の状態遷移を取り込み，2つの状態遷移を1つに統合する「状態遷移統合機能」である．4つ目の機能は，作成した状態遷移定義を状態遷移制御用 RT コンポーネント用の設定ファイルとして RTC コンフィギュレーションの形式で出力する「設定情報生成機能」である．そして5つ目の機能は，作成した状態遷移定義から，状態遷移制御用 RT コンポーネントと対象システムを構築する RT コンポーネントとの接続情報を RTSProfile の形式で出力する「接続情報作成機能」である．本機能にて作成した RTSProfile を使用して RTSystemEditor の復元機能を利用することで，対象となるシステムをロードすることが可能となる．

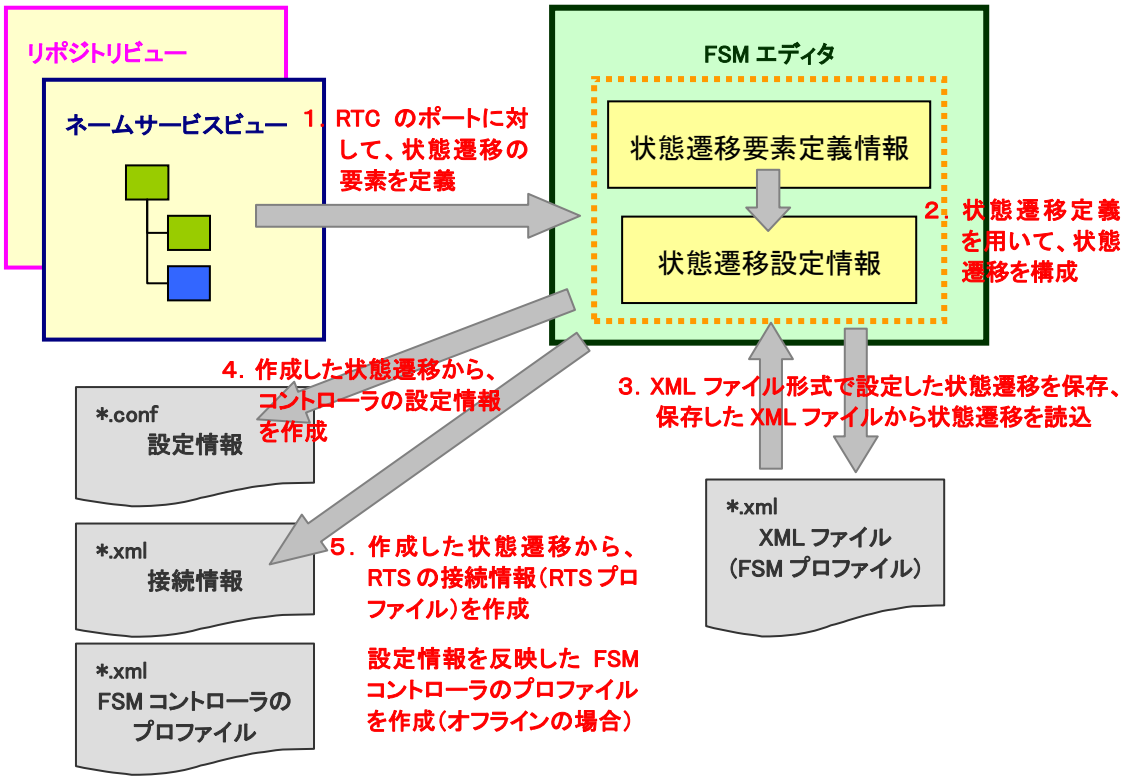


図 a-3-65 RT システム開発支援ツール 機能概要

対象となる RT システムの制御は，状態遷移制御用 RT コンポーネントが，システムを構成する RT コンポーネントからのイベントを受け取り，状態を判定し，それぞれの状態におけるアクションを起動することで実現する．
状態遷移制御用 RT コンポーネントが受け取るイベントは，RT コンポーネントのデータ出力ポートから特定の条件を満たす入力があった場合や，特定のサービスポートからのメッセージを受信した場合を意味する．また，時間的な条件によるイベントも存在する．また，状態遷移制御用 RT コンポーネントが起動するアクションは，RT コンポーネントの特定のサービスポートにメッセージを送信することを意味する．このため，RT システムが状態遷移を行うためには，状態遷移制御用 RT コンポーネントが状態遷移を制御するための設定情報と，イベントやアクションの送受信を行う RT コンポーネントのポートとの接続情報が必要となる．

本ツールを用いて対象システムの状態遷移を定義する場合，まず対象となる RT コンポーネントが登録されているネームサーバを，ネームサービスビューに追加する必要がある．そして，各種定義ビューを用いて RT コンポーネントのポートや，インターフェースに対してトリガやガード、アクションなどの構成要素を定義する．その後，グラフィカルエディタを用いて状態遷移図を構築し，インスペクタビューを用いて，状態や遷移の属性としてあらかじめ定義したトリガやガード，アクションを適用する．その後，定義した状態遷移をファイルに保存し，状態遷移制御用 RT コンポーネントの設定情報（rtc.conf 形式）を作成するとともに，状態遷移制御用 RT コンポーネントと構成する RT コン

ポーネントとのポート接続情報を含む接続情報を作成する.
　その後，設定情報(*.conf)を読み込ませて，状態遷移制御用 RT コンポーネントを起動する(この際，状態遷移制御用 RT コンポーネントにイベントを受け取るポートと，アクションを送信するポートが自動で設定される)．そして，RTSystemEditor のオンラインエディタの復元機能を用いて，接続情報(RTSProfile)を読み込み，復元を行うと，状態遷移制御用 RT コンポーネントに設定されたポートと，各 RT コンポーネントのポートが接続される.

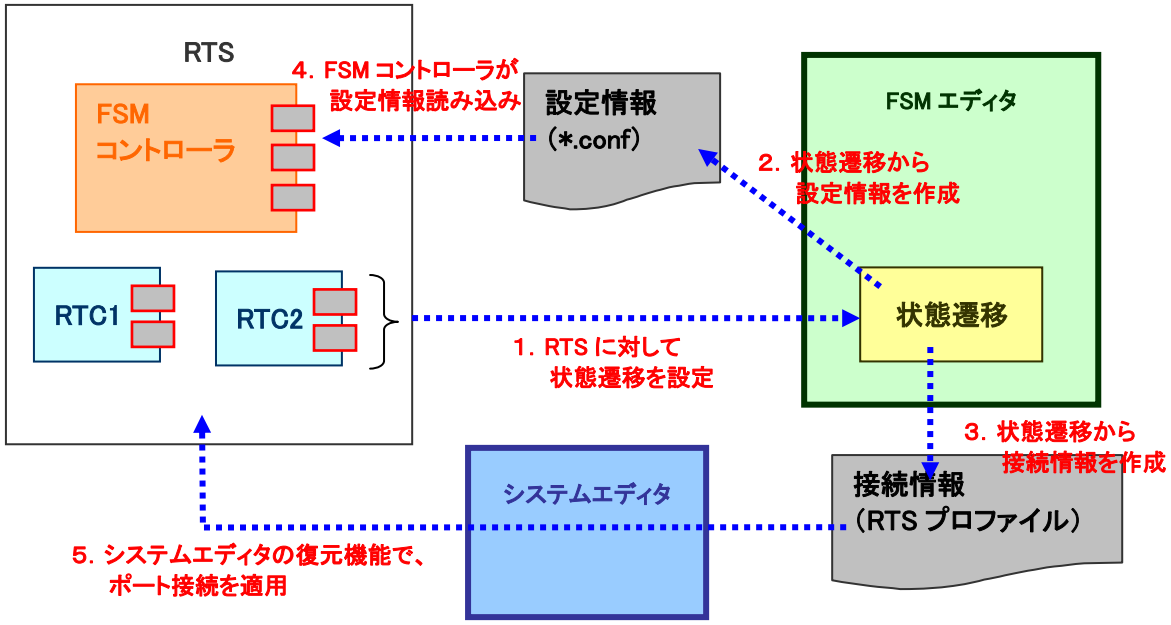


図 a-3-66 RT システム開発支援ツール 利用方法概要

○外部仕様

以下に，本ツールの画面構成を示す．

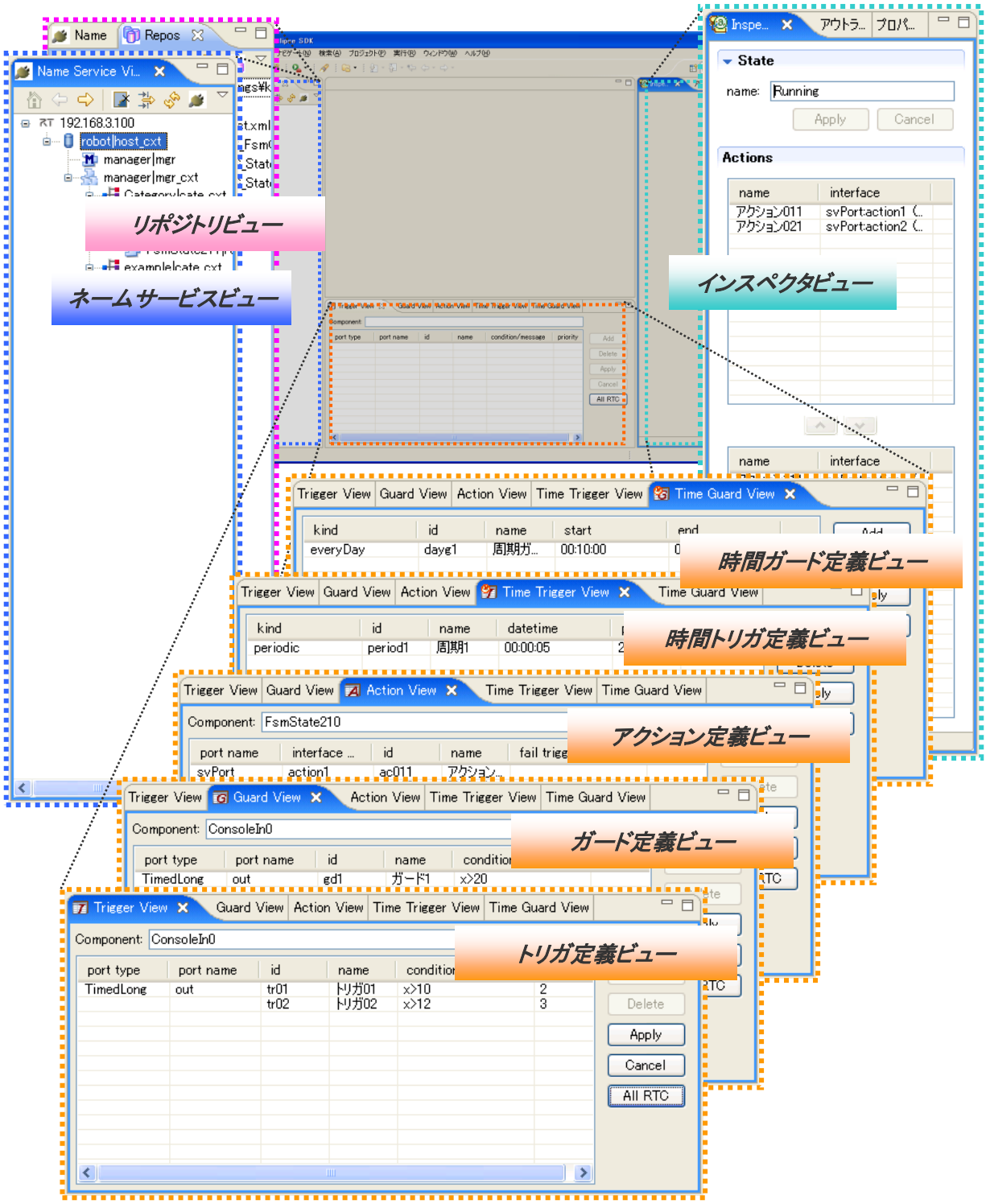


図 a-3-67 RT システム開発支援ツール　画面構成

「トリガ定義ビュー」では，ネームサービスビューで選択した RTC のポートに対してトリガの定義を行う．トリガには，データトリガとイベントトリガが存在する．ネームサービスビューで選択した RTC に条件に適合するポートが存在すると，ポートの一覧がテーブルに表示され，トリガを定義できるようになる．トリガ定義ビューの画面構成を以下に示す．

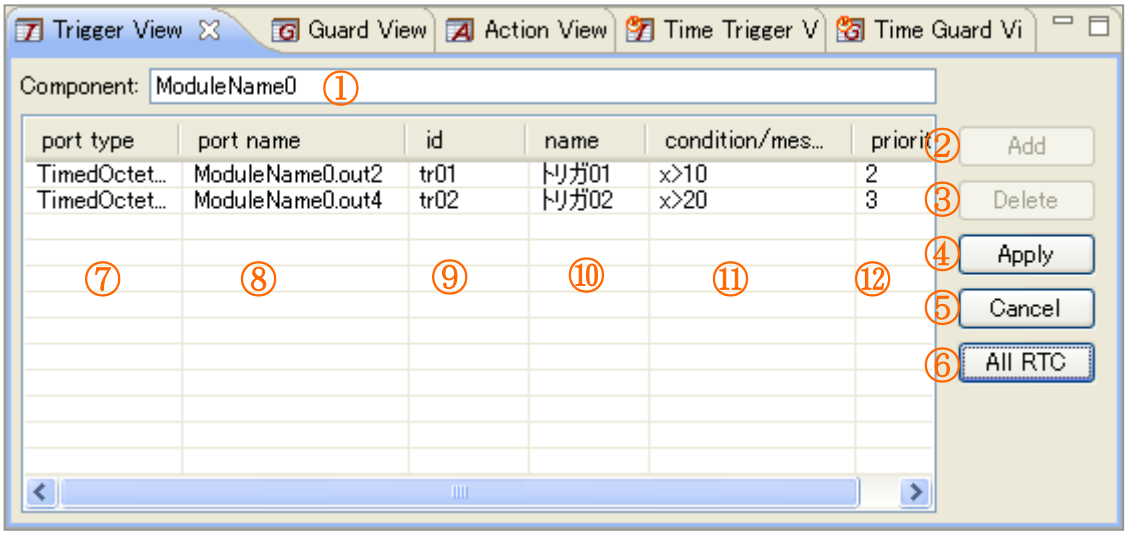


図 a-3-68 トリガ定義ビュー

①では現在選択されている RT コンポーネントの名称を表示する. ②は選択中のポートにトリガを追加し, ③で選択中のトリガを削除する. ④は現在編集中のトリガ定義内容を確定し, ⑤は逆に編集中のトリガ定義内容を破棄する. ⑥は RTC 一覧表示ダイアログを表示する.

⑦はポート種別を表示し, データ出力ポートの場合は「Data Type」, サービスポートの場合には「event」をそれぞれ表示する. ⑧はポート名称を表示する. ⑨はトリガ ID を定義する. 本項目は必須入力項目である. ⑩はトリガ名を定義する. ここで定義された名称が他の画面にて使用される. ⑪はトリガ定義本体を記述する. データ出力ポートの場合にはデータ条件を, サービスポートの場合にはメッセージ名を表示する. そして, ⑫ではトリガの優先順位を定義する. 複数トリガが同時に発生した場合, ここで指定した優先度に従って処理が実行される.

「ガード定義ビュー」では, ネームサービスビューで選択中の RTC のポートに対してガードを定義する. ガードはデータ出力ポートに対して設定する事が可能であり, ネームサービスビューで選択した RTC にデータ出力ポートが存在すると, ポートの一覧がテーブルに表示され, ガードを定義することが可能である. ガード定義ビューの画面構成を以下に示す.

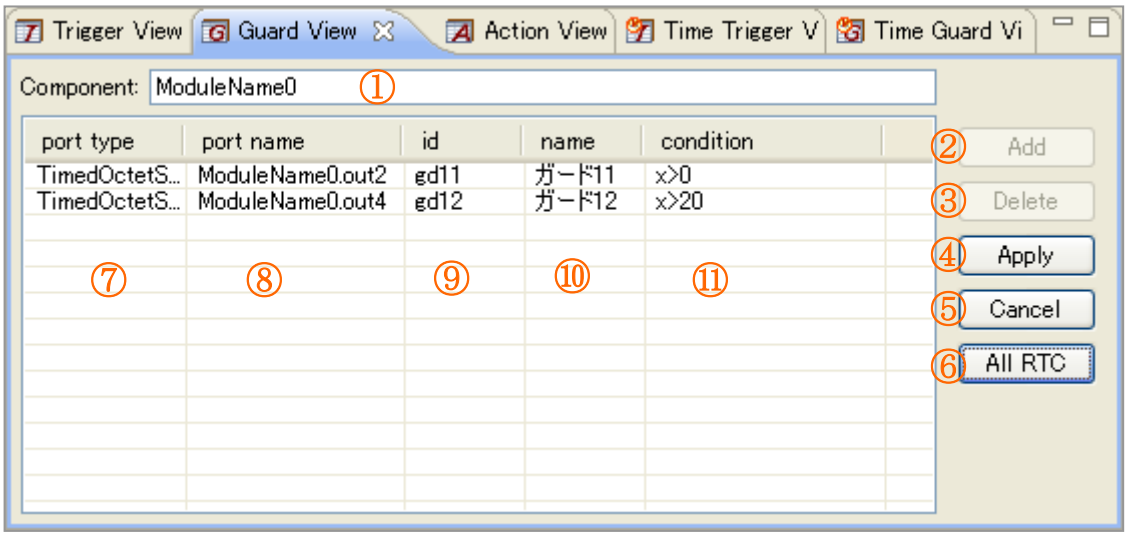


図 a-3-69 ガード定義ビュー

①では現在選択されている RT コンポーネントの名称を表示する. ②は選択中のポートにガードを追加し, ③で選

択中のガードを削除する. ④は現在編集中のガード定義内容を確定し, ⑤は逆に編集中のガード定義内容を破棄する. ⑥は RTC 一覧表示ダイアログを表示する.

⑦はポート種別を表示し, データ出力ポートを表す「Data Type」を表示する. ⑧はポート名称を表示する. ⑨はガード ID を定義する. 本項目は必須入力項目である. ⑩はガード名を定義する. ここで定義された名称が他の画面にて使用される. ⑪はガード定義本体を記述する.

「アクション定義ビュー」では, ネームサービスビューで選択中の RTC のサービスインターフェースに対して, アクションを定義する. RTC 選択時に条件に適合するサービスインターフェースが存在すると, インターフェースの一覧がテーブルに表示され, アクションを定義できるようになる. アクション定義ビューの画面構成を以下に示す.

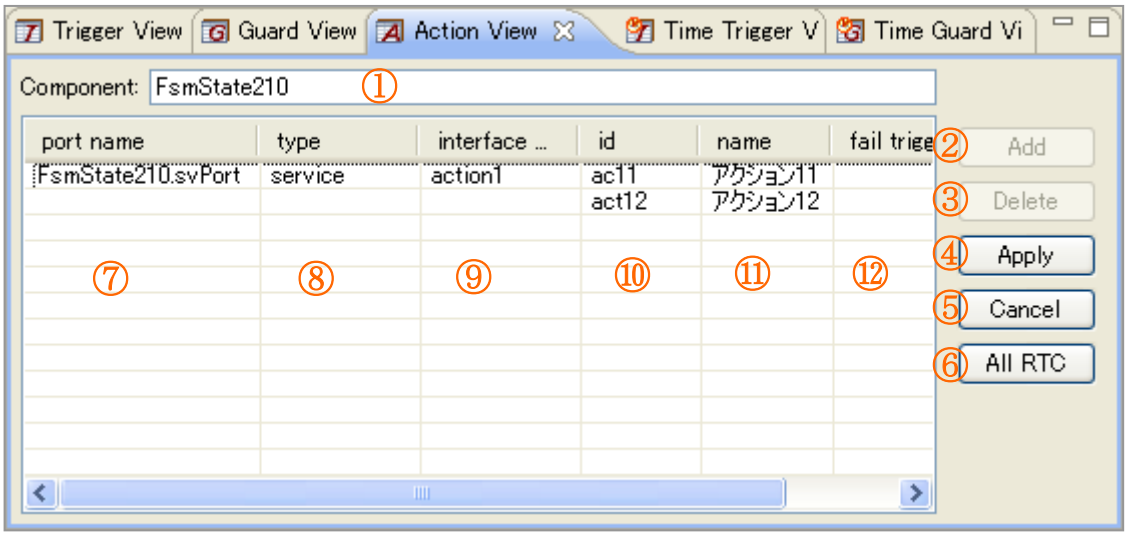


図 a-3-70 アクション定義ビュー

①では現在選択されている RT コンポーネントの名称を表示する. ②は選択中のインターフェースにアクションを追加し, ③で選択中のアクションを削除する. ④は現在編集中のアクション定義内容を確定し, ⑤は逆に編集中のアクション定義内容を破棄する. ⑥は RTC 一覧表示ダイアログを表示する.

⑦はポート名を, ⑧はポート種別をそれぞれ表示する. ポート種別がサービスインターフェースの場合は「service」、コマンドポートの場合はコマンド種別を表示する. ⑨はインターフェース名を表示し, ⑩はアクション ID を定義する. 本項目は必須入力項目である. ⑪はアクション名を定義する. ここで定義された名称が他の画面にて使用される. ⑫はアクション失敗時に起動するトリガ ID を指定する. アクションがエラーとなった場合にここで指定されたトリガを起動する.

「時間トリガ定義ビュー」では, 時間条件によるトリガを定義する. 時間トリガ定義ビューの画面構成を以下に示す.

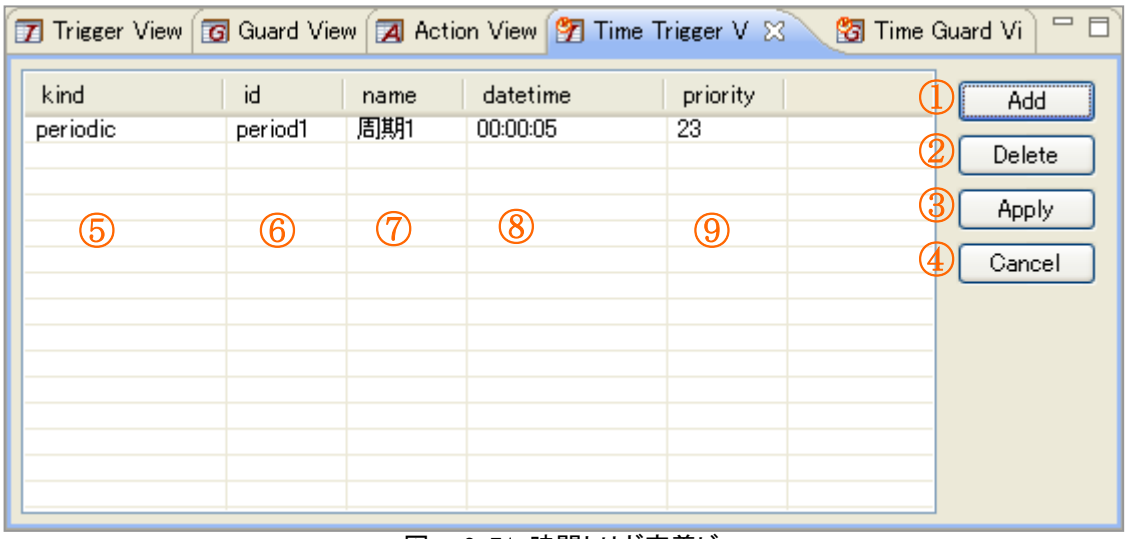


図 a-3-71 時間トリガ定義ビュー

①は時間トリガを追加し，②で選択中の時間トリガを削除する．③は現在編集中の時間トリガ内容を確定し，④は逆に編集中の時間トリガ定義内容を破棄する．

⑤は時間トリガ種別を選択する．時間トリガ種別は，periodic(周期実行)，everyDay(日毎定時実行)，everyMonth(月毎定時実行)，everyYear(年毎定時実行)の中からコンボボックスにて選択する．⑥はトリガ ID を定義する．本項目は必須入力項目である．⑦はトリガ名を定義する．ここで定義された名称が他の画面にて使用される．⑧は時間トリガ本体の時間条件を記述する．⑨はトリガの優先順位を定義する．複数トリガが同時に発生した場合，ここで指定した優先度に従って処理が実行される．

「時間ガード定義ビュー」では，時間条件によるガードを定義する．時間ガード定義ビューの画面構成を以下に示す．

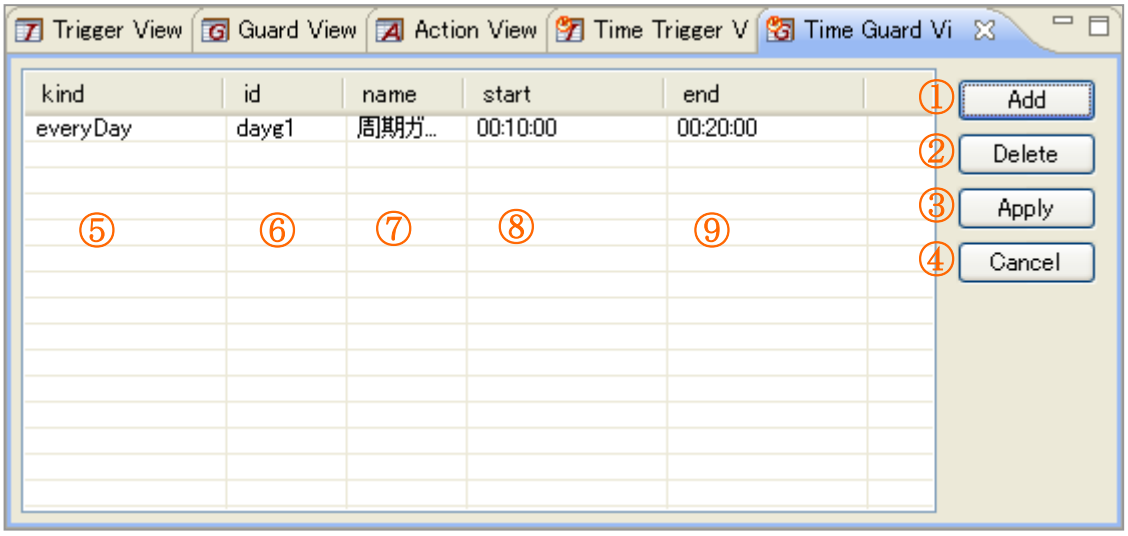


図 a-3-72 時間ガード定義ビュー

①は時間ガードを追加し，②で選択中の時間ガードを削除する．③は現在編集中の時間ガード内容を確定し，④は逆に編集中の時間ガード定義内容を破棄する．

⑤は時間ガード種別を選択する．時間ガード種別は，everyDay(日毎指定)，everyMonth(月毎指定)，everyYear(年毎指定)の中からコンボボックスにて選択する．⑥はガード ID を定義する．本項目は必須入力項目である．⑦はガード名を定義する．ここで定義された名称が他の画面にて使用される．⑧は時間ガード本体のうち開始時間に関する条件を記述し，⑨で終了時間に関する条件を記述する．

「インスペクタビュー」では，FSM エディタで構築する状態遷移図中の状態，遷移に対して属性を設定する．FSM エディタ中の要素を選択すると，要素の種別に応じて設定フォームを切り替える．インスペクタビューの画面構成を以下に示す．

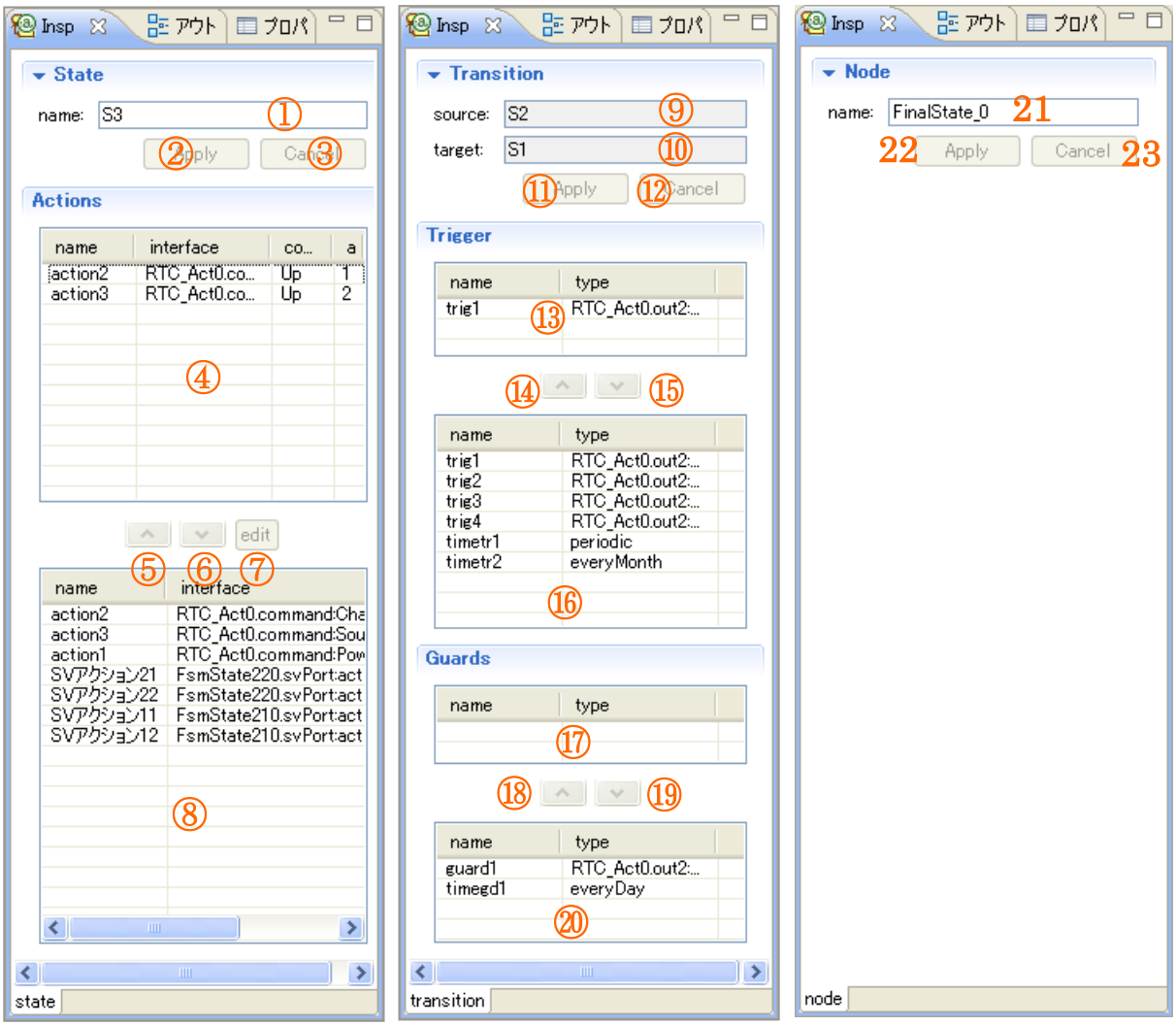


図 a-3-73 インスペクタビュー

左図は FSM エディタ上にて「状態」を選択した場合の表示である．①は状態の名称を示す．②で編集した各種属性を状態遷移図に反映させ，③は逆に編集中の各種属性の内容を破棄する．④は選択された状態に設定されているアクション一覧を示し，⑧で定義済みアクション一覧を表示する．⑤では定義済みアクション一覧から，現在選択中の状態にアクションを追加し，⑥は逆に現在選択中の状態に追加されたアクションを削除する．⑦はアクションがコマンドの場合の設定を行う．

中図は FSM エディタ上にて「遷移」を選択した場合の表示である．⑨は遷移元のノード名を，⑩は遷移先のノード名をそれぞれ表示し，⑪で変更中の各種属性の内容を状態遷移図に反映し，⑫で逆に編集中の各種設定内容を破棄する．⑬は選択されている遷移に設定されるトリガを表示し，⑯は定義済みのトリガ一覧を表示する．そして，⑭にて定義済みトリガから遷移に設定するトリガを選択し，⑮にて逆に遷移に設定したトリガを削除する．⑰は選択された遷移に設定されているガードを示し，⑳は定義済みのガード一覧を表示する．そして，⑱にて定義済みガードから遷移に設定するガードを選択し，⑲にて逆に遷移に設定したガードを削除する．

右図は FSM エディタ上にて「ノード」を選択した場合の表示である．21 にてノード名を設定するとともに，22 にて変更したノード名を状態遷移図に反映，逆に 23 にて編集中のノード名を破棄する．

「FSM エディタ」は，グラフィカルに状態遷移を定義するためのエディタであり，定義した状態遷移図から RTS の構築に必要な設定情報，接続情報を作成する． FSM エディタの画面イメージは以下に示す．

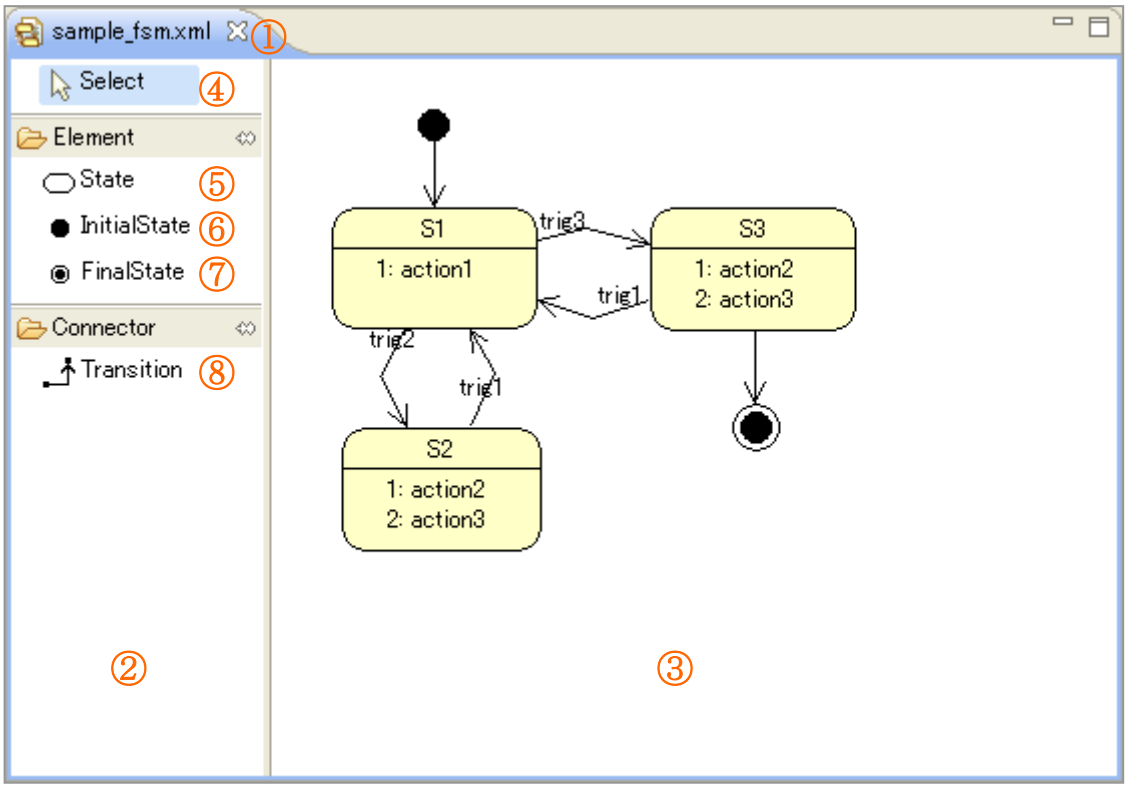


図 a-3-74 FSM エディタ

①は FSMProfile の名称を表す．状態遷移設定を保存，読み込みした場合にはプロファイル名を，新規作成する場合には，「FSMEditor」をそれぞれ表示する．②は描画要素の選択を行うパレット領域であり，③が描画領域である．

④を選択することで，描画領域が選択モードとなる．また，⑤にて「状態」の追加モード，⑥にて「開始ノード」の追加モード，⑦にて「終了ノード」の追加モード，⑧にて「遷移」の追加モードにそれぞれなる．なお，各要素は描画領域中の追加位置をクリックすることで要素が追加される．遷移を追加する場合には，遷移元，遷移先の順にノードをクリックする．また，開始ノードは状態遷移図中に1つのみ設定可能である．

本ツールでは RT システム全体の状態遷移を定義するとともに，定義した状態遷移に従う形で対象システムの制御を実行しているが，同様な仕組みは OMG にて標準化が行われている Robotic Technology Component Specification 内にて，Stimulus Response Processing パターンとして FSM コンポーネントという形で定義されている．そこで，本ツールでは，サービスポートを用いたアクションについては，まずは上記仕様中で定義されている FSM コンポーネントのインターフェースを持つコンポーネントを対象とした．以下に，FSM コンポーネントのインターフェース定義を示す．

```
module RTC
{
  typedef EXECUTION_HANDLE_TYPE_NATIVE ExecutionContextHandle_t;

  enum ReturnCode_t
  {
    RTC_OK,
    RTC_ERROR,
    BAD_PARAMETER,
    UNSUPPORTED,
    OUT_OF_RESOURCES,
    PRECONDITION_NOT_MET
  };

  interface FsmParticipantAction
  {
    ReturnCode_t on_action(in ExecutionContextHandle_t exec_handle);
  };

  interface FsmObject
  {
    ReturnCode_t send_stimulus(in string message, in ExecutionContextHandle_t exec_handle);
  };
```

一方，RT 要素部品(RT コンポーネント)化支援ツールの開発において，住宅システムに利用可能な RT コンポーネントの開発を支援するために ECHONET にて規定されている各種標準コマンドを送受信するためのインターフェース定義に対応した雛形コード生成機能も実現している．そこで，本ツールでは，この機能を用いて生成した ECHONET コマンド対応 RT コンポーネントについても，制御対象として設定できるようにした．

更に，本ツールで定義した内容を外部ファイルに保存する／読み込みを行うために，プロファイル情報(FSM プロファイル)を定義した．今回策定したプロファイルの XML スキーマを以下に示す．

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema                                                    xmlns:xsd=http://www.w3.org/2001/XMLSchema
xmlns:fsm="http://www.openrtp.org/namespaces/fsm"
   targetNamespace=http://www.openrtp.org/namespaces/fsm
   elementFormDefault="qualified" attributeFormDefault="qualified">

   <xsd:element name="FsmProfile" type="fsm:fsm_profile"/>
   <xsd:complexType name="fsm_profile">
      <xsd:sequence>
         <xsd:element name="StateMachine" type="fsm:state_machine" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="xsd:string" use="required"/>
   </xsd:complexType>

   <xsd:complexType name="state_machine">
      <xsd:sequence>
         <xsd:element name="State" type="fsm:state" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element name="Transition" type="fsm:transition" minOccurs="0" maxOccurs="unbounded"/>
         <xsd:element          name="ActionDefinition"          type="fsm:action_definition"          minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element         name="TriggerDefinition"         type="fsm:trigger_definition"         minOccurs="0"
maxOccurs="unbounded"/>
         <xsd:element          name="GuardDefinition"          type="fsm:guard_definition"          minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="version" type="xsd:string" use="required"/>
   </xsd:complexType>
   <xsd:complexType name="state">
      <xsd:sequence>
         <xsd:element name="Entry" type="fsm:entry" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="name" type="xsd:string" use="required"/>
      <xsd:attribute name="kind" use="optional">
         <xsd:simpleType>
            <xsd:restriction base="xsd:string">
               <xsd:enumeration value="normal"/>
               <xsd:enumeration value="initial"/>
               <xsd:enumeration value="final"/>
            </xsd:restriction>
         </xsd:simpleType>
```

```
    </xsd:attribute>
    <xsd:attribute name="left" type="xsd:integer" use="required"/>
    <xsd:attribute name="top" type="xsd:integer" use="required"/>
    <xsd:attribute name="right" type="xsd:integer" use="required"/>
    <xsd:attribute name="bottom" type="xsd:integer" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="entry">
    <xsd:sequence>
      <xsd:element name="OwnedOperation" type="fsm:owned_operation" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="owned_operation">
    <xsd:sequence>
      <xsd:element name="Action" type="fsm:action" minOccurs="1" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="action">
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="priority" type="xsd:integer" use="optional"/>
  </xsd:complexType>

  <xsd:complexType name="transition">
    <xsd:sequence>
      <xsd:element name="Trigger" type="fsm:trigger" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="Guard" type="fsm:guard" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element name="BendPoint" type="fsm:bend_point" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="source" type="xsd:string" use="required"/>
    <xsd:attribute name="target" type="xsd:string" use="required"/>
    <xsd:attribute name="sourceX" type="xsd:integer" use="required"/>
    <xsd:attribute name="sourceY" type="xsd:integer" use="required"/>
    <xsd:attribute name="endX" type="xsd:integer" use="required"/>
    <xsd:attribute name="endY" type="xsd:integer" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="trigger">
    <xsd:attribute name="id" type="xsd:string" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="guard">
    <xsd:attribute name="id" type="xsd:string" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="bend_point">
    <xsd:attribute name="posX" type="xsd:integer" use="required"/>
    <xsd:attribute name="posY" type="xsd:integer" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="action_definition">
    <xsd:sequence>
      <xsd:element name="targetInterface" type="fsm:target_interface" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="required"/>
    <xsd:attribute name="name" type="xsd:string" use="optional"/>
    <xsd:attribute name="failTrigger" type="xsd:string" use="optional"/>
  </xsd:complexType>

  <xsd:complexType name="target_component">
    <xsd:sequence>
    </xsd:sequence>
    <xsd:attribute name="componentId" type="xsd:string" use="required"/>
    <xsd:attribute name="instanceName" type="xsd:string" use="required"/>
    <xsd:attribute name="pathUri" type="xsd:string" use="required"/>
  </xsd:complexType>

  <xsd:complexType name="target_port">
    <xsd:complexContent>
      <xsd:extension base="fsm:target_component">
        <xsd:attribute name="portName" type="xsd:string" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

  <xsd:complexType name="target_interface">
    <xsd:complexContent>
      <xsd:extension base="fsm:target_port">
        <xsd:attribute name="interfaceName" type="xsd:string" use="required"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

```
    <xsd:complexType name="guard_source">
      <xsd:choice minOccurs="1" maxOccurs="1">
        <xsd:element name="DataSource" type="fsm:data_source"/>
        <xsd:element name="TimePeriod" type="fsm:time_period"/>
      </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="data_source">
      <xsd:sequence>
        <xsd:element name="SourcePort" type="fsm:target_port" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="time_period">
      <xsd:attribute name="kind" type="xsd:string" use="required"/>
      <xsd:attribute name="start" type="xsd:string" use="required"/>
      <xsd:attribute name="end" type="xsd:string" use="required"/>
    </xsd:complexType>

    <xsd:complexType name="trigger_definition">
      <xsd:sequence>
        <xsd:element name="TriggerSource" type="fsm:trigger_source" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:string" use="required"/>
      <xsd:attribute name="name" type="xsd:string" use="optional"/>
      <xsd:attribute name="priority" type="xsd:integer" use="optional"/>
    </xsd:complexType>

    <xsd:complexType name="trigger_source">
      <xsd:choice minOccurs="1" maxOccurs="1">
        <xsd:element name="DataTrigger" type="fsm:data_trigger"/>
        <xsd:element name="EventTrigger" type="fsm:event_trigger"/>
        <xsd:element name="TimeTrigger" type="fsm:time_trigger"/>
      </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="data_trigger">
      <xsd:complexContent>
        <xsd:extension base ="fsm:data_source">
          <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="Constraint" type="fsm:constraint"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="event_trigger">
      <xsd:complexContent>
        <xsd:extension base ="fsm:data_source">
          <xsd:sequence minOccurs="1" maxOccurs="1">
            <xsd:element name="Message" type="fsm:message"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="time_trigger">
      <xsd:sequence minOccurs="1" maxOccurs="1">
        <xsd:element name="TimeEvent" type="fsm:time_event"/>
      </xsd:sequence>
    </xsd:complexType>

    <xsd:complexType name="constraint">
      <xsd:attribute name="body" type="xsd:string" use="required"/>
    </xsd:complexType>

    <xsd:complexType name="time_event">
      <xsd:attribute name="kind" type="xsd:string" use="required"/>
      <xsd:attribute name="body" type="xsd:string" use="required"/>
    </xsd:complexType>

    <xsd:complexType name="message">
      <xsd:attribute name="body" type="xsd:string" use="required"/>
    </xsd:complexType>

</xsd:schema>
```

(a-4) RT 要素部品管理モジュール用組み込み RT ミドルウェアの研究開発
(委託先:独立行政法人産業技術総合研究所)
a-4-1 概要
本開発項目の課題は以下の通りである。
① OpenRTM-aist の組込み MPU 向け RT ミドルウェアの開発
従来の RT ミドルウェアは Linux や Windows 上といった高スペックな PC で動作しており、住宅設備等で使われている組込系 MPU で動作することは難しかった。そこで、組込 OS 上で動作しかつ、通常の RT ミドルウェアに参加可能とする組込 MPU 向け RT ミドルウェアを現在の RT ミドルウェアをベースとして開発した。

② 住宅システムにおけるシステム設計ならびに、それによって構築されたシステムのスループット等の評価
住宅の設計は基本的に住宅メーカーが行うが、住宅内の設備については、様々なメーカーの設備を組み合わせて構築している。すなわち、住宅内に RT システムを導入する上で、ネットワークプロトコルの共通化は重要な項目である。また、住宅のシステムは、24 時間、365 日安定に動作することが要求され、たとえ停電でシステムが停止したとしても、自動で復帰する機能が必要とされる。以上の要求仕様に対するシステム設計ならびに、構築されたネットワークを介したシステムの安定性に係る機能の評価を行った。

③ 住宅用実証 RT システムの構築並びに住宅システムとしての評価
産業技術総合研究所の実験室内ある住宅モデル、およびミサワホーム展示場のモデルハウス内に開発した RT システムを導入した。産業技術総合研究所内住宅モデルには、全てのシステムが導入され、ミサワホーム展示場のモデルハウスには、インテリジェント空調システムの一部が導入された。これらの実際の環境を模擬した建物に導入し、建物全体としてのシステムの評価を行った。

以上研究課題に対する成果および達成度を表 a-4-1 にまとめる。

表 a-4-1 目標に対する研究開発成果および達成度

目標	研究開発成果	達成度
RT 要素部品管理モジュール用組み込み RT ミドルウェアの研究開発		
(1)OpenRTM-aist の組込み MPU 向け RT ミドルウェアの開発	(1)TOPPERS 版 OpenRTM-aist として SH2 および ARM といった組込み MPU ボードに実装し動作確認を行った。	(1)達成
(2)住宅システムにおけるシステム設計ならびに、それによって構築されたシステムのスループット等の評価	(2)ホームコントローラの集中制御ではなく、その下部に位置する要素部品管理モジュールが各設備に設置する基盤通信モジュールを管理し、その基盤通信モジュールに RT コンポーネントが動作することで、分散処理系を構成し、住宅システムの安定性を向上させ、動作検証を行った。詳細は c-1-6-4-c 節に記載。	(2)達成
(3)住宅用実証 RT システムの構築並びに住宅システムとしての評価	(3)産業技術総合研究所の実験室内ある住宅モデル、およびミサワホーム展示場のモデルハウス内に開発した RT システムを導入し、システム全体として①RT システムとしての動作確認、②RT 要素部品のプラグアンドプレイ機能確認、③PLCにおける基盤通信モジュール間のネットワーク動作、④住宅システム内の RT 要素部品の安定動作、⑤通信モジュールの待機消費電力の評価、以上の 5 つの評価を行った。 詳細は c-1-6-4-d 節に記載	(3)達成

a-4-2　OpenRTM-aist の組込み MPU 向け RT ミドルウェアの開発

・TOPPERS 版 OpenRTM-aist の SH2 への移植
従来 TOPPERS 版 OpenRTM-aist が動作していた ARM9 等よりもさらに省スペックで小型・安価な CPU 上での動作を目指し、TOPPERS 版 OpenRTM-aist を MMU 非搭載の SH2 への移植を行った（図 a-4-1）。

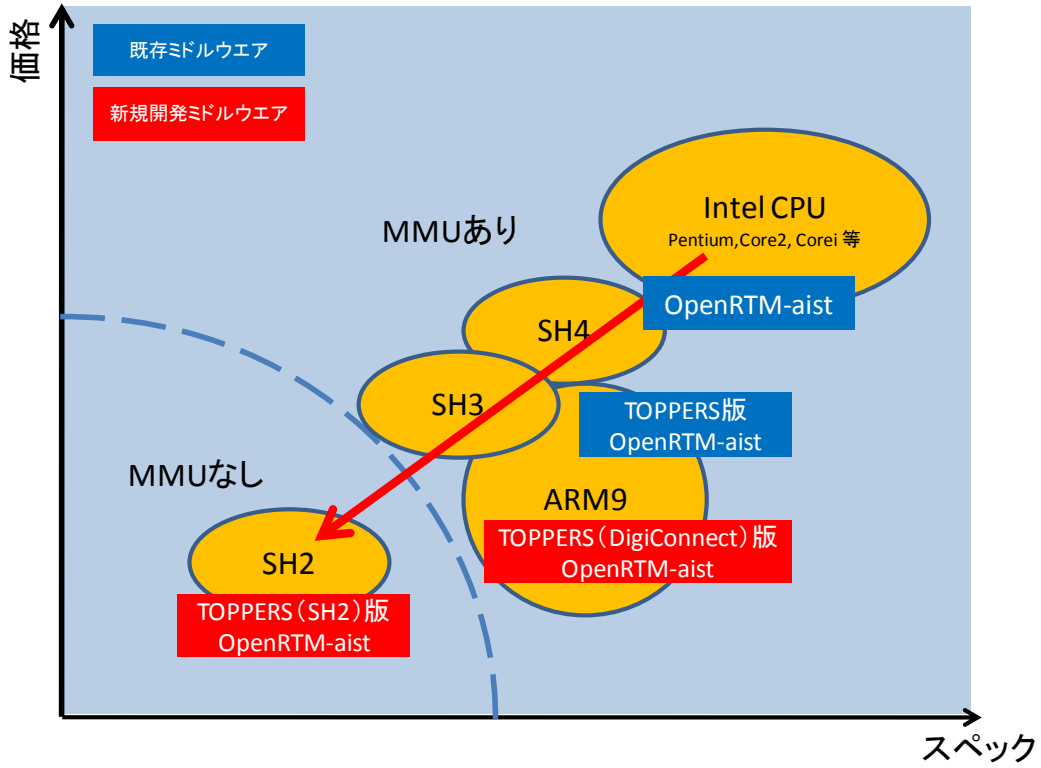


図 a-4-1　従来ミドルウエアと今回の開発ターゲット

Alpha Project 社製 SH2 ボード（図 a-4-2、表 a-4-2）上で動作するように OpenRTM-aist を移植し、同 CPU ボード上で RT コンポーネントが動作、AD ボード（表 a-4-3）等を用いて入出力が行えることを確認した。これにより、これまで MMU 搭載 CPU のみでしか動作しなかった OpenRTM-aist を、今後は MMU 非搭載の低スペック、安価な CPU でも動作が可能であることが示された。

図 a-4-2　アルファプロジェクト SH2 CPU ボード（μ ST-SH2）外観

表 a-4-2 アルファプロジェクト SH2 CPU ボード（μ ST-SH2） 仕様

CPU	ルネサス エレクトロニクス(株)社製 R4S76190B125BGV（SH7619）
クロック	14.7456MHz 発振子 CPU コア：117.9648MHz, バス：58.9824MHz, 周辺 29.4912MHz
メモリ	FlashROM： 8MB, SDRAM： 32MB
I/O ポート	汎用入出力端子 17 本
Ethernet	10/100BaseTX（CPU 内蔵） 1 ポート
シリアル I/F	3ch（クロック同期／調歩同期モード、2ch は兼用端子）
SD/MMC スロット	1 スロット（SPI モード）
RTC	S-35190A（SII 製） 電気二重層コンデンサにてバックアップ
外部拡張バス	40 ピン×2（バス、I/O、H-UDI）
リセット	リセット IC、リセット SW 搭載 外部拡張コネクタ（未実装）からのリセット可能
電源	5V ±10%
消費電流	Max 約 400mA
動作温度範囲	0℃～50℃
基板寸法（mm）	71×51（4 層基板）

表 a-4-3 アルファプロジェクト AD・DIO ボード（μ ST-ADIO）仕様

アナログ入力	シングルエンド入力 4ch 差動入力 2ch
アナログ入力レンジ	バイポーラ±5V ユニポーラ 0V～＋5V
入力インピーダンス	940KΩ
分解能	10/8bit（ソフトウェアにて設定）
AD コンバータ	ADC10154
A/D 変換時間 *1	4.5μ s（最大）
A/D 変換誤差	1%以内
デジタル入力	フォトカプラ絶縁入力 3ch（動作電圧範囲 0V～＋24V）
デジタル入力遅延	約 20μ s
デジタル出力	フォトカプラ絶縁出力 3ch（最大定格電圧 ＋80V） オープンコレクタ出力
デジタル出力遅延	約 20μ s（470Ω プルアップ時）
デジタル出力最大駆動電流	25mA
絶縁耐圧	3.75kV
端子台	ML-700-NH20（サトーパーツ）
CPLD	XC9572XL-10VQ64C（Xilinx）
バス接続	拡張コネクタ 40pin×2 HIF-40PB-2.54DSA（ヒロセ）
消費電流	Max 約 200mA
動作温度範囲	0℃～50℃

さらに小型の CPU 上で動作の確認を試みた。Digi 社の小型デバイスサーバ DigiConnectME（図 a-4-3、表 a-4-4）上での動作を試みるため、まず TOPPERS/ASP を DigiConnectME 上へ移植し動作することが確認できた。

図 a-4-3　Digi 社、Digi Connect ME 9210 外観

表 a-4-4　Digi 社、Digi Connect ME 9210　仕様

CPU	32 ビット Digi NS9210 プロセッサ@75MHz(ARM926EJ-S) 内蔵 256 ビット AES アクセラレータ
入出力	フレキシブルインタフェースモジュール(FIM) 300MHz DRPIC165X CPU 2kbyte プログラム／192byte データ RAM
内蔵メモリ	2/4 MB NOR Flash 8MB SDRAM
シリアル入出力	高速 TTL シリアルインタフェース TXD, RXD,RTS, CTS, DTR, DSR, DCD
シリアルペリフェラル	マスタデータレート 最大 33.3Mbps スレーブデータレート 最大 7.5Mbps
I2C	v1.0 バスインタフェース 7 ビットおよび 10 ビットアドレスモード
GPIO ポート	10 ビット，最大 3 つの拡張 IRQ オプション搭載 1%以内
パワーマネジメント	オンザフライ・クロックスケーリング、低電力スリープモード、コンフィグラブルスケーリング／ウェイクアップ イベント (EIRQ, UART，イーサネットなど)
ネットワークインタフェース	物理層 : 10/100Base-T データレート : 10/100Mbps(オートセンシング) モード : 全二重または半二重(オートセンシング) コネクタ : RJ-45 (magnetics 付属) 802.3af パワーパススルー ミッドスパンおよびエンドスパン
消費電流	3.3V DC@346mA typical (1.14W) ロースピード・アイドルモード : 3.3V DC@186mA (613mW) スリープモード : 3.3V DC@34mA (113mW)
動作温度範囲	-40～85℃

・ 移植・変更内容一覧

TOPPERS/ASP をターゲット実機 Digi Connect Me 9210 に移植を行った。TOPPERS/ASP では、ARM 向けに既に移植が行われており、ARM7TDMI をコアに使用している AT91SAM7S に対応している。これをベースに移植を行った。

ハードウェア情報は以下である。

表 a-4-5 ハードウェア構成表

	Digi Connect Me 9210
プロセッサファミリー	ARM9TDMI
アーキテクチャ	v5TEJ
コア	ARM926EJ-S
キャッシュ	4KB/4KB, MMU
プロセッサクロック	最大 75MHz
プロセッサ	NS9210
SDRAM	8MB
FLASH	4MB

移植に伴い、以下の対応を行った。

TOPPERS/ASP

- Digi Connect Me 9210 用の開発環境構築
- プロセッサ向けのカーネルコード作成
- ターゲットボード向けのカーネルコード作成
- 各種ドライバ作成
- ブートローダの TOPPERS/ASP への対応

TINET

- イーサネットドライバの新規作成

・ 開発環境

開発環境についての情報を以下に記述する。

表 a-4-6 開発環境一覧

項目	内容	備考
OS(PC 側)	Window2000、WindowsXP	
コンパイラ	GCC バージョン 4.2.0	Digi 製 開発ツールに同梱
デバッガ	GDB バージョン 6.1	開発ツールに同梱
JTAG ツール	J-Link	開発ツールに同梱
その他ツール	Cygwin バージョン 1.5.25	開発ツールに同梱
	Eclipse バージョン 3.1.2	開発ツールに同梱

・ RtORB 移植内容詳細

Armadillo 版の OpenRTM-aist1.0RC の開発で対応した RtROB をベースに移植を行った。

Armadillo 版との違いは以下である。

- エンディアンがビッグエンディアン。(Armadillo 版はリトルエンディアン)

SH 版の OpenRTM-aist0.4.2 の開発時に対応した RtORB を参考にしてビッグエンディアンに対応した。上記以外の修正は以下である。

- デバッグマクロの使用箇所の修正(#if DEBUG → #ifdef DEBUG)

DEBUG の切り替えが正しく動作しないために修正した

- 構造体パッキングのずれ対応
- IOR 追加対応
- POA アクティブ API 追加対応

・ 今後の展望

DigiConnectME 上の TOPPERS に OpenRTM-aist の移植を試みたが、DigiConnectME の搭載メモリが少ないため OpenRTM-aist を搭載することができなかった。DigiConnectME レベルの CPU 上で RTC を動作させるには、将来的に C 言語等で RT ミドルウエアを実装し、より省メモリのミドルウエアおよび RT コンポーネントフレームワークが必要であることが明らかとなった。

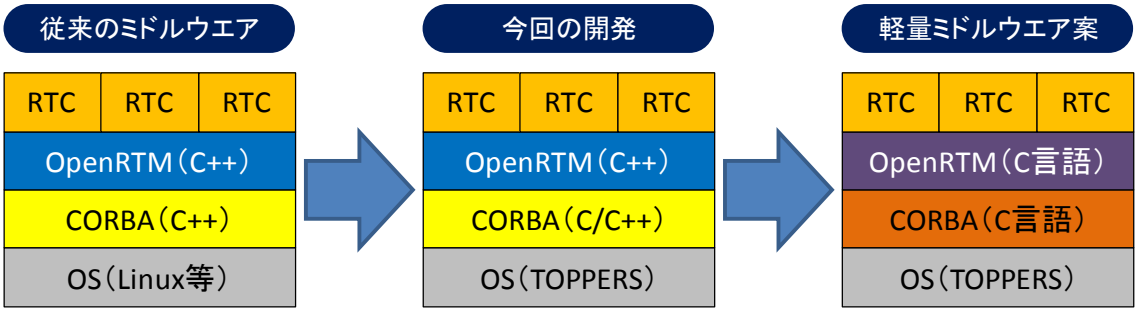


図 a-4-4　今回開発した軽量ミドルウエアとさらなる軽量化案

・ OpenRTM-aist-1.0 対応

現行の RT ミドルウエアは OpenRTM-aist-1.0 であるが、TOPPERS 版 OpenRTM-aist はバージョン 0.4 のみであり、互換性の問題から 1.0 系の RTC と相互運用することはできない。そこで、TOPEPRS 版 OpenRTM-aist を OpenRTM-aist の新バージョン 1.0 に対応させた。これにより、TOPPERS を搭載した小型 CPU 上で、OpenRTM-aist-1.0 が動作し、かつ他の 1.0 版 RT コンポーネントと協調動作を実現した。アットマークテクノ社の Armadillo シリーズ 210,220,240 での動作を確認した。

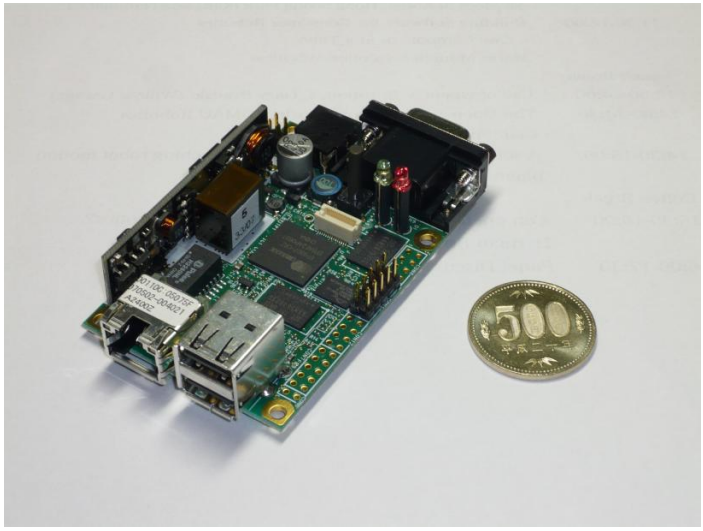


図 a-4-4　Armadillo240 外観

表 a-4-7 Armadillo240 仕様

プロセッサ	EP9307 (Cirrus Logic)
CPU コア	ARM920T
システムクロック	CPU コアクロック：200MHz
SDRAM	64MB
FLASH	8MB (NOR) NAND フラッシュメモリ搭載可能（オプション）
無線 LAN	–
Ethernet	10BASE-T/100BASE-TX (PoE 対応可能)
シリアルポート	TTL 2 ポート
汎用入出力	16 ビット + SW 入力 x1
USB	2.0 Full Speed (12Mbps) 2 チャンネル Type A コネクタ
画像出力	コネクタ形状：D-sub15 ピン (ミニ) 最大解像度 1024x768
外形サイズ	基板：75.0 x 50.0〔mm〕 ケース：W83.0 x D58.0 x H24.3〔mm〕
電源	DC5V の外部電源 または Power over Ethernet（Type-A/B）による電源供給（オプション）
消費電力(Typ.)	1.5W
使用温度範囲	0 ~ 60 ℃

OpenRTM-aist-1.0 の Linux 版を TOPPERS/ASP 版へ移植する為、以下の対応を実施

- OpenRTM 起動方法の TOPPERS/ASP 対応の為の変更
- TOPPERS 版 COIL の実装
- TOPPERS/ASP に対応する為、標準出力、標準エラー出力処理の対応
- ターゲットボード上で動作させる為、動的モジュールロード機能の削除
- TOPPERS/ASP に対応する為、動的生成を静的にリソースを受け渡すように変更
- TOPPERS/ASP に対応する為にログファイルへの出力を標準出力に出力するよう変更
- ターゲットボード上で動作させる為、コンフィグファイル(rtc.conf)からフラッシュメモリに変更
- 軽量 CORBA（RtORB）の使用により対応しないコーディングの変更

・ COIL への対応

OpenRTM-aist1.0 は COIL を使用しているが、TOPPERS/ASP に対応する為に、TOPPERS 版 COIL を実装した。POSIX 版の COIL をベースに TOPPERS 版の COIL を実装。

➢ TOPPERS 未対応機能の削除対応

ファイル名	変更内容
DynamicLib.cpp,DynamicLib.h	動的ロード未対応。 コンパイル通らない箇所を削除した。
Signal.cpp,Singal.h	シグナル未対応。 コンパイル通らない箇所を削除した。

➢ TOPPERS 向けの機能の変更対応

ファイル名	変更内容
Mutex.cpp,Mutex.h	pthread_mutex の代わりに TOPPERS のセマフォを使用するように変更。
Properties.cpp,Properties.h	プロパティ情報の保存機能削除。 プロパティ情報の読込み方法をフラッシュ ROM のアドレスポインタから読み込むように変更。
stringutil.cpp,stringutil.h	➢ 行文字列の取得処理の変更。 フラッシュ ROM のアドレスポインタから取得するように変更。 ➢ sprintf の最大バッファ数の変更 1024→512
Task.cpp,Task.h	TOPPERS 用のタスク管理クラスの実装。 予めプールしているタスクリソースリストからタスクリソースを取得するように変更。
Time.cpp,Time.h	TOPPERS 版 OpenRTM-aist 用のスリープ API 追加 sleep_until()
UUID.cpp,UUID.h	TOPPERS 版 OpenRTM-aist 用の UUID 生成処理の実装。 UUID のライブラリを利用しないように、独自に UUID を生成する処理を実装。
Condition.h	unix 用の pthread_cond が TOPPERS には存在しないため、代わりにセマフォを使用するように変更。
Factory.h	Windows 向けの DLL export マクロ宣言の削除。 TOPPERS では使用しないため。
File.h	unix 用の dirname が TOPPERS には存在しないため、ディレクトリ名取得 API(dirname)の実装の追加。
Os.h,Os.cpp（追加）	➢TOPPERS では使用しない変数を使わないように変更。 ➢TOPPERS 用のシステム名称取得 API の実装

➢ TOPPERS 向けの機能の追加対応

ファイル名	追加内容
String.cpp,String.h	TOPPERS 版 OpenRTM-aist 用の printf 関数の追加
toppers_task.c,toppers_task.h, toppers_task.cfg	TOPPERS 版 OpenRTM-aist 用のタスクリソースを管理する処理の実装。 あらかじめプールしているタスクリソースリストより、タスクリソースを取得する API の追加実装。
toppers_semph.c,toppers_semph.h, toppers_semph.cfg	TOPPERS 版 OpenRTM-aist 用のセマフォリソースを管理する処理の実装。 あらかじめプールしているセマフォリソースリストより、セマフォリソースを取得する API の追加実装。

・ OS リソースの管理について

TOPPERS は、必要となる OS リソース（タスク、セマフォ等）はコンパイル前に静的に決定する必要がある。本

実装では、予めリソースをリソースリストの形式で用意し、必要なときにリストからリソースを取得する、必要が無くなったときにリストに返す処理を実装した。このような対応を行うことで、OpenRTM-aist のソースコードを変更することなく TOPPERS 版の実装をすることができる。

・ 標準入出力、標準エラー出力の対応

TOPPERS 版 OpenRTM-aist では標準入出力、標準エラー出力に関して、ソースコードの変更は行っていない。以下のような呼び出しは TOPPERS 版 OpenRTM-aist でも有効である。

```
std::cout << nv[i].name << ": not a string value" << std::endl;
```

上記の呼び出しが有効となるように低レベル API の実装を行った。具体的は、newlib の API である _read(),_write()関数において、シリアルの入出力処理を実装した（toppers_asp/asp/pdic/syscalls.c にて実装）。

・ ログファイル出力対応

ログファイルへの出力は、TOPERS 版 OpenRTM-aist はファイルシステム対応していない。そのため、ファイルへの出力ではなく、標準出力を行うようにし、シリアルからログを出力するように変更した。実装方法は、標準入出力、標準エラー出力の対応の対応内容と同様。

・ 設定ファイルの書込み対応

Armadillo-210/240 上で動作させる為に、設定ファイルへの設定の保存を、フラッシュメモリに対し書き込みを行うようにする為、設定プログラムの追加を行う。なお、この処理は、TOPPERS／ASPにて、Armadillo-210/240 のジャンパーJP2 がショート状態での起動時に実行される。フラッシュメモリの 0x603f0000 から OprnRTM の設定パラメータを書き込む

書き込みを行う設定データは以下の 7 データである。

"corba.endpoint"	:複数 NIC 使用時の使用 I/F の指定
"corba.nameservers"	:ネームサーバーの IP アドレス及びポート番号
"naming.formats"	:RTC のの命名規則
"logger.enable"	:ログ出力の有効／無効
"logger.log_level"	:ログ出力レベル
"exec_cxt.periodic.rate"	:周期実行タスクの起動間隔
"timer.tick"	:タイマー間隔

設定ファイルをフラッシュメモリに書込みを行う為に以下のファイルを追加。

- OpenRTMSetup.h：フラッシュメモリへの書込みを行う処理の初期値や、関数の定義を行う。
- OpenRTMSetup.C：フラッシュメモリへの書込みを行う処理のソースコード。

関数名	処理概要
is_ipaddr_str	IP アドレスチェック関数 パラメータ（"corba.endpoint"）、パラメータ（"corba.nameservers"）にて使用。入力文字が '0' ～ '9' 、':' 、'.' ならば正常、前述以外の文字ならば異常と判定する。
is_enable	入力文字列判定関数 パラメータ（"logger.enable"）にて使用。 入力文字列が"YES"、"NO"の何れか以外の場合、異常と判定する。
is_log_level	入力文字列判定関数 パラメータ（"logger.log_level"）にて使用。 入力文字列が "SILENT" 、"ERROR" 、"WARN" 、"INFO" 、"NORNAL" 、"DEBUG" 、"TRACE" 、"VERBOSE" 、"PARANOID" 、"MANDATORY"の何れか以外の場合、異常と判定する。
is_number	数値チェック関数 パラメータ（"exec_cxt.periodic.rate"）、パラメータ（"timer.tick"）にて使用。

関数名	処理概要
	入力文字が、'0' ～ '9' か、'.' ならば正常、前述以外の文字ならば異常と判定する。
s_putc	1 文字出力関数 文字の値が'¥n' で有った場合、'¥r' を前に付加した後、出力を行い、その他の文字はそのまま出力。
s_puts	1 行出力関数 文字列の中に'¥n' が有った場合、'¥r' を前に付加した後、出力を行い、その他の文字は、そのまま出力する。'¥0' が出力されるまで順次文字出力を行う。
s_copy_default	引数にセットされているパラメータ("corba.endpoint","corba.nameservers","naming.formats","logger.enable","logger.log_level","exec_cxt.periodic.rate","timer.tick")に対する初期値を判定し、呼び出し元の関数に返す。
s_strcmp	2つの文字列を比較し、1文字でも違っていれば、1を呼び出し元の関数に返し、全て同じ文字列ならば、0を呼び出し元関数に返す。
s_strncmp	2つの文字数を指定の長さ分比較を行い、1文字でも違っていれば、1を呼び出し元の関数に返し、全て同じ文字列ならば、0を呼び出し元関数に返す。
s_strlen	引数にセットされている文字列の長さを呼び出し元の関数に返す。
s_strcpy	2番目の引数にセットされている文字列を、1番目の引数にセットされている文字列の変数にセットする。
s_strncpy	2番目の引数にセットされている文字列を、引数でセットされている長さ分だけ1番目の引数にセットされている文字列へセットする。
OpenRTMSetup	・フラッシュメモリの 0x603f0000 番地から"OpenRTM"と書かれているかチェックを行い、書かれている場合には、フラッシュメモリに既に設定値が有ると判断し、"OpenRTM"と書かれていない場合には、フラッシュメモリには設定値が無いと判断する。 ・フラッシュメモリの 0x603f0000 番地へ"OpenRTM"と書きこむ ・シリアルポートに対し、"corba.endpoint","corba.nameservers","naming.formats","logger.enable","logger.log_level","exec_cxt.periodic.rate","timer.tick の順番にパラメータと現在フラッシュメモリに書かれている値を表示し入力を行う。未設定パラメータの場合はデフォルトの表示を行う。 ・フラッシュメモリ上に既に設定値が有る場合で、パラメータの入力が行われなかった場合、現在、フラッシュメモリ上に有る値を入力値へコピーする。 ・フラッシュメモリに設定値が無く、入力も行われなかった場合にはデフォルト値を入力値にセットする。 ・全てのパラメータを、フラッシュメモリへ書き込む。 ・書き込み完了後、シリアルポートに対し、"complete!"と出力を行う。
flash_writer	・フラッシュメモリの 0x603f0000 番地から 2000hバイト分0クリアを行う。 ・全設定値をフラッシュメモリの 0x603f0000 番地から書き込む
checkMagicWordForFlash	・フラッシュメモリの 0x603f0000 番地からメモリの値を読み込み、"OpenRTM"と書かれているかチェックする。 ・"OpenRTM"と書かれている場合には、1を返す。 ・"OpenRTM"と書かれていない場合には、0を返す。
setMagicWord	・設定値を格納する先頭に"OpenRTM"とセットする。
getNameServerAddr	・フラッシュメモリの 0x603f0000 番地からメモリの値を読み込み、"OpenRTM"と書かれているかチェックする。 ・"OpenRTM"と書かれていない場合には、処理を行わず終了 ・フラッシュメモリ内から NameServer のパラメータ"corba.nameservers"を検索し読み込む。 ・NameServer のパラメータに続く、NameServer の設定値をフラッシュメモリから読み込む。

関数名	処理概要
	・NameServer の設定値の先頭1文字目が'¥n'ならば、処理を終了 ・NameServer の設定値の先頭1文字目が'¥n'以外ならば、設定値に'¥n'か'¥0'が入ってくるまで、フラッシュメモリから読み込みを行い、呼び出し元関数へ設定値を渡す。

・ 軽量 CORBA（RtORB）対応

OpenRTM-aist1.0 の Linux 版では、omniORB を使用している。軽量 CORBA（RtORB）では対応していない実装があり、実装内容を変更した。また、CDR は対応していないため、新規に RtORB 版の CDR を実装した。以下の実装変更を実施した。

- NVList 型から NVList_ptr 型への変更
- NamingContext のオブジェクト参照方法の変更
- SDOPackage::NameValue newNV(const char* name, Value value)のパラメータ Value 型の対応
- 未対応クラス ServantAlreadyActive
- Object の in メソッド未対応
- CdrData のインスタンス生成方法の変更
- RtORB が対応していない参照

・ 分散センサアプリケーション例

具体的なアプリケーションとして、Armadillo240（今回開発した TOPPERS 版 OpenRTM-aist-1.0）に接続されたレーザレンジセンサからのデータを PC 上で動作する OpenRTM-aist-1.0 Python 版のレーザ測域データビューアに表示させる分散センサシステムを開発した（図 a-4-6）。組込機器上では TOPPERS 版 OpenRTM-aist の RT コンポーネントを動作させ、統合アプリケーションは通常の OpenRTM-aist を用途に応じて言語を使い分けシステムを構築することが可能となった。

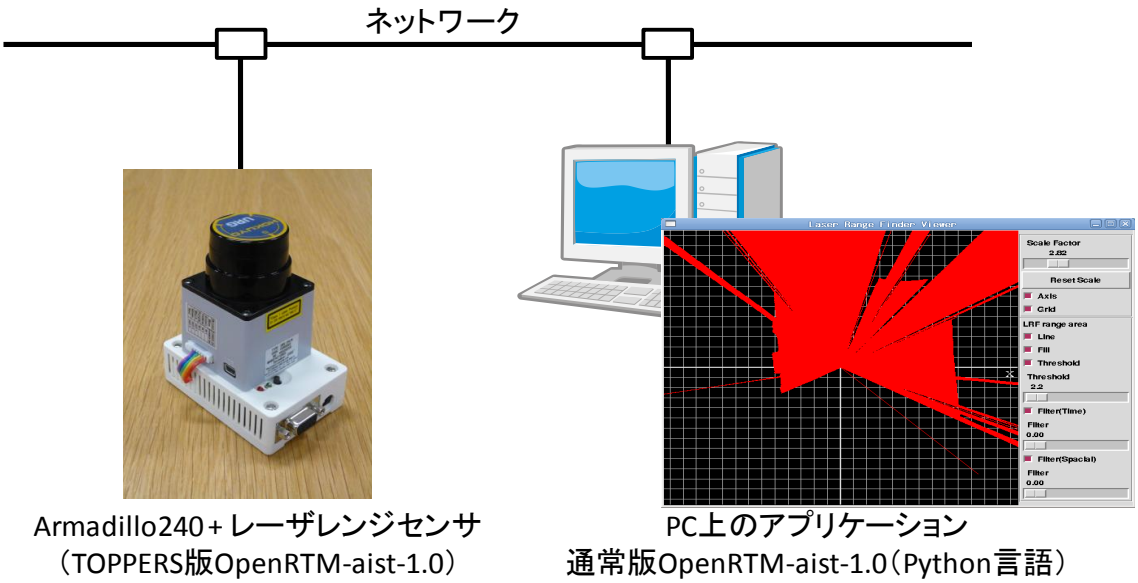


図 a-4-6 TOPPERS 版 OpenRTM-aist-1.0 を用いた分散センサアプリケーションの例

a-4-2 住宅システムにおけるシステム設計ならびに、それによって構築されたシステムのスループット等の評価

住宅のシステムは、24 時間、365 日安定に動作することが要求され、たとえ停電でシステムが停止したとしても、自動で復帰する機能が必要とされる。そのため、通常ロボットシステムで利用される PC 一台の集中管理システムでは、その PC にトラブルが起こった場合、全ての家の機能が失われることになる。そこで、PC（ホームコントローラ）だけでなく、各設備においても RT コンポーネントを動作させるといった分散処理系を採用している。詳細については、c-1-6-4-c 節の構築した実証住宅システムのシステム構成の報告の中にまとめた。

a-4-3 住宅用実証 RT システムの構築並びに住宅システムとしての評価

本事業では住宅 RT システムの実証先として、産業技術総合研究所の実験室内ある住宅モデル、およびミサワホーム展示場のモデルハウスを対象とした。産業技術総合研究所内住宅モデルには、開発した要素基板管理モジュールおよび基盤通信モジュールを介して住宅設備が RT 化された RT 要素部品のすべてがが導入され、ミサワホーム展示場のモデルハウスには、インテリジェント空調システムの一部が導入された。本件では、実際の住宅設備に導入された際の、建物全体としてのシステムの評価を行った。詳細は、c-1-6-4-d 節の構築した実証住宅システム全体の結果と評価の報告内にまとめた。

(b)「基盤通信モジュールを用いた RT 要素部品の開発」
(b-1) 小型通信ドライバモジュールと小型リニアアクチュエータによるRT要素部品の開発
(委託先:THK 株式会社)
b-1-1 概要
小型通信ドライバにおけるシステム全体に関する設定した目標と達成度を以下に示す。
・ 「基盤通信モジュール」、又は「共通基盤モジュール」と組み合わされている。これらは要素部品と一体化されていることが望ましいが、処理部として分離されても良い。
⇒ ○達成
:基盤通信モジュールと組み合わせてインテリジェントウインドウシステムを構築

・ RTシステムから OpenRTM 仕様に基づきRTコンポーネントとして利用できる。
⇒ ○達成
:小型通信ドライバモジュールに miniRTCc を実装して仮想 RTC 化

・ 開発したRT要素部品及びRTミドルウェアにて動作させる際に必要となる項目を記載した仕様書及び取扱説明書を、RTシステム開発機関が利用できる形で提供する。
⇒ ○達成
➢ 小型通信ドライバモジュールの仕様書(別紙「SEED 仕様書」)

次に、実施計画書に沿って設定した目標と達成度を以下に示す。

① RTC－Lite フレームワークに準拠した小型通信ドライバモジュールの開発
(ア) 小型の CAN による通信機能を有した小型モータ向けモータコントローラドライバのラインナップを、各モータタイプにおいて開発する
⇒ ○達成
:DC モータ、DC ブラシレスモータ、ステッピングモータに対応
(イ) インタフェースなどの共通の機能の確認
⇒ ○達成※
:I/O、A/D、CAN の機能を確認。
※宣言したサイズより 2mm、重量で 1g 超過したが機能的問題はない
(ウ) これらの、小型モータコントローラドライバのプロトコルを変換・中継する、RT ミドルウェアとの共通インターフェースを持つ RTC-Lite フレームワークにあわせた、小型通信ドライバモジュールを開発する
⇒ ○達成
:小型モータ向けモータコントローラドライバに RTCc を適用した

② 小型通信ドライバモジュールのパラメータエディタ RTC の開発
各小型基盤通信モジュールへの、モータパラメータや一連の動作を登録するためのパラメータエディタRTCの開発を行う
⇒ ○達成
:各種パラメータおよび動作の設定が可能

③ RT 要素部品として小型リニアアクチュエータの開発
小型・安価・簡易でかつ安全装置が付加されたリニアアクチュエータの開発を行う
⇒ ○達成
:軸長 50mm、ストローク 30mm、軸径 16mm、瞬時最大推力 50N、
リミットセンサ具備
小型通信ドライバモジュールから簡単に制御可能

④ RT 要素部品のラインナップ
RT システムとして使用の可能性の高いモータを選び、それぞれのモータに合わせた小型通信ドライバモジュールを用意し、パラメータエディタを用いて簡単に設定できるようにする
⇒ ○達成
:THK 製アクチュエータを RT 要素部品としてラインナップ。

➢ パンフレット作成
（別紙「次世代ロボット向けエンドエフェクタ構成要素 SEED」）

⑤ 窓サッシのインテリジェント化
（ア）開発された RT 要素部品を利用し、ミサワホーム総合研究所と共に、インテリジェントウインドウシステムを設計・試作する
⇒ ○達成
：小型通信ドライバモジュールおよび小型リニアアクチュエータを適用
（イ）これを実証 RT システムとしてネットワークと結合しシステム化することで、全体システムからみた RT 要素部品の評価を行う
⇒ ○達成
：基盤通信モジュールを介してネットワーク内での動作を確認
（ウ）窓の開閉を補助するアシスト機能およびコントローラより施錠を可能とする自動施錠機能を有する RT 要素部品
⇒ ○達成
：要求機能を満たす窓開閉ユニットおよび窓施錠ユニットを試作

b-1-2 RTC-Lite フレームワークに準拠した小型通信ドライバモジュールの開発

＜目的と目標＞
THK㈱が次世代ロボット向け構成要素 SEED(Smart End Effector Devices) Solutions というコンセプトで開発している RT 要素部品群の1つである、小型・軽量なモータコントローラドライバ「SEED DRIVER」を、各種のモータに対応させる。また、SEED DRIVER を RTC-Lite フレームワークに準拠させ、RT ミドルウェアに組み込まれたモータコントローラドライバ「小型通信ドライバモジュール」として機能させる。（注：以降、SEED DRIVER を RTC-Lite フレームワークに準拠させたものを小型通信ドライバモジュールと称す）

＜成果＞
SEED DRIVER のコンセプトは、小型で汎用的なモータコントローラドライバであることとした。
開発した SEED DRIVER は、流せる電流値によって2つのサイズのものを用意した（図 b-1-1）。1A タイプの目標サイズは当初、36mm x 23mm x 7mm、10g 以内であったが、回路の安全設計の面を考慮した結果、長尺方向に基板を 2mm、高さ方向に収納ケースを 4mm 広げた。SEED DRIVER を格納すべきスペースがあらかじめ決められていなかったため、サイズ超過による問題は発生していない。モータコントローラドライバを小型化したことで、ロボット開発における重量や設置スペースの問題が大幅に改善される。モータの直近にドライバを配置することができるようになるため、モータ/エンコーダ/センサ等を接続する配線が短くて済む。これにより、配線の取り回しを考慮する手間や断線のリスクが激減される。

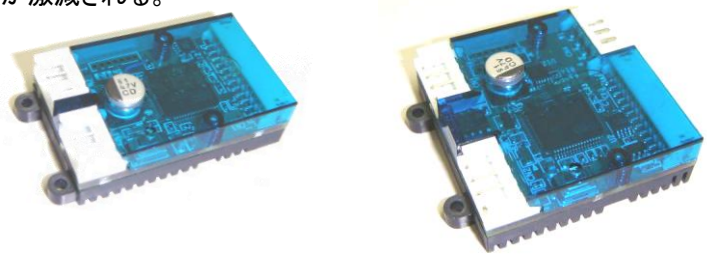


図 b-1-1 SEED DRIVER
左図：最大 1A タイプ 38mm x 23mm x 11mm、11g
右図：最大 2.8A タイプ □35mm^2、16g

SEED DRIVER で動作させることができることとなったモータは、DC モータ、DC ブラシレスモータ、2相 HB ステッピングモータ、および PM モータである。SEED DRIVER には、対応するモータによって種類がある（図 b-1-2）。制御のためのマイコンとしては、ARM7 が搭載されており、そこに開発した制御用のファームウェアを記録した。

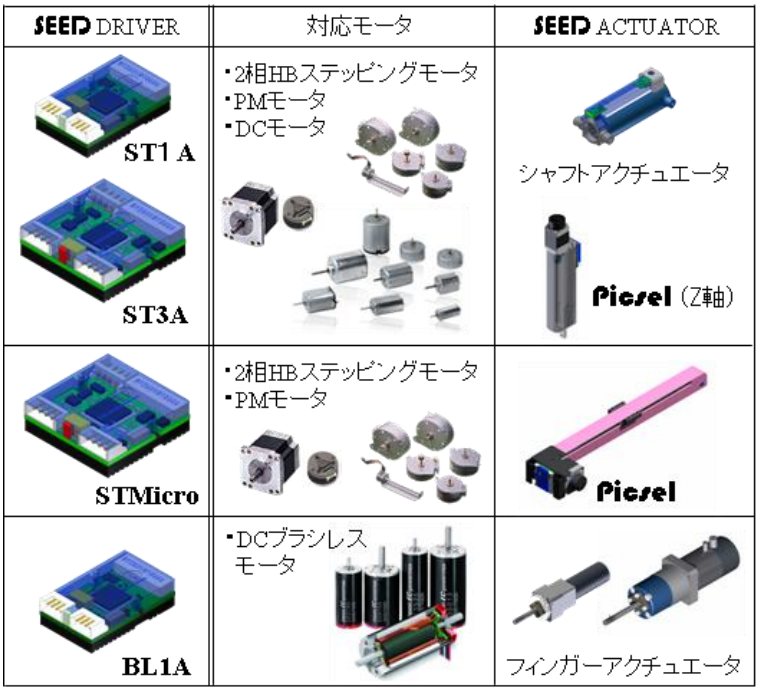


図 b-1-2　SEED DRIVER の種類と対応モータおよび THK 製アクチュエータ

通信には、主に車載ネットワークに利用されている CAN を使用しており、最高速度 1Mbps で耐ノイズ性の高いネットワークを構築できる。そして、SEED DRIVER 同士を直列に連結していく、いわゆるデイジーチェーン接続をすることができるため、省配線かつ多軸制御（図 b-1-3）をすることが可能である。その際、繋がれたモータ同士の種類が異なっていても問題なく多軸制御ができ、最大 14 軸まで接続することができる。CAN の入出力用コネクタ 2 つのほかにも、DIO:4ch、 A/D:4ch の入出力ポートがあり、各種センサを接続できる。

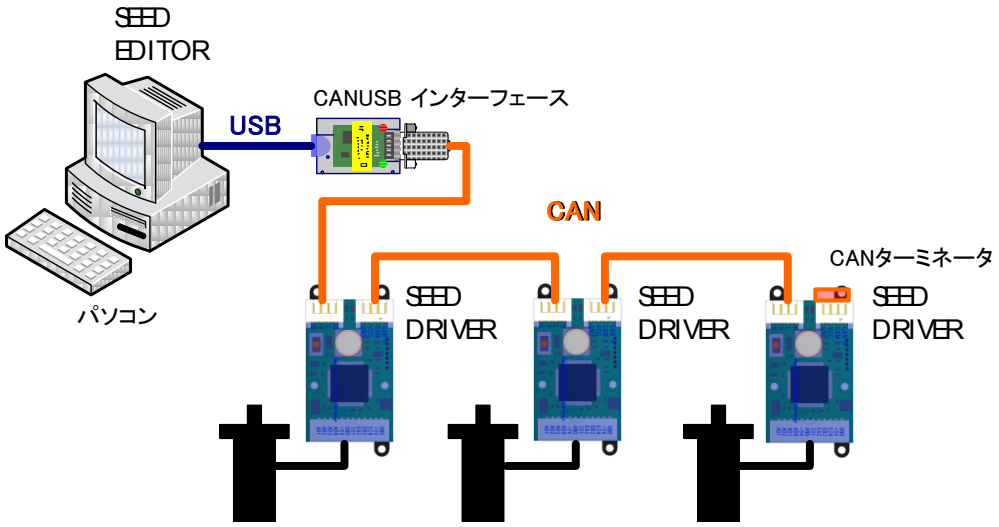


図 b-1-3　SEED DRIVER 接続例（最大１４軸接続可）

小型モータコントローラドライバ SEED DRIVER を RT ミドルウェアに組み込むためには、RTC-Lite フレームワークにもとづいて㈱セックが開発した高速制御用 RTC-Lite「miniRTCc」を適用した。

今までは、あるデバイスを RT コンポーネントとするためには、RT ミドルウェアの全ての機能を保有させなくてはならなかったため、OS として CORBA 等を実装したリッチなモジュール「基盤通信モジュール」であることが必須であった。しかし SEED DRIVER は、最小限の機能しか持っておらず、RT ミドルウェアとしてのインターフェースを持っていない。

そこで SEED DRIVER を、RTC-Lite フレームワークにもとづいて仕様が決められた miniRTCc の構成に準拠させ、RTC-Lite Manager を介することで、あたかも小型モータコントローラドライバ SEED DRIVER が１つの RT コンポーネントの中に存在するかのように仮想化することが可能となった。

miniRTCc を適用するためには、SEED DRIVER を、miniRTCc の構成でいうところのデバイスコンポーネントとして動作させる必要がある。そのために、あらかじめ miniRTCc に用意されたデバイスコンポーネント向けのファームウェアプログラムの中の、デバイス個々の動作を記述すべき部分に、SEED DRIVER のモータ制御プログラムを記述した。

以上の作業で、SEED DRIVER 単体で RT ミドルウェアに組み込まれる準備ができ、小型通信ドライバモジュールとして機能する。

しかし、RT ミドルウェアのネットワークにおいては、データポートによって通信先が固定されてしまうため、miniRTCc を適用する以前の SEED DRIVER のように、小型通信ドライバモジュール同士をデイジーチェーン接続して、互いに信号をやりとりすることができない。また、すべての小型通信ドライバモジュールをデイジーチェーン接続しようとすると、RT System Editor における結線が複雑になる。

そこで、以下のような対策を講じた。

- 小型通信ドライバモジュールは割り振られた Node ID によって親、子の役割を得る。(ファームウェアは共通)
- RT System Editor 上において、親となる小型通信ドライバモジュールを配置しその他の小型通信ドライバモジュールは親とのみ接続される
- 基盤通信モジュールからの命令は親の小型通信ドライバモジュールを介して子の小型通信ドライバモジュールへ伝達される
- 子の小型通信ドライバモジュールからのレスポンスも同様に親の小型通信ドライバモジュールを介して基盤通信モジュールへ返す
- 子の小型通信ドライバモジュール間の通信は親の小型通信ドライバモジュールを介して行う
- 各小型通信ドライバモジュールの ID 判断はデータフィールド 1byte 目で判断する
- ID15 となる設定用パソコン(開発項目②のパラメータエディタを起動)はすべての小型通信ドライバモジュールに対して CAN コマンドにて接続を行う

これらの対策を、RT コンポーネントとしての小型通信ドライバモジュールに反映させ、以下のように設定した。

- Node ID1 は基盤通信モジュールに割り振る
- 小型通信ドライバモジュールの Node ID は 2～14 までの 13 台までとする
- Node ID15 は SEED Editor(開発項目②のパラメータエディタに相当)に割り振る
- 小型通信ドライバモジュールのファーム、コンポーネントはすべて共通とし、データポート1は Out Port、2～15 は In Port として固定する
- 小型通信ドライバモジュールの Node ID はコマンドにて変更可能とする
- Node ID2 を割り当てられた小型通信ドライバモジュールは自身宛以外のコマンドを受け取った際に、同じコマンドを再送信する

また、CAN の通信プロトコルを以下のように設定した。(図 b-1-4)

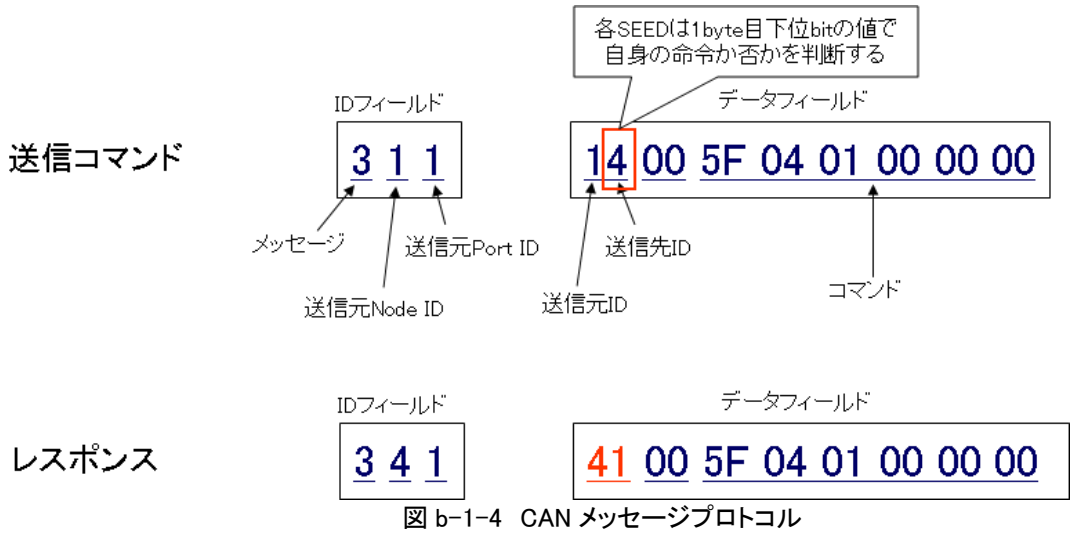


図 b-1-4 CAN メッセージプロトコル

以上を考慮して、複数の小型通信ドライバモジュールを RT ミドルウェアのネットワークで動作させるために策定した接続方法を図 b-1-5 に示す。また、例として、基盤通信モジュールから小型通信ドライバモジュールへ動作指令をした場合のフロー(図 b-1-6)と、小型通信ドライバモジュールから別の小型通信ドライバモジュールへ動作指令をした場合のフロー(図 b-1-7)を示す。

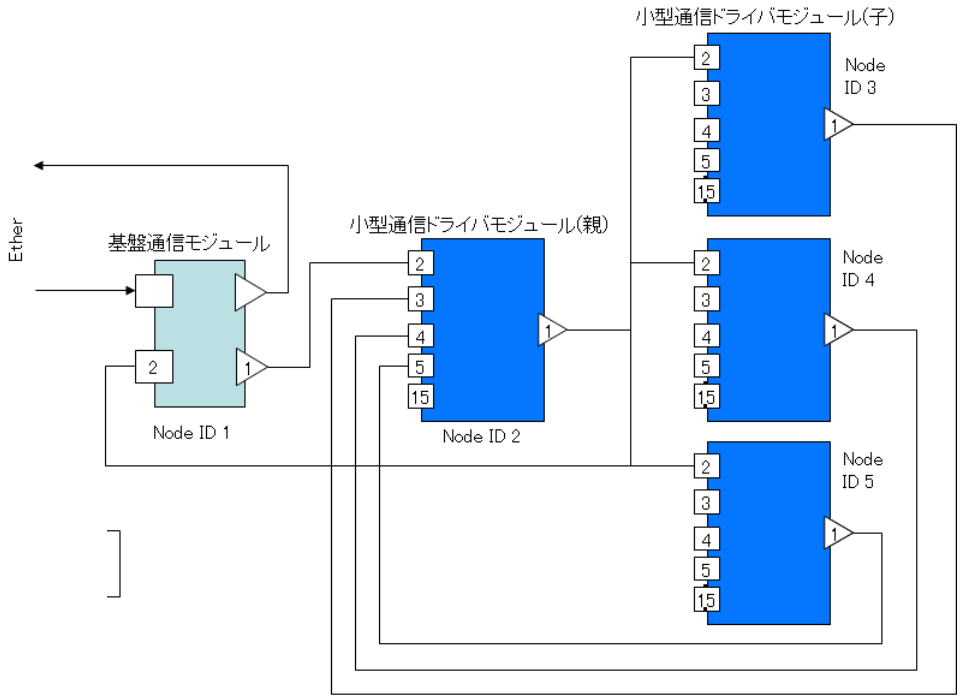


図 b-1-5 RT システムエディタにおける小型通信ドライバモジュールの結線図

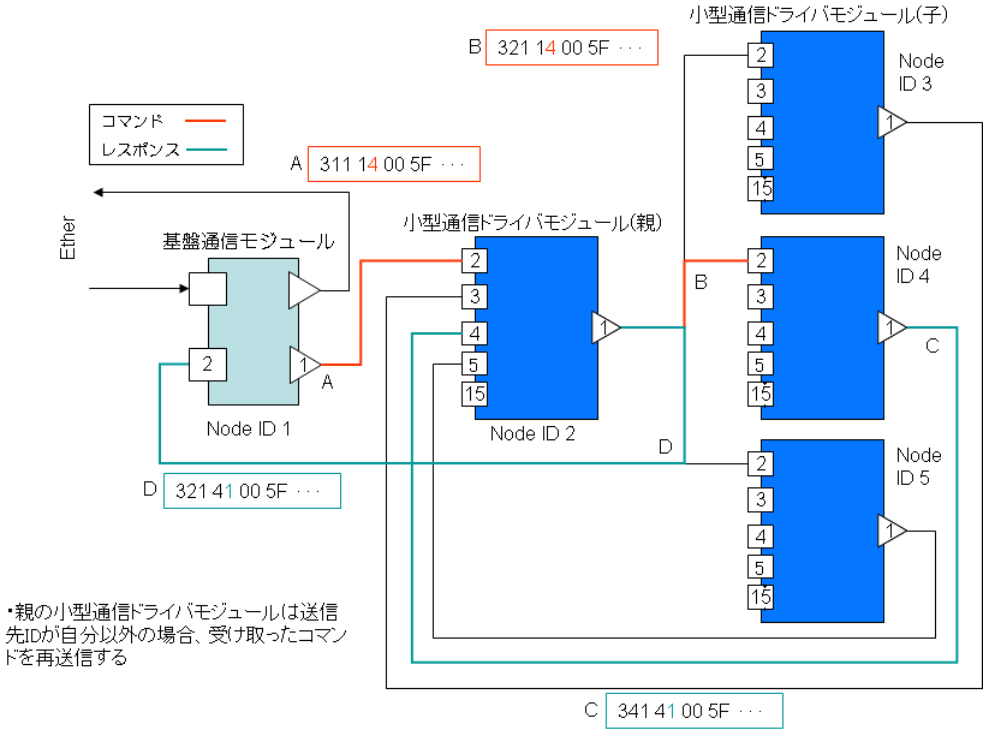


図 b-1-6 基盤通信モジュールから ID4 の小型通信ドライバモジュールへ動作指令

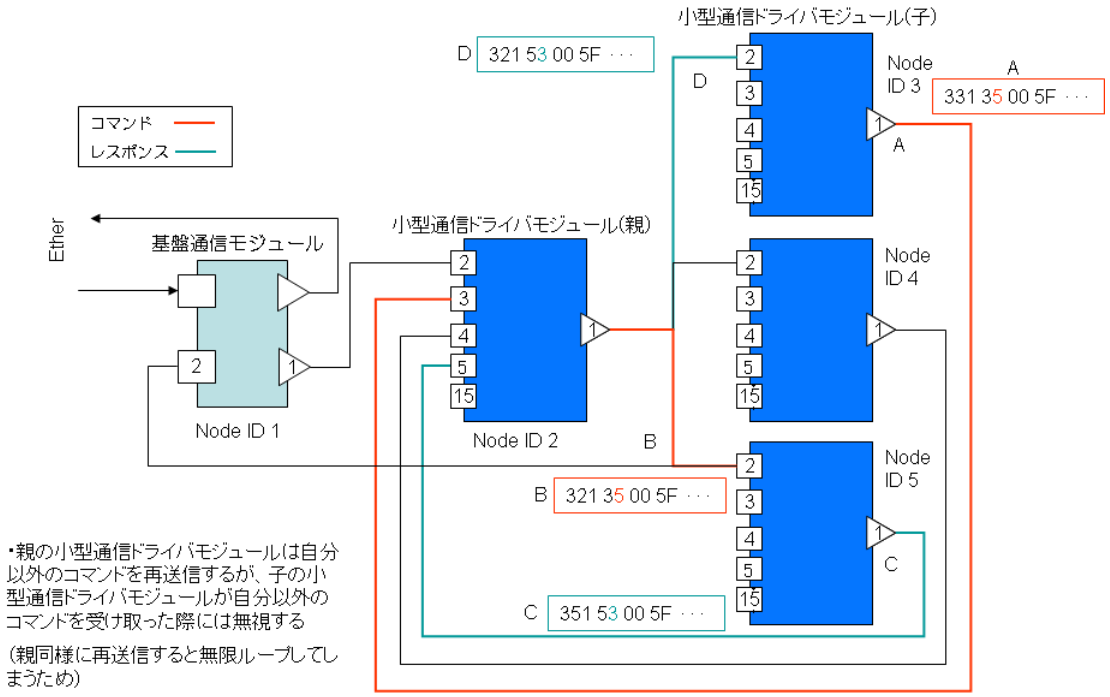


図 b-1-7　ID3 の小型通信ドライバモジュールから ID5 のへ動作指令

＜達成度＞

目標に対して実施した開発成果を以下にまとめる。小型通信ドライバモジュールの基板サイズが超過してしまった点のみ未達成であったが、機能上の問題はない。

（ア）　小型の CAN による通信機能を有した小型モータ向けモータコントローラドライバのラインナップを、各モータタイプにおいて開発した

- ステッピングモータ
- DCモータ
- DCブラシレスモータ

（イ）　インタフェースなどの共通の機能を備えていることを確認した

- 同一 CPU（ARM7）
- A/D　4ch
- I/O　4ch（エンコーダ入力対応）
- CAN 通信機能
- 基板サイズ　38mm x 23mm x 11mm

（ウ）　小型モータコントローラドライバのプロトコルを変換・中継する、RT ミドルウェアとの共通インターフェースを持つ RTC-Lite フレームワークにあわせた、小型通信ドライバモジュールを開発した

b-1-3　小型通信ドライバモジュールのパラメータエディタ RTC の開発

＜目的と目標＞
　小型通信ドライバモジュールにおいて、そのモータ動作用のパラメータや一連の動作シーケンスを設定するためのパラメータエディタ RTC を開発する。

＜成果＞
　RT ミドルウェアに組み込まれた小型通信ドライバモジュールに対して、その動作プログラムや、モータのパラメータ等の設定を行うパラメータエディタ「SEED EDITOR」を開発した（図 b-1-8）。

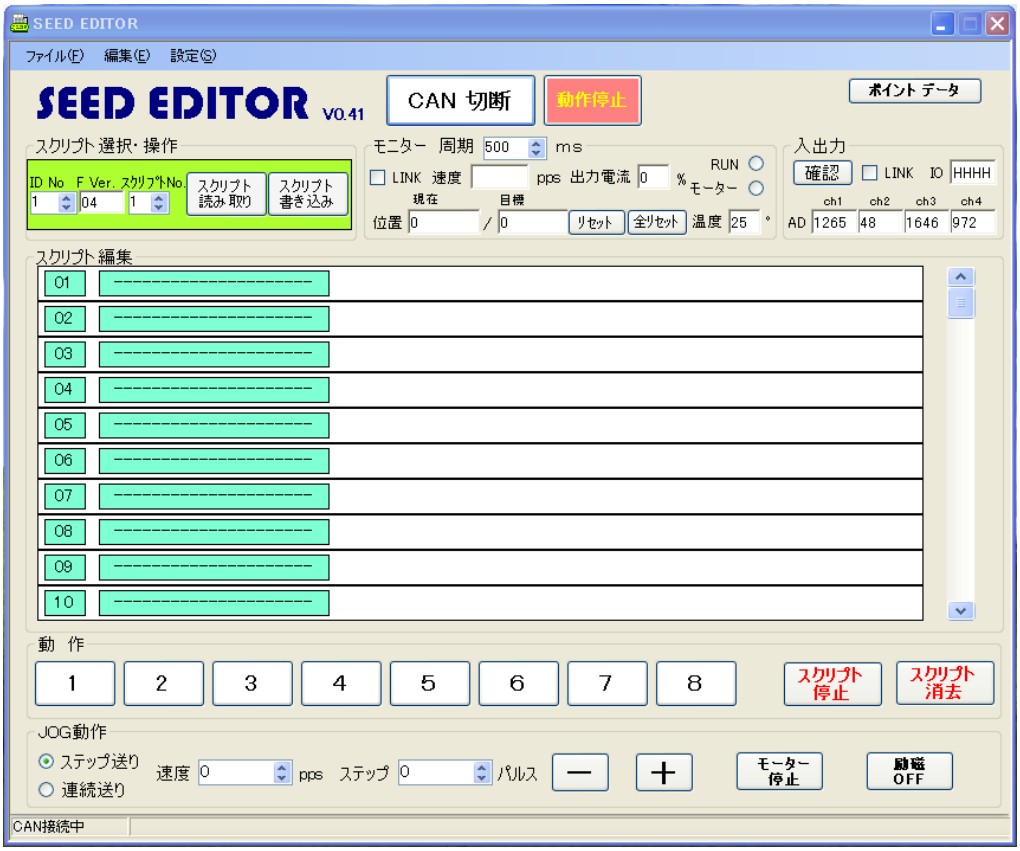


図 b-1-8　パラメータエディタ「SEED EDITOR」

　パラメータエディタは、小型通信ドライバモジュールに専用の PC 向けソフトウェアであり、Windows2000/XP パソコンで動作する。パソコンに接続されている最大 14 個までの小型通信ドライバモジュールのパラメータと、1つの小型通信ドライバモジュールに格納できる最大8つのスクリプトを編集し、ファイルへの保存をすることができる。また、オフラインでの編集を行うことも可能である。

　小型通信ドライバモジュールの動作を指示する通信方式としては、大きく分けて直接位置指令とモーション再生指令の2つがある。
　直接位置指令では、各軸に対し直接にポジションを指示し移動させる。モーション再生指令は、各小型通信ドライバモジュールに記録された一連の動作を再生させる。モーション再生指令では、あらかじめ記録されている一連の動作をいっぺんに呼び起こすことができるコマンド1つ分の通信しか必要としないため、通信負荷を低減することができる。

　小型通信ドライバモジュールの動作シーケンスを組むには、用意された様々な動作コマンド（電流制御、速度制御、位置指定、DI やアナログ値などによる条件分岐、wait、停止、など他多数）（図 b-1-9）の中から適切なコマンドを選び、順に並べる（100 行まで）だけで、1つのスクリプトとして実行することができる。スクリプトは8種類まで SEED DRIVER に記憶させることができる。
　また、目標位置「ポイントデータ」を 256 個まで記憶させることもでき、「ポイント GO」コマンドを用いることで、簡単

に所望の多点動作を実現することができる（図 b-1-10）。

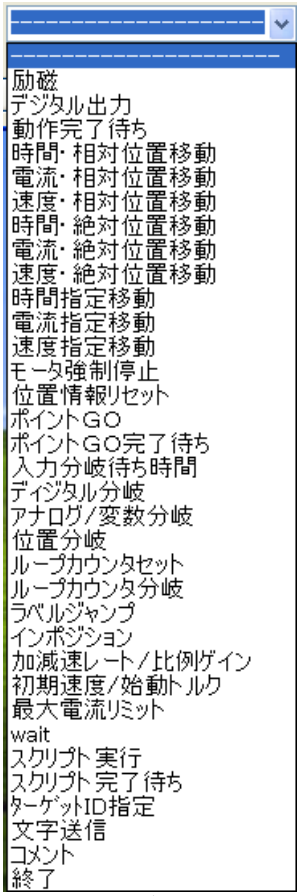


図 b-1-9　パラメータエディタ上で選択できる動作

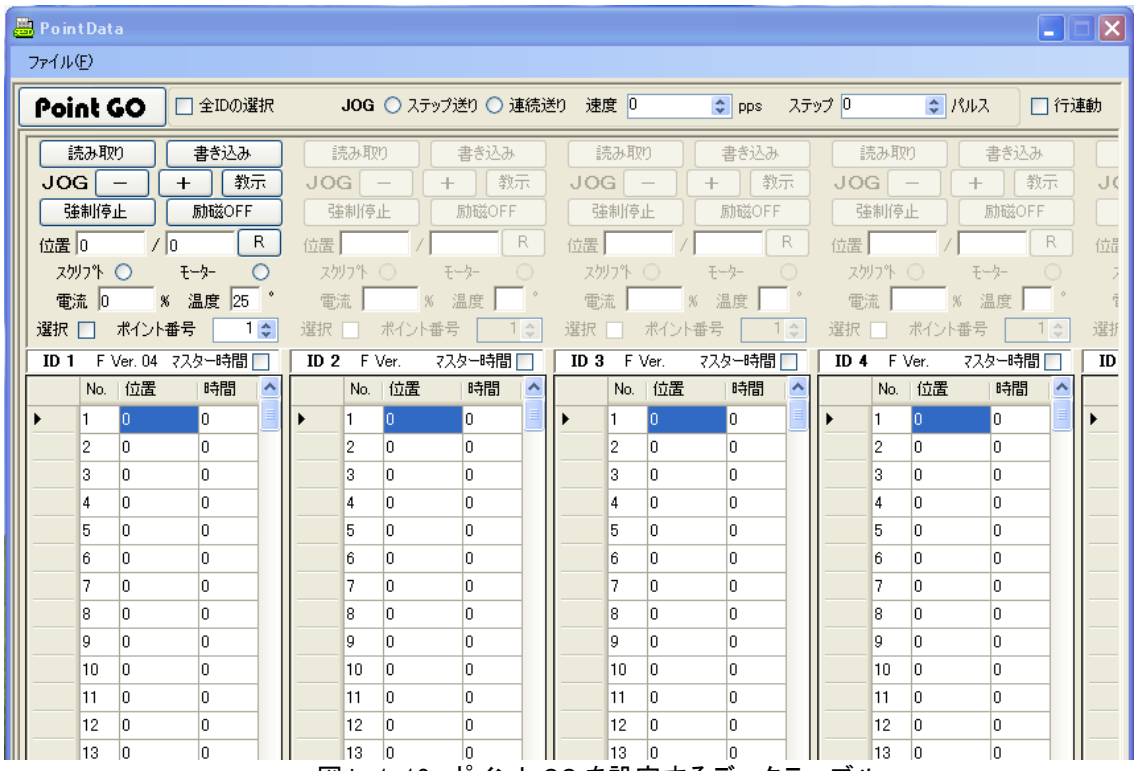


図 b-1-10　ポイント GO を設定するデータテーブル

＜達成度＞

目標に対して実施した開発成果を以下にまとめる。

（ア）　各小型基盤通信モジュールへの、モータパラメータや一連の動作を登録するためのパラメータエディタRTCの開発を行った。

- 直接位置指令
- モーション再生指令

（イ）　パラメータエディタの取扱説明書を作成した。（別紙「SEED 仕様書」）

b-1-4　RT 要素部品としての小型リニアアクチュエータの開発

＜目的と目標＞

RT 要素部品として、従来から小型の回転モータは多く用いられてきたが、使い勝手のよいリニア（直動）タイプのアクチュエータは存在しない。そこで、小型でかつ高推力を出力できる安価なリニアアクチュエータを開発する（図 b-1-11）。

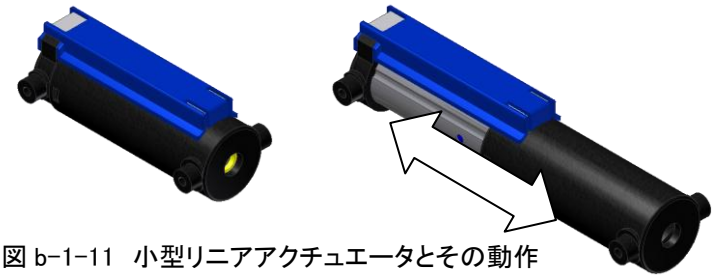


図 b-1-11　小型リニアアクチュエータとその動作

＜成果＞

開発した小型リニアアクチュエータおよび仕様を以下に示す（図 b-1-12）。

	SA-16	SA-19
長さ(縮時)[mm]	50	60
ストローク[mm]	30	40
瞬時最大推力[N]	50	150
最大速度[mm/sec]	15.6	14.6
分解能[mm]	0.042	0.042
目標価格ドライバ除	¥19,800 (Lot150)	¥19,800 (Lot150)

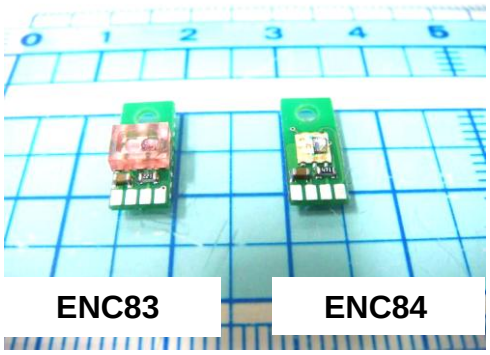
図 b-1-12　小型リニアアクチュエータの仕様

小型リニアアクチュエータの外形は円筒状とし、直径 16mm のタイプと直径 19mm の2タイプを開発した。

小型リニアアクチュエータの駆動源としては、コストを考慮して、DC モータを選定し、動力伝達部分には樹脂製めねじによるすべりねじ駆動を採用した。すべりねじ駆動はバックドライバビリティが低く、負荷を与えたまま電源をOFFしても保持された状態を維持することができる。ヒューマノイドロボットにおいては、ロボットハンドに適用すれば、対象を把持したままの状態を、住宅においては、たとえばルーバタイプのサッシの開閉機構に適用すれば、開いたままの状態を維持することができ、さらに省エネ化を図ることもできる。構想は図 b-1-13 に示すが、本プロジェクトでは実施していない。代わりに、開発項目⑤において、小型リニアアクチュエータを窓施錠ユニットの動力源として使用した。

図 b-1-13　小型リニアアクチュエータの用途例（本プロジェクトでは実施せず）

小型化の工夫として、小型リニアアクチュエータのストローク方向（図 b-1-11 に示す矢印の方向）の全長を短縮するために、DC モータ用のロータリエンコーダは使用せず、リニアエンコーダを設置した。エンコーダ受光部は新規に開発した。（図 b-1-14）また、小型リニアアクチュエータのストロークの端部には、安全のために小型リミットセンサを設置した。（図 b-1-12 左側の実写の緑色の部分）



	ENC83	ENC84
分解能	150（LPI） 0.17mm	254（LPI） 0.1mm
検出距離	2±0.5mm	0.43±0.2mm
チャンネル	A/B	
電源電圧	5V	
大きさ	5.2x12.5　mm	
重さ	2g	

図 b-1-14　開発した小型エンコーダ

小型リニアアクチュエータに組み込まれたリニアエンコーダと小型リミットセンサの信号は、小型通信ドライバモジュール（SEED　ST1A）に入力され、DC モータが制御される。DC モータの回転運動は、すべりねじ機構によって直線運動に変換される。

基礎的な評価としては、推力測定と耐久試験を行った。試験における繰り返し動作シーケンスを組むためには、小型通信ドライバモジュール SEED Driver のパラメーエディタを用いた。耐久試験では、図 b-1-15 のように、小型通信ドライバモジュールの特長であるデイジーチェーン接続ができることを利用し、複数の小型リニアアクチュエータを連結させ、全軸を同時に動作させた。

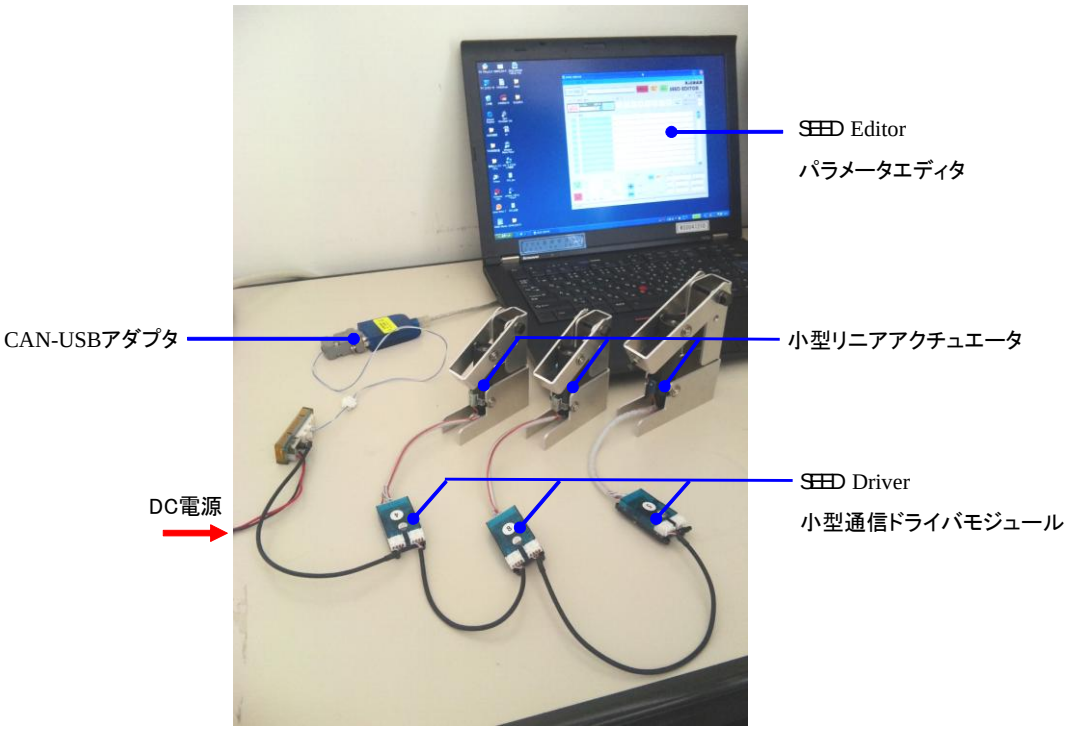


図 b-1-15　小型リニアアクチュエータの耐久試験の模様

小型リニアアクチュエータの直径 19mm タイプの推力を測定した結果を図 b-1-16 に示す。各サンプルに対して印加した直流電圧は、使用した DC モータの定格電圧値に従って、12V とした。しかし、グラフに示されるように、設計時に計算した予想値（設計値）に比べ、各サンプルの実際の瞬時最大推力の値は1／2程度しかに出力されていなかった。原因究明の結果、採用した DC モータが最大で消費するはずの電流量が流れていないことが判明した（カタログ仕様値は 1.3A であったが、実際は 0.85A）。カタログの仕様どおりの最大電流を流すために、印加する電圧値を 12V から 15V に引き上げたところ、設計値の 80%程度の瞬時最大推力が得られた。

耐久試験としては、小型リニアアクチュエータの直径 19mm タイプと直径 16mm タイプのそれぞれにおいて、ストロークの往復動作を無負荷で 50 万回行った。結果、破損等の不具合は見られなかった。

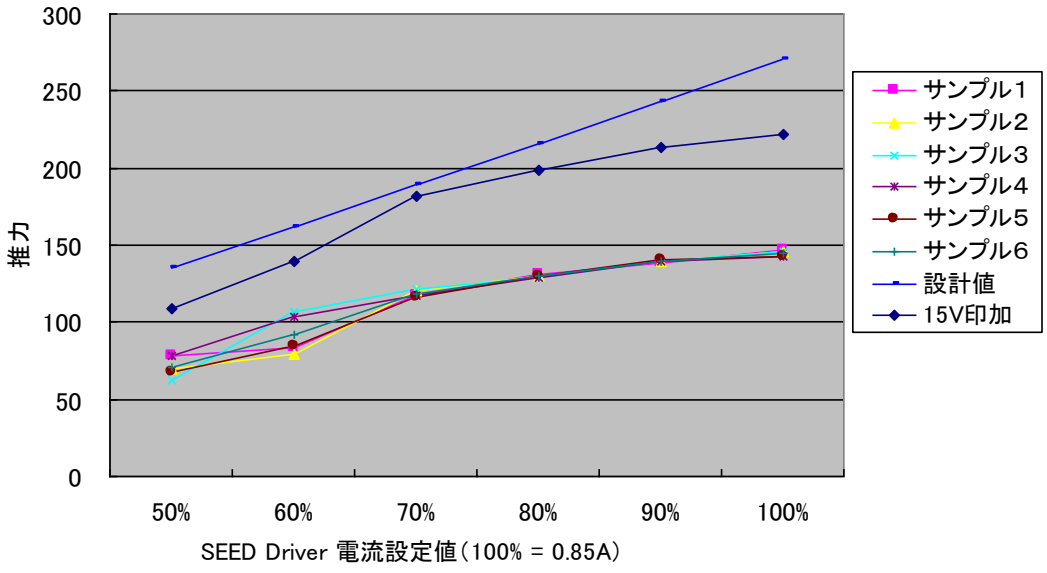


図 b-1-16　小型リニアアクチュエータの推力と印加電流

＜達成度＞

目標に対して実施した開発成果を以下にまとめる。未達事項はない。

（ウ）　小型・安価・簡易でかつ安全装置が付加されたリニアアクチュエータを開発した。

- 小型　　：軸長 50mm、ストローク 30mm、軸径 16mm
- 簡易　　：小型通信ドライバモジュールで制御可能
- 安全装置：ストロークエンドに小型リミットセンサを設置

b-1-5　RT 要素部品のラインナップ

＜目的と目標＞

RT システムとして使用する可能性が高いアクチュエータを選び、小型通信ドライバモジュールと組み合わせることで、予めパラメータや動作が設定されていて扱いやすい RT 要素部品として、販売・展開していくための体制を整える。

＜成果＞

小型通信ドライバモジュールを適用したアクチュエータとしては、開発項目③で開発した小型リニアアクチュエータと、THK で既に開発が完了している 3 つのアクチュエータ（下記 ii－iv、概観：図 b-1-17、対応表：図 b-1-2）となる。また、小型通信ドライバモジュールが流せる最大電流より容量の大きいモータや交流 100V を電源とする機器の ON／OFF または正転／逆転等をするために、メカニカルリレーユニットを新規に開発した（図 b-1-18）。住宅向けの電動機器に対して小型通信ドライバモジュールを適用したものについては、開発項目⑤に記載する。

それぞれ異なるモータを内蔵したアクチュエータであるが、いずれも小型通信ドライバモジュールに接続すれば、パラメータエディタを用いて簡単に動作やパラメータを設定することができる。また、小型通信ドライバモジュールは RTC-Lite フレームワークに準拠した miniRTCc が適用されているので、RT ミドルウェアのネットワークに組み込まれたこととなる。

（ｉ）	小型リニアアクチュエータ	：DC モータ内蔵	、小型・高推力
（ii）	フィンガーアクチュエータ	：ブラシレス DC モータ内蔵	、小型・超高推力
（iii）	Picsel（ピクセル）	：ステッピングモータ内蔵	、セル生産向け
（iv）	ハンド	：PM モータ内蔵	、セル生産向け
（ｖ）	メカニカルリレーユニット	：IO 信号によりスイッチング	、ON/OFF 動作

図 b-1-17　小型通信ドライバモジュールが適用可能な RT 要素部品
（左より、フィンガーアクチュエータ、Picsel、ハンド）

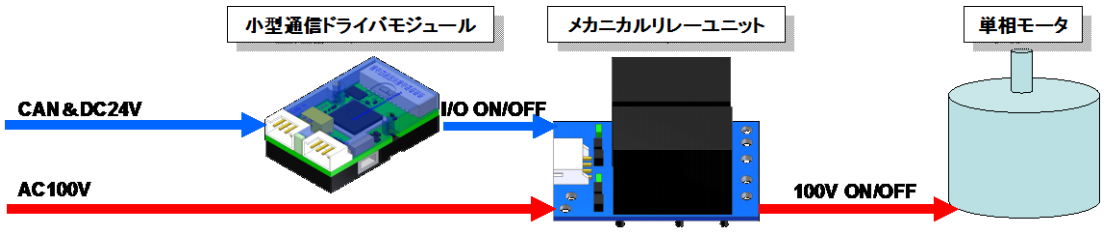


図 b-1-18　メカニカルリレーユニット

選んだアクチュエータと小型通信ドライバモジュールを RT 要素部品として販売していくための事前 PR として、2009 年度にビックサイトにて開催された国際ロボット展に出展した。展示ブースには専用の展示台（図 b-1-19）を設け、小型通信ドライバモジュールやアクチュエータの有用性を訴えた。また、RT 要素部品を網羅したパンフレットを用意した。パンフレットについては別紙参照のこと。ただし当時は、miniRTCc の実装が済んでいなかったため、RT ミドルウェアについては触れず、あくまでの RT 要素部品としての PR がなされた。

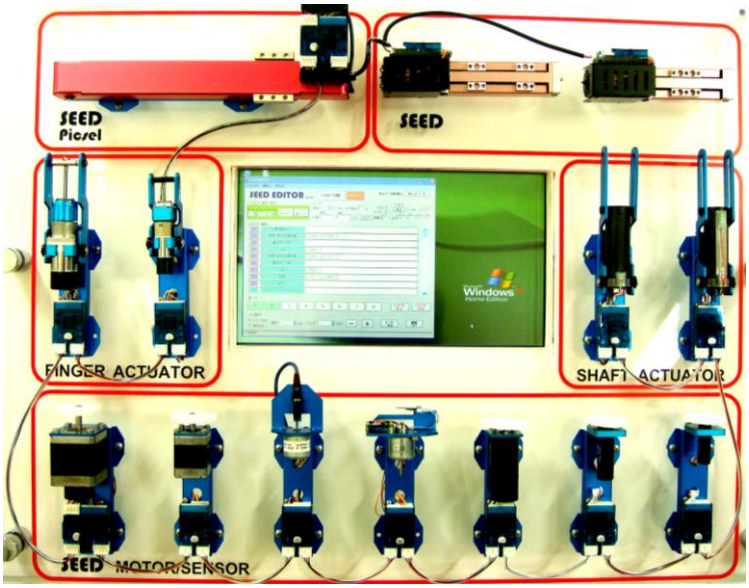


図 b-1-19 RT 要素部品の展示台

＜達成度＞

目標に対して実施した開発成果を以下にまとめる。

（ア） RT システムとして使用の可能性の高いアクチュエータを選び、対応した小型通信ドライバモジュールと組み合わせて、パラメータエディタを用いて簡単に設定できるようにし、ラインナップした

（イ） 用意したラインナップを網羅したパンフレットを作成した
（別紙「次世代ロボット向けエンドエフェクタ構成要素 SEED」）

b-1-6　窓サッシのインテリジェント化

＜目的と目標＞

開発した RT 要素部品を利用し、インテリジェントウインドウシステムを設計・試作する。また基盤通信モジュールを介することで、RT ミドルウェアで構成された RT 住宅のネットワークにインテリジェントウインドウシステムを繋ぎ、RT 住宅全体のシステムからみた RT 要素部品の評価を行う。また、RT 住宅向けに、窓の開閉を補助するアシスト機能およびコントローラより施錠を可能とする自動施錠機能を有する RT 要素部品を開発する。

＜成果＞

試作したインテリジェントウインドウシステムの概要図を図 b-1-20 に示す。インテリジェントウインドウを構成しうるすべての電動機器は、それぞれのモータドライバ・コントローラとして開発した小型通信ドライバモジュールが接続されている。図 b-1-20 のように、小型通信ドライバモジュールおよび基盤通信モジュールを、CAN 通信用の信号線（High/Low の 2 線）でデイジーチェーン接続するのみで、インテリジェントウインドウのネットワークが構築されている。また、引き違いの大型の引き窓の開閉を補助するパワーアシスト機能におけるスイッチからの入力信号は、無線＝Zigbee 通信により、ネットワークに接続されている。

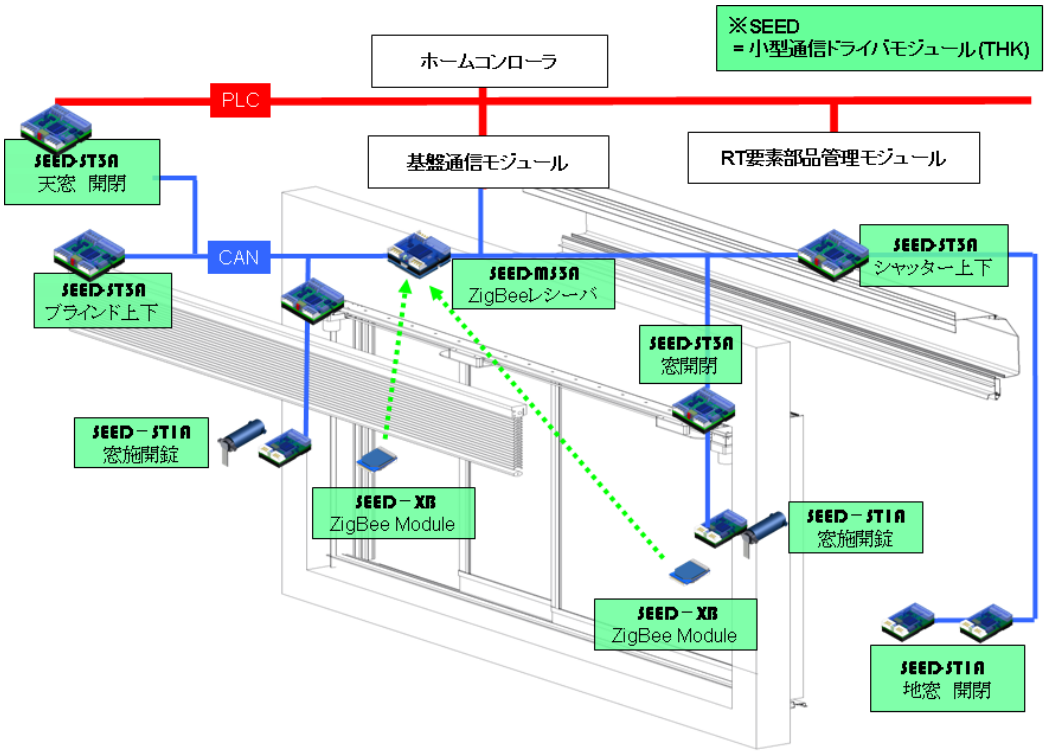


図 b-1-20　インテリジェントウインドウシステム

図 b-1-21　インテリジェントウインドウ概観

シャッターやブラインドのように、すでに製品として市場に出回っている住宅向け電動機器については、株式会社ミサワホーム総合研究所にて選定された機器を使用した。そして、電動機器に付属の純正のドライバ・コントローラを、小型通信ドライバモジュールに置き換え、RT ミドルウェアのシステムに組み込んだ。結果、純正のドライバ・コントローラユニットを小型化することができ、日本住宅のような小さな住居のための省スペース化に寄与できる可能性を示した。

表図 b-1-1　住宅向け電動機器とそのモータおよび対応する小型通信ドライバモジュール

住宅向け機器	メーカー	駆動源	小型通信ドライバモジュール
シャッター	三和シャッター	ステッピングモータ	ST3A
ブラインド	ニチベイ	DCモータ	ST3A
天窓	トステム	DCモータ	ST3A
地窓	YKK ap	DCモータ	ST1A
窓開閉ユニット	THK	DCモータ	ST3A
窓施錠ユニット	THK	小型リニアアクチュエータ	ST1A

小型通信ドライバモジュールが接続された電動機器の外観を図 b-1-22 に示す。

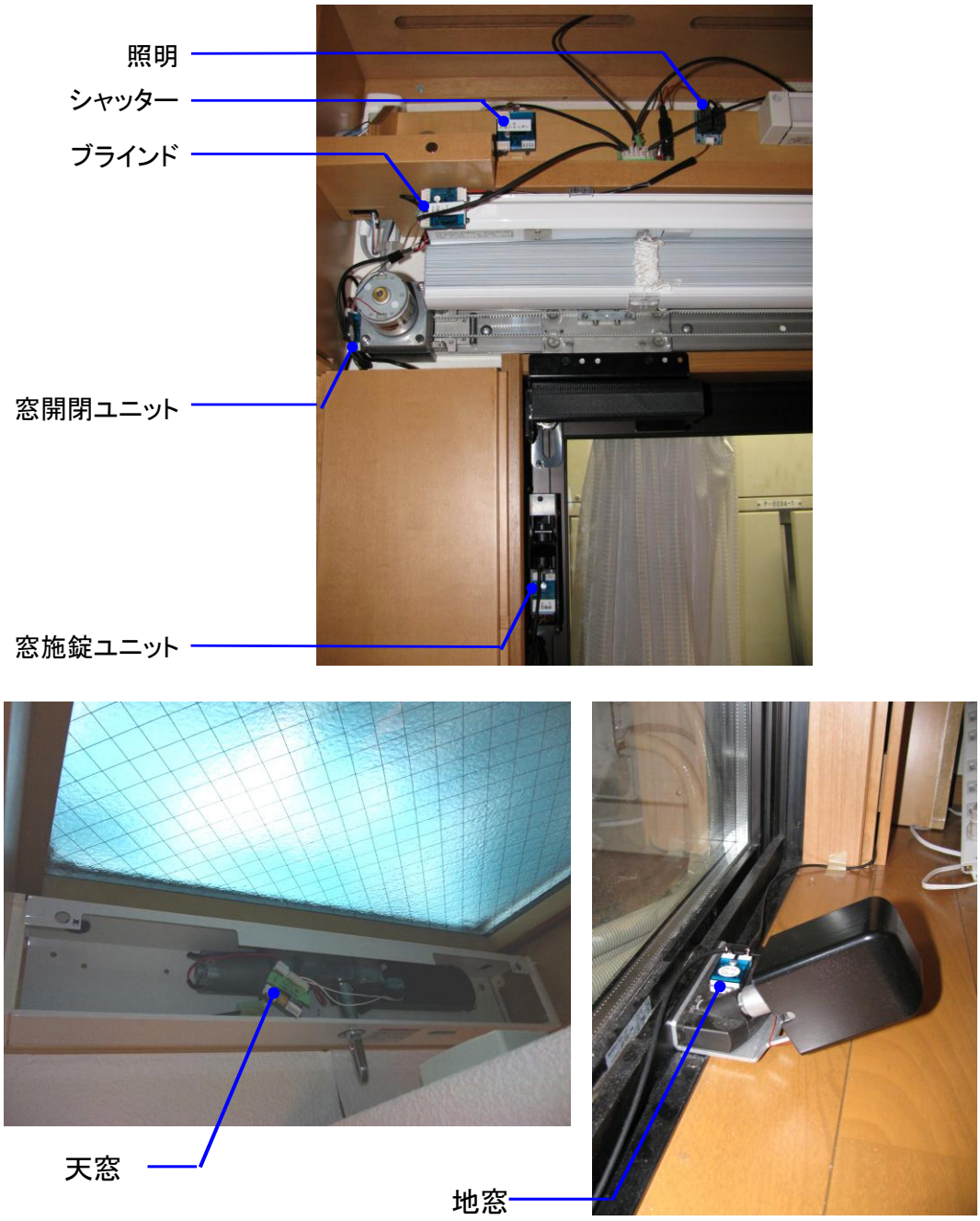


図 b-1-22　インテリジェントウインドウシステムにおける
小型通信ドライバモジュールの配置

小型通信ドライバモジュールでは、既存のドライバ・コンとローラと同様に、障害物にぶつかったり手を挟んだりしたときのための安全機構として、動作中に一定の電流値以上が流れようとすると停止するように設定した。

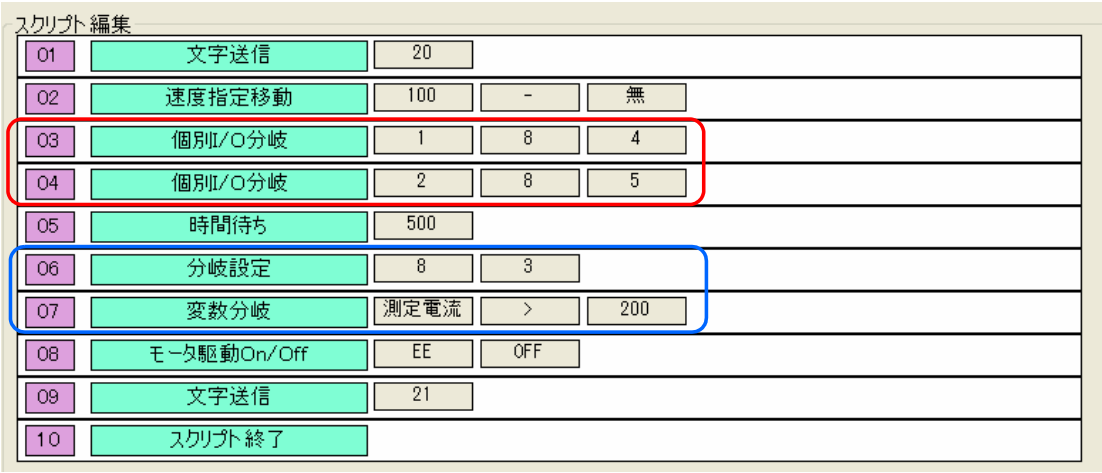


図 b-1-23　ブラインドのスクリプト

開発したパラメータエディタを用いて設定したブラインドのスクリプト（動作シーケンス）を図 b-1-23 に示す。各行（コマンド）の説明は以下のとおり。

01 行目：　開き動作を開始することを基盤通信モジュールへ伝えるために、
　　文字列「20」を CAN 送信
02 行目：　モータの速度と回転方向を宣言
03 行目：　I/O ポート1からの入力（リミットセンサ）が Hi の場合は、
　　08 行目へジャンプ、Low の場合は 04 行目へジャンプ
04 行目：　I/O ポート 2 からの入力（リミットセンサ）が Hi の場合は、
　　08 行目へジャンプ、Low の場合は 04 行目へジャンプ
05 行目：　500[ms]の wait
06 行目：　次行の条件式が真なら 08 行目へジャンプ、偽なら 03 行目へジャンプ
07 行目：　条件式「小型通信ドライバモジュールの消費電流値が 200 以上の場合」
08 行目：　モータへ電流を流すのをやめる
09 行目：　開き動作が完了したことを基盤通信モジュールへ伝えるため、
　　文字列「21」を CAN 送信

図 b-1-23 において赤枠で囲った部分には、動作開始時にリミットセンサを踏んでいた場合は動作を即座に終了するためのコマンドが、また、青枠の部分には、消費電流値が既定値以上になった場合に動作を終了するためのコマンドが記述されている。

RT ミドルウェアのネットワークが構築された RT 住宅システムにおいて、インテリジェントウインドウ内の電動機器（RT 要素部品）が、他の RT コンポーネントからの動作指令を受けて適切に動作するか検証した。検証にあたっては、セック㈱が用意した操作用タブレットを使用し、その画面に記載された各種のボタンに対応して、電動機器が単体で、または複数の電動機器が連係して動作することを確認した。また、その動作状態を示すステータス値が、ネットワークを経由して操作用タブレットへ返され、表示されることも確認した。

ユーザーが選択できるモードと、それに応じて連係動作する電動機器を以下にまとめる。

・　おはようモード　：ブラインドとシャッターが同時に開く（図5．5）
・　おやすみモード：ブラインドとシャッターが同時に閉じる
・　自動空調モード：温度を考慮して天窓と地窓が開閉して空調管理（図5．6）
・　外出モード　　：すべての窓およびブラインドが閉じる

2010 年 9 月と 2011 年 1 月に、本プロジェクト関係者に対して実証デモンストレーションが行われたが、その際も問題なく動作した。メーカーの異なる住宅向け電動機器であっても、1つの RT ミドルウェアのネットワークに簡単に組み込まれて、連係して動作することが可能となった。

図 b-1-24　おはようモード

図 b-1-25　自動空調モード

引き違いの大型の引き窓の自動開閉機能、および弱い力でも開閉動作を容易に行えるようにするためのパワーアシスト機能を実現するにあたっては、既製品がないため、窓開閉ユニットの試作を行った。(図 b-1-26)

窓開閉ユニットの仕様としては、以下の点を考慮した。

- 人間共存ロボットの安全基準値「速度:250mm/s、動力 150N」以上のパワーは仕様として出せないような低容量のモータを選定
- 万が一、窓開閉ユニットが膠着した場合の、窓サッシとの着脱機能(図 b-1-28)
- 過って人間や障害物が挟まってしまったときに停止させるために、挟み込み検知機能として、サッシ端部にタッチセンサを設置(図 b-1-27)。消費電流値も監視
- 押した分だけ高速度で窓が追従するパワーアシストスイッチを設置(図 b-1-29)

図 b-1-26(左図)　窓開閉ユニット
図 b-1-27(右図)　パワーアシストスイッチおよびタッチセンサ(グレー部分)

図 b-1-28　窓開閉ユニットと窓サッシとの着脱機構(左図:連結　右図:解放)

図 b-1-29　パワーアシスト動作の模様

パワーアシストスイッチのレバーを指で押すと、パワーアシストスイッチから無線信号（Zigbee）が発せられ、カーテンボックス内に配置されたレシーバユニットがその無線信号を受信し、窓開閉ユニットの小型通信ドライバモジュールに対して、動作速度を指示する。指の押し量に従い窓を追従させるためには、パワーアシストスイッチのレバーの傾斜角の閾値を3段階に分けて、それに応じた窓開閉の速度を指示している。

開口部としての窓の防犯性を確保するために、窓開閉ユニットにあわせて施錠ユニットの試作を行った（図 b-1-30）。施錠ユニットの仕様としては、以下の点を考慮した。

- 窓が閉じると自動的（電動ではなくばねを用いた機械式）にロックがかかり、施錠／開錠状態が確認できる
- 開錠するためには、ホームコントローラからの信号で自動的に開けることも、災害時などのために屋内から手動で開けることもできる。
- 窓を開ける際には、窓開閉ユニットと連係して、開く前に自動で開錠する
- 防犯性を考慮し、バックドライバビリティの小さいアクチュエータとして、駆動源に、開発した小型リニアアクチュエータを適用する

図 b-1-30　窓施錠ユニット
（左図：構造→小型リニアアクチュエータ内蔵
中央：設置概観　右図：手動開錠）

窓施錠ユニットは、内蔵された小型リニアアクチュエータの直動の動作によりフックを開閉して、窓サッシに設置した突起を引っ掛ける、または解放することで、窓の施錠・開錠を行う。開錠状態のときと施錠状態のときではフックの位置が異なり、窓施錠ユニットに具備されたリミットセンサによってその状態を判別することが可能である。屋内側に飛び出したレバーを手で引くことで、開錠することも可能である。これは、万が一の動作不良や災害が生じた場合に備えた、緊急脱出用の機能である。小型リニアアクチュエータを用いているので、空き巣などが窓を外部から無理にこじ開けようとしても、アクチュエータがバックドライブして窓が開いてしまうといったことは生じ得ない。

＜達成度＞

目標に対して実施した開発成果を以下にまとめる。

（ア） 開発されたRT要素部品を利用し、ミサワホーム総合研究所と共に、インテリジェントウインドウシステムを設計・試作した

- 市販の住宅向け電動機器に小型通信ドライバモジュールを適用してRTC化
  - 天窓　（TOSTEM）
  - 地窓　（YKKap）
  - シャッター　（三和）
  - ブラインド　（ニチベイ）
  - 照明器具
- 基盤通信モジュールと小型通信ドライバモジュールを接続してネットワーク化
  - RT System Editor上での接続方法については開発項目①に記載（図1．5）

（イ） インテリジェントウインドウシステムを実証 RT システムとしてネットワークと結合しシステム化することで、全体システムからみたRT要素部品の評価を行った

- ホームコントローラからの動作指令を受けて適切に動作
  - 操作用タブレットより動作指令・ステータス取得
- メーカーの異なる住宅向け電動機器の連係動作が実現
  - おはようモード
  - おやすみモード
  - 自動空調モード
  - 外出モード

（2010年9月、2011年1月のデモンストレーションにおいて実証）

（ウ） 窓の開閉を補助するアシスト機能およびコントローラより施錠を可能とする自動施錠機能を有する RT 要素部品を開発した

- 窓開閉ユニット
  - 安全機能　：低容量モータ、緊急時の着脱
  - 開閉アシスト機能　：押し量に合わせて速度を変化させて追従
- 窓の自動施錠ユニット
  - 防犯機能　：小型リニアアクチュエータを利用
  - 安全機能　：緊急時の手動による開錠
  - 窓開閉ユニットとの連係動作

(b-2) 環境情報計測用センサ要素部品の開発
（委託先:株式会社アルゴシステム）

b-2-1 概要

基盤通信モジュールを有したRT要素部品の1つとして住宅内や住宅外に設置する環境情報計測センサ要素部品を開発した。既存のセンサに基盤通信モジュールを付加することで RT ミドルウエア上に参加可能な要素部品のハードウエアを開発、実証 RT システムに導入して評価した。当初、センサ要素部品として Zigbee を組み込んだ温度、湿度センサモジュール、人感センサモジュールや照度センサモジュールの開発を計画し、実施したがコスト、サイズや電池駆動に関する課題を解決するため、一般的なセンサ共通に接続可能な Zigbeee エンドポイントを開発した。本件での目標、並びに達成度は表 b-2-1 の通りである。

表 b-2-1　研究項目に対する目標並びに達成度

研究開発項目	目標	達成度
環境情報計測用センサ要素部品の研究開発	既存のセンサに基盤通信モジュールを付加することで RT ミドルウエア上に参加可能な温湿度、人感、照度などの各センサの RT 要素部品化したハードウエアの研究開発。 安価なRT要素部品化に使用するZigbeeエンドポイントの原価を 2,000 円以下にする。	温湿度センサ、人感センサ、照度センサなどの RT 要素部品化を図り、RT ミドルウエア上に参加可能にするための Zigbee エンドポイントのモジュール開発を行い、関係機関に配布した。 ・センサーなど安価な RT 要素部品化に使用の Zigbee エンドポイントの原価は 2,000 円以下を実現した。 ・Zigbee エンドポイントは名刺サイズの 1/8 を実現した。 ・低消費電力化についは消費電流を 980μ A を実現して電池駆動を可能にした。

b-2-2 Zigbee エンドポイントの開発

1)概要と特徴

・本製品はTI製 CC2530を搭載。

・センサ用コネクタを搭載。(A/D 4ch、DIO 4 点、センサ用電源)

・Zigbee 通信により、Zigbee ステーションと通信可能。

本製品の特長を以下に示す。

① PUはTI製ZigbeeであるCC2530(16MHz)を搭載
② CPU内蔵メモリ 8KByte搭載
③ CPU内蔵FLASH 256KByte搭載

図 b-2-1 Zigbee エンドポイント

2)仕様

・ 電気仕様

表 b-2-2 電気仕様

項 目		仕 様
電 源	定格電圧	DC3V
	電圧許容範囲	DC2.4V～3.3V
	許容瞬時停電時間	1ms 以下
	内部消費電力	1W 以下

・ 環境仕様及び質量

表 b-2-3 環境仕様及び質量

項 目		仕 様
物理的環境	使用周囲温度	0～50℃(取付け角度による制限有り)
	保存周囲温度	-25～70℃
	使用周囲湿度	30～90%RH(結露無きこと)
	保存周囲湿度	30～90%RH(結露無きこと)
	使用雰囲気	腐食性ｶﾞｽ無きこと

・ 機能仕様

表 b-2-4　機能仕様

項　目	仕　様
CPU	Texas Instruments 社製 CPU CC2530 16MHz
RAM	8Kbyte(CPU 内蔵)
ROM	256Kbyte(CPU 内蔵)
RS-232C	1ch（Max38400bps）
A/D	4ch
DIO	4 点
センサ用電源	3.3V　1点

3)各部の名称

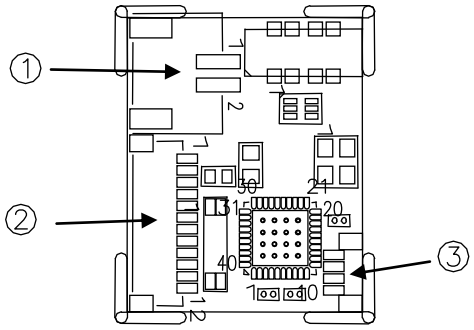


図 b-2-2　各部の名称

表 b-2-5　名称の説明

No.	名　称	内　容
①	電源ｺﾈｸﾀ	1　3.3V 2　GND 適合ｺﾈｸﾀ　:PHR-2 (JST 製) 適合電線ｻｲｽﾞ　:AWG#30～AWG#24

<table>
<tr><td>②</td><td>ｾﾝｻｺﾈｸﾀ</td><td>
<table>
<tr><td>1</td><td>3.3V</td></tr>
<tr><td>2</td><td>DIO 1</td></tr>
<tr><td>3</td><td>DIO 2</td></tr>
<tr><td>4</td><td>DIO 3</td></tr>
<tr><td>5</td><td>DIO 4</td></tr>
<tr><td>6</td><td>AD 1</td></tr>
<tr><td>7</td><td>AD 2</td></tr>
<tr><td>8</td><td>AD 3</td></tr>
<tr><td>9</td><td>AD 4</td></tr>
<tr><td>10</td><td>TX</td></tr>
<tr><td>11</td><td>RX</td></tr>
<tr><td>12</td><td>GND</td></tr>
</table>
適合ｺﾈｸﾀ　　：
SM12B-SRSS-TB(JST 製)
適合電線ｻｲｽﾞ　：AWG#30～AWG#28
</td></tr>
<tr><td>③</td><td>Zigbee 用 ICE ｺﾈｸﾀ</td><td>Zigbee 用 ICE ｺﾈｸﾀです
<table>
<tr><td>1</td><td>TCK</td></tr>
<tr><td>2</td><td>TDA</td></tr>
<tr><td>3</td><td>RST</td></tr>
<tr><td>4</td><td>GND</td></tr>
</table>
適合ｺﾈｸﾀ　　：SHR-04V-S-B (JST 製)
適合電線ｻｲｽﾞ　：AWG#30～AWG#28
</td></tr>
</table>

4)ブロック図

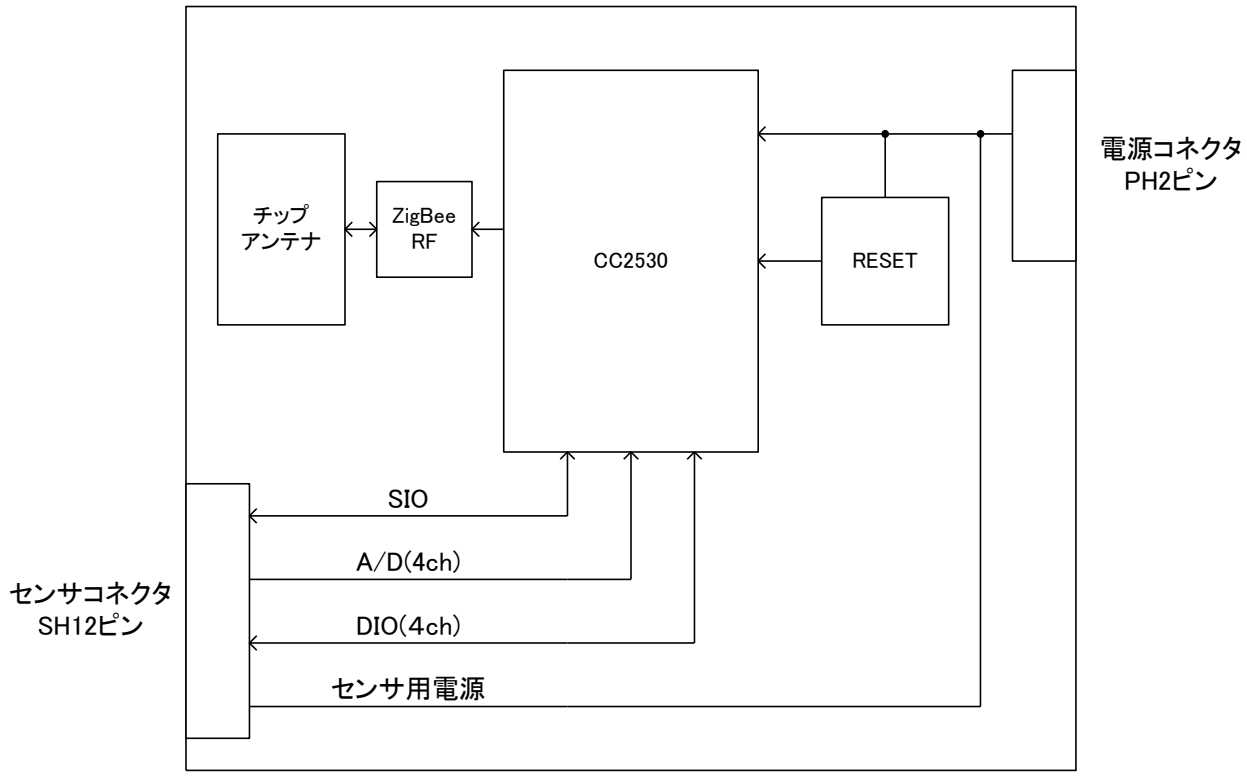


図 b-2-3　ブロック図

5)外形寸法図

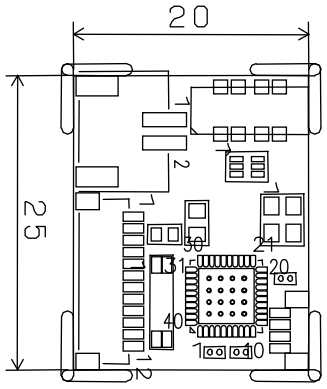


図 b-2-4　外形寸法図

(b-3) 分散する RT 要素部品の RT コンポーネント開発とコンポーネント作成ツールの評価
（委託先：国立大学法人大阪大学）

b-3-1 概要

本研究開発課題では、RT ミドルウェアに精通していないデバイスベンダと RT Middleware、RTC-Lite およびその利用のためのツール群といった RT コンポーネントを作成するための基盤フレームワークを提供するメーカとの間を取り持ち，その間での検証を行うことを目的として、以下の課題を設定し，開発を進めた。

①デバイスベンダより提供されるデバイスに応じた RT コンポーネント開発
②RTC-Lite および開発ツールの検証およびその評価

上記は本研究開発チーム内での研究成果内の成果物への RTC-Lite の適用による RT ミドルウェアネットワークへ参加するための研究開発項目について述べたが，汎用的に利用されている CAN 通信を利用する製品を同様のネットワークに参加させるための仕組みも利用することでシステムの汎用性が増す。そこで，RTC-Lite が適用されていないデバイスを部品管理モジュールにおいて認識できる仕組みとして，上記研究開発案件に加え以下の項目についても検討を行った。

③RTC-Lite を利用しない直接的な RT ミドルウェアネットワーク参加を実現するブリッジの開発

それぞれの目標、成果ならびに達成度について表 b-3-1 にまとめる。

表 b-3-1 課題の目標、成果および達成度

目　標	研究開発成果	達成度
① デバイスベンダより提供されるデバイスに応じた RT コンポーネント開発		
デバイスの推奨する制御方法と RT ミドルウェアフレームワークとの間での整合性を取りながら，安定して動作するためのコンポーネント設計，インタフェース設計，各デバイスベンダへのコンポーネント供給を行う.	オカテック社のモータドライバを対象とし，モータドライバのコントローラへの組み込み RTM の実装，および実装を通じたフィードバックを行うとともに，組み込み RTM 対応製品開発に関わるコスト試算を行った.	企業との協力の下での既存製品への組み込み RTM の導入，および開発に関わるコスト試算ができており，普及に向けた十分な成果が創出できた.
② RTC-Lite および開発ツールの検証およびその評価		
RT 要素部品の実装を通じて，RTC-Lite の評価を行い，問題点を開発機関にフィードバックする. および RT 要素部品の実装時にテクノロジックアートより提供される開発支援ツールを用い，その評価および問題点のフィードバックを行う.	ミサワホーム住宅展示場システム，および産総研実証スペースにおけるシステムの実装を通じて，逐次問題点をフィードバックし，組み込み RT ミドルウェア開発の開発者視点からの利用事例の提供を行った.	組み込み RT ミドルウェアを用いた RT 要素部品による制御系の構築，及びシステム化を通じて，多くの利用事例の創出ができており，十分な成果を上げた.
③ RTC-Lite を利用しない直接的な RT ミドルウェアネットワークの開発		
CAN ネットワークも RT ミドルウェアに参加できるようなブリッジ機能を RT ミドルウェアに追加する.	旭光電機社製エリア型人感センサにおいて，ブリッジユニットによる RT ミドルウェアへの対応を行った.	ブリッジユニットにより，既存製品に手を加えることなく，RT ミドルウェアに対応できることが確認され，実用化に向けたコスト試算も行えたため，十分な成果が上がったと言える.

b-3-2　デバイスベンダより提供されるデバイスに応じた RT コンポーネント開発

ホームネットワークシステムの開発において，株式会社セックが開発する RT ミドルウェア miniRTCs,microRTCs（以降，組み込み RTM）を用いたシステムを普及させていくためには，利用事例を創出していくことが重要と言える．特に，実際に市場に流通している製品に対して組み込み RTM を導入した事例を見せていくことは，開発コストの試算をする上でも重要な要素と言える．そこで，RT 要素部品を開発する企業の製品を対象として，組み込み RTM の導入を行った．

モータドライバの販売を行っているオカテック社と協力し，オカテック社製品への組み込み RTM の実装を行った．組み込み RTM は元々THK 株式会社で開発される SEED 上に実装されている MPU である Cortex-M3 に合うように実装されているため，ここでの研究課題は，下記の通りである．

- Cortec-M3 に移植する場合の負荷試算
- 組み込み RTM および開発ツール群の課題抽出

特に前者は重要であり，組み込み MPU の場合，開発環境や開発言語の違いにより同一 MPU においても仕様が異なっている．そのため，各企業の採用する MPU において組み込み RTM に対応するための課題を明らかにすることは重要である．オカテック社のモータドライバでは，SH2 マイコンボードを採用しており，この作業を通じて，組み込みにかかる時間の算出を行った．本開発に用いるモータドライバを図 b-3-1 に示す．

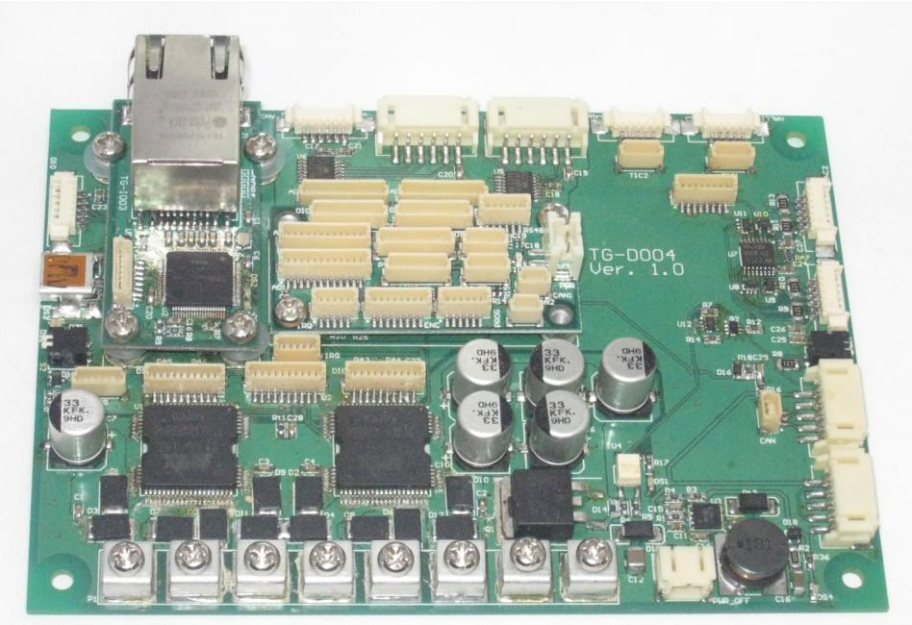


図 b-3-1　モータドライバ外観

miniRTCs の SH2 への移植作業では，仕様策定，CAN 用のドライバーの作成，RTC-Lite Manager の実装，ドキュメント制作まで含め，約 200 時間を要した．詳細を表 b-3-2 に示す．

表 b-3-2 移植作業内容および作業時間

作業内容	時間数
miniRTCs 要件確認・検討	56
CAN Device Driver 作成	32
CAN Device Driver デバッグ	32
RTC Lite Manager コーディング	64
ドキュメント制作	32
合計	216

要件の確認・検討や RTC-Lite Manager のコーディングにおいて時間がかかった要因として，ユーザが修正すべき要素と修正する必要のない一般的な要素について，きちんと切り分けができていなかった点が挙げられており，この旨を開発を通じたフィードバックとして株式会社セックにフィードバックを行った．

開発したモータドライバを実際にロボットへの応用も行っており，Yanboo-II への移植も行っている．Yanboo-II への移植に際して，モータの制御周期と miniRTCs の実行周期との間での整合性の点で，問題が挙げられたが，この点についても株式会社セックにフィードバックを行った．また，ソースについてはオカテック社および関連企業においてソースレベルでの公開を行える形にしている．

以上，組み込み RTM の展開を行っていく上での一つの参考となるデータを挙げることができ，さらには開発に関わる課題についても明らかにし，組み込み RTM 改善へと貢献を行った．

b-3-3 RTC-Lite および開発ツールの検証・評価

開発された RTC-Lite フレームワークに従った基盤通信モジュール用の RT ミドルウェアである miniRTCs を公開し，普及を行っていくためには実システム構築を通じた運用が不可欠である．そこで本開発項目では，miniRTCs を THK 株式会社で開発された SEED や株式会社アルゴシステムで開発された基盤通信モジュールに接続された RT 要素部品の制御へと適用していく過程で、miniRTCs や各種開発支援ツールの問題点の抽出，およびフィードバックを行った．また，最終的な成果として RT 要素部品の制御を実現し，miniRTCs を利用するうえでの導入事例の強化に努めた．

以下に，ミサワホーム住宅展示場におけるモデルハウスへの導入事例，産業技術総合研究所内に構築した実証住宅モデルへの導入事例について示す．

b-3-3-1 ミサワホーム住宅展示場モデルハウスへの導入

ミサワホーム住宅展示場に新規に建築されたモデルハウス内に，インテリジェント空調システムを構築した．構築したインテリジェント空調システムは，電動地窓，シーリングファン，エアコン，トップライトから構成され，電動地窓は1階と2階の2か所に各2個ずつ設置されており，計5か所に設置された RT 要素部品を対象とし，SEED および基盤通信モジュール，要素部品管理モジュールを用いた制御装置および RT コンポーネントの開発を行った．構築したシステム構成を図 b-3-2 に示す．

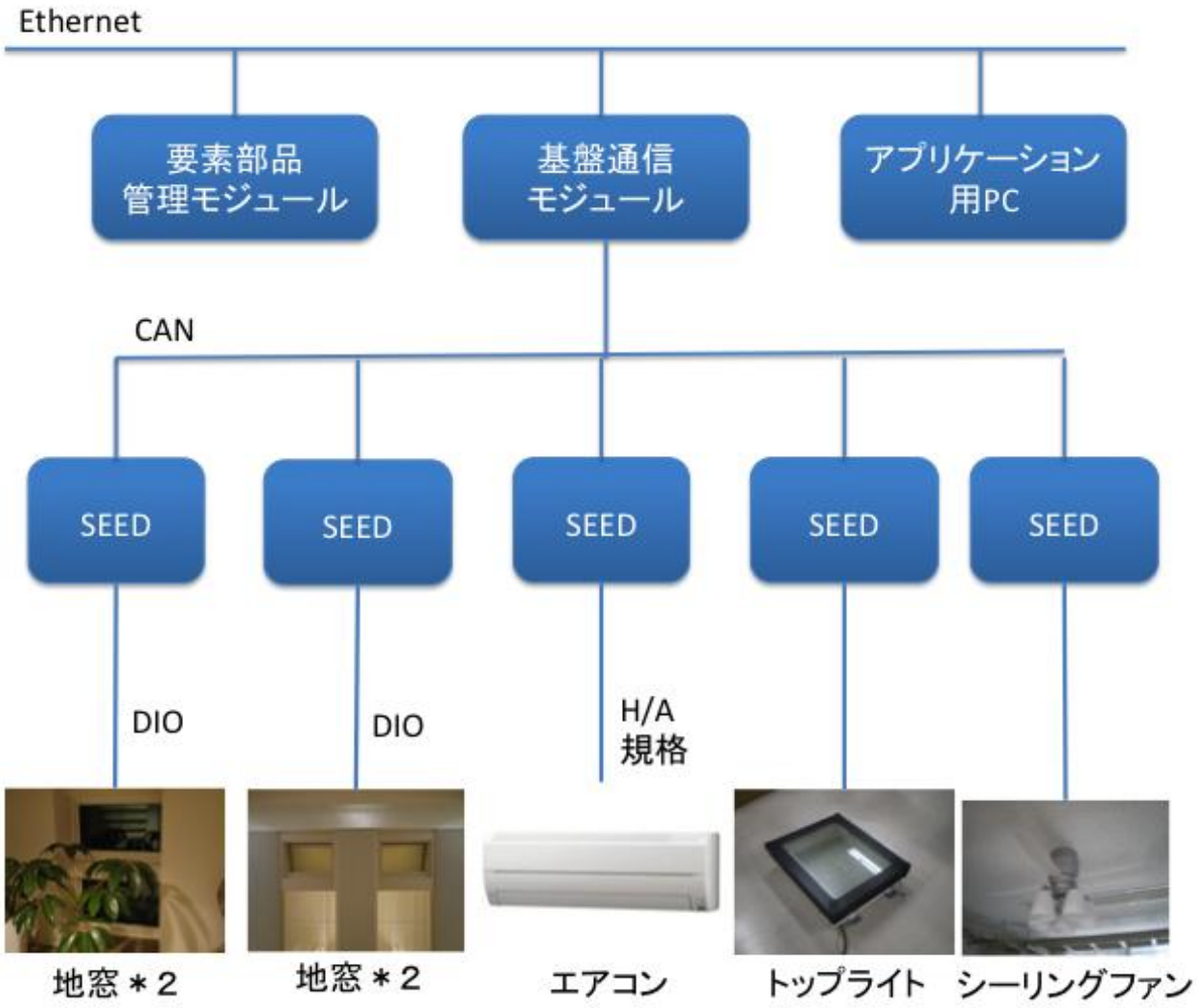


図 b-3-2 システム構成

本システムでは，各 RT 要素部品に付き，1 個の SEED を用いることとした．そのため，各 RT 要素部品の RT コンポーネントはそのまま制御を行う SEED と対応づけられる．制御ボックスとして，モデルハウス内の機器ルームに設置しやすいよう，図 b-3-3，図 b-3-4　　に示すようにインテリジェント空調システムに関わる制御系をすべて 1 つのボックス内に格納している．

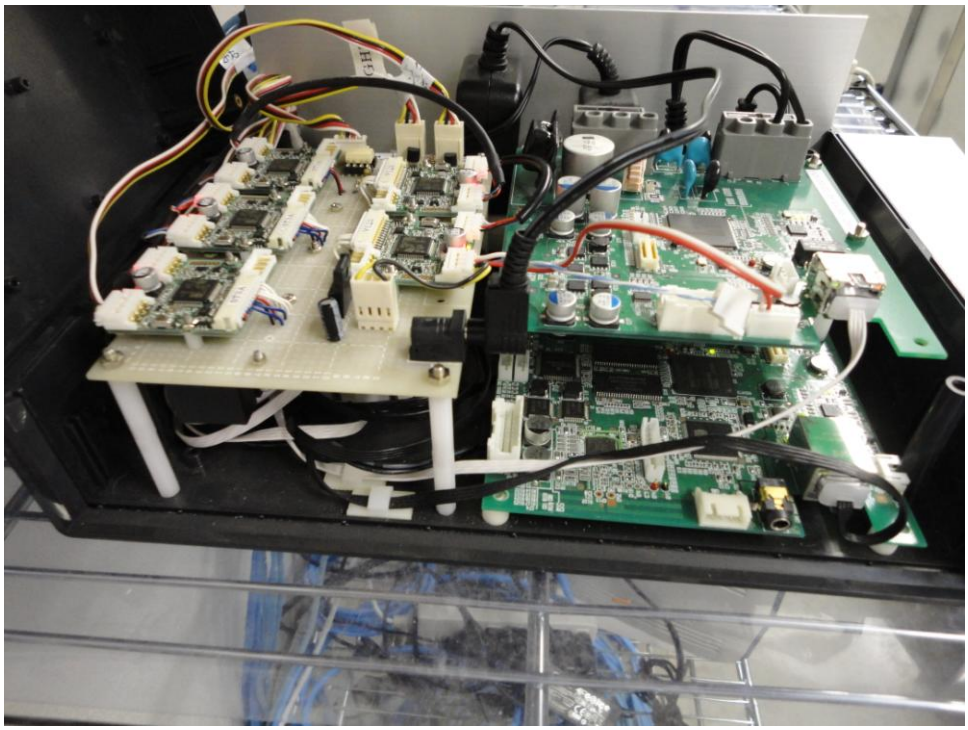


図 b-3-3　インテリジェント空調システム制御ボックス（正面図）

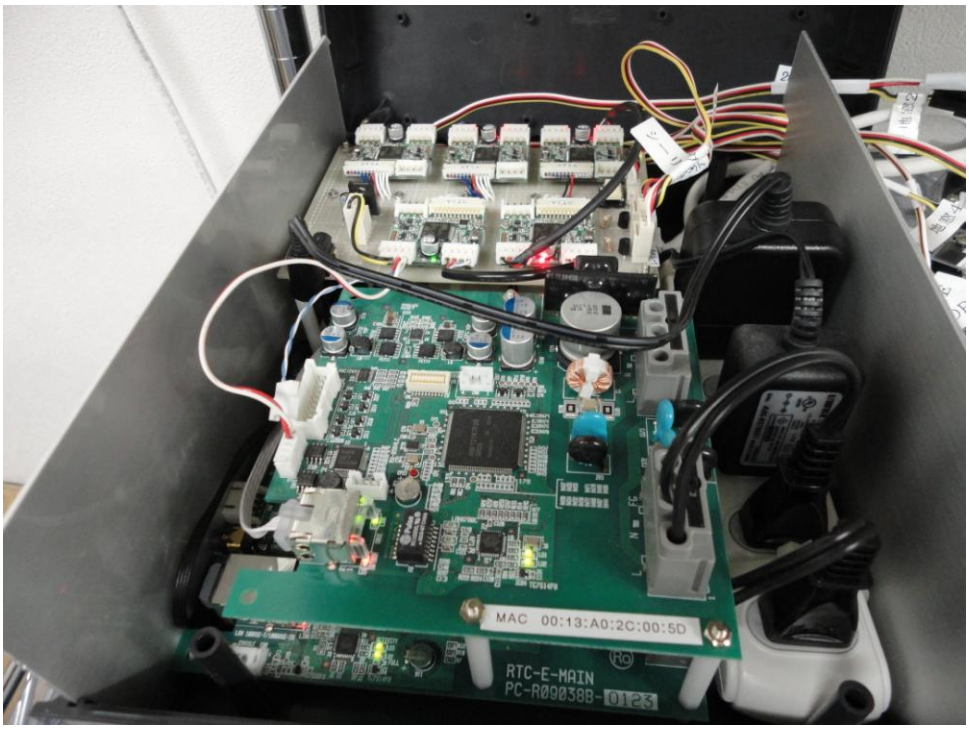


図 b-3-4 インテリジェント空調システム制御ボックス（側面図）

本システムでは，各 RT 要素部品の直接の制御は SEED から行っている．電動地窓は SEED からデジタル出力信号により開閉を制御している．また，各地窓にはタッチセンサが設置されており，センサの情報を獲得することで，開閉状態の確認を行っている．エアコンの制御は H/A 規格に従っており，SEED からフォトカプラを介して，H/A 規格に従った制御信号を送信し，ON/OFF の制御を行う．また，状態監視についても H/A 規格に従い，フォトカプラを介して，SEED ないで逐次情報の獲得を行っている．トップライト，シーリングファンの制御は赤外線リモコンを代替する形で行っており，トップライトの開/閉，シーリングファンの正回転，逆回転，停止の制御を行っている．赤外線通信になるため，各 RT 要素部品の状態監視を行うことができないため，各コンポーネント内でソフトウェア的に状態管理を行っている．

SEED 内部では，上述のルールに従った制御ロジックを miniRTCs により，RT コンポーネントとして実装しており，onExecute 内で，アプリケーションコンポーネントからの制御コマンド指令の受信，コマンドに従う制御，またアプリケーションコンポーネントへの状態の送信を行っている．

入出力は ECHONET 規格に準拠したフォーマットを用いており，上記 RT 要素部品の挙動に当てはまるように，表 b-3-3 のように規定している．

表 b-3-3 RT 要素部品制御コマンド

RT 要素部品	制御コマンド		状態確認	
電動地窓	OPEN		OPEN	
	CLOSE		CLOSE	
シーリングファン	STOP	0x31	–	–
	UP	0x32	–	–
	DOWN	0x33	–	–
トップライト	OPEN	0x41	OPEN	0x38
	CLOSE	0x42	CLOSE	0x30
エアコン	ON		ON	
	OFF		OFF	

b-3-3-2 産業技術総合研究所内実証住宅モデルへの導入

産総研に構築された実証実験スペースでは，ZigBee 通信を用いる RT ユニットを用いたシステム(システム1)，および THK 株式会社で開発された SEED PC を用いたシステム(システム2)により，実証実験スペース内の RT 要素部品の制御を実現した．

システム1では，RT ユニットに大阪大学において開発された RTC-Lite を搭載している．住宅内に設置する RT 要素部品は多岐に及ぶため，個別の機器に合わせたソフトウェア開発は負荷がかかり，開発コストの増加につながる．そこで，住宅設備機器に用いられる代表的な制御方式を取り上げ，その制御方式に従ったテンプレートを用意することにより，開発コストの削減を目指したモジュールデザインについて検討を行った．このモジュールデザインでは，下記の要素から構成される．

1） スイッチ入力などを検知するデジタル入力モジュール（DI モジュール）
2） 住宅設備機器の ON・OFF の制御を実現するデジタル出力モジュール(DO モジュール)
3） センサ情報を獲得するアナログ入力モジュール(AD モジュール)
4） アクチュエータや直接の調光制御などを実現するアナログ出力モジュール（DA モジュール）
5） 住宅設備機器の多くで採用されるリモコンを代替する赤外線リモコンモジュール（IR モジュール）

なお，(1)から(4)までのモジュールデザインについては，株式会社テクノロジックアートで開発された開発支援ツールにも反映されている．これらのモジュールデザインに基づき，なおかつハードウェアである RT ユニットにおいて，これらの機能を利用するための接続ポートを明示することで，RT ユニットを利用する RT 要素部品メーカーでは，ソフトウェア開発のコストが削減できる．結果として，RT ユニットへの結線作業のみで自社製品を RT ミドルウェア上で動作する RT 要素部品を簡易に開発することができるようになる．

本モジュールデザインに従い，産業技術総合研究所内実証住宅モデルへ IR モジュールとしてエアコン制御モジュール，テレビ制御モジュール，シーリングファン制御モジュール，AD モジュールとして温度センサモジュール，湿度センサモジュール，照度センサモジュール，人感センサモジュールを設置した．

IR モジュールでは，便宜上モジュールとしての機能を分けているが，アプリケーションコンポーネントより赤外線の発光パターンを受信し，受動的に赤外線発光を行うモジュールであるため，発光パターンが既知のデバイスであれば，どの住宅設備機器も制御可能であり，容易に既存の住宅設備機器を RT ミドルウェアで構築されたホームネットワーク上から制御することが可能である．AD モジュールについても便宜上，モジュールとして切り分けているが，接続されているセンサによる計測値がアナログ値で計測できるセンサであれば接続するだけでホームネットワークシステムに組み込むことが可能である．

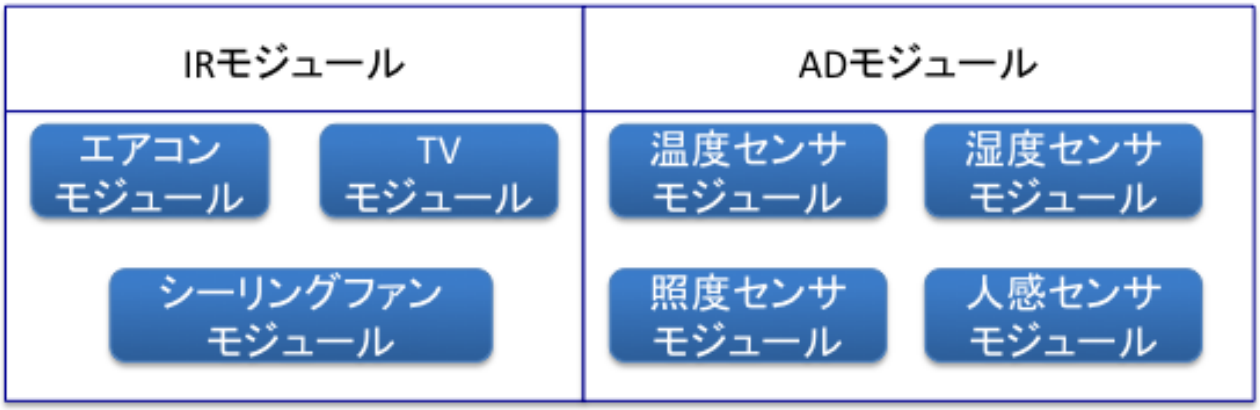


図 b-3-5 開発したモジュール

各モジュールは通信機能として，ZigBee による無線通信機能を有している．ZigBee では基地局となるコーディネータ，子機となるエンドノードから構成され，コーディネータは実証実験スペースではホームサーバに，エンドノードが上述の各モジュールに設置されている．各モジュールに設置される ZigBee 通信を行う通信モジュールとして Digi 社の XBee を採用した(図 b-3-6)．XBee は安価な ZigBee デバイスであり，ホビー用途をはじめに研究開発で幅広く利用されているデバイスである．XBee では，ZigBee Alliance の規格に従ったプロトコルを実装しているため，同一の規格を満たした製品であれば XBee 同士ではなくとも理論上相互通信が可能である．

図 b-3-6 Digi 社 XBee モジュール

ZigBee では，コーディネータを介して通信をするため複数のコンポーネントから同時にアクセスことは困難である．そこで，コーディネータを介する通信を管理・制御する枠組みとして，ZigBee Manager を開発した．従来開発してきた RTC-Lite は図 b-3-7 における左側の図のように，具体的に RT 要素部品を制御するデバイスとコンポーネントの間は TCP/IP のソケット通信で一対一対応がとられていた．ZigBee Manager においても従来版のリソースを利用できるようにするために，右図のように従来のリソースが利用できる形態で ZigBee Manaer の実装を行った．

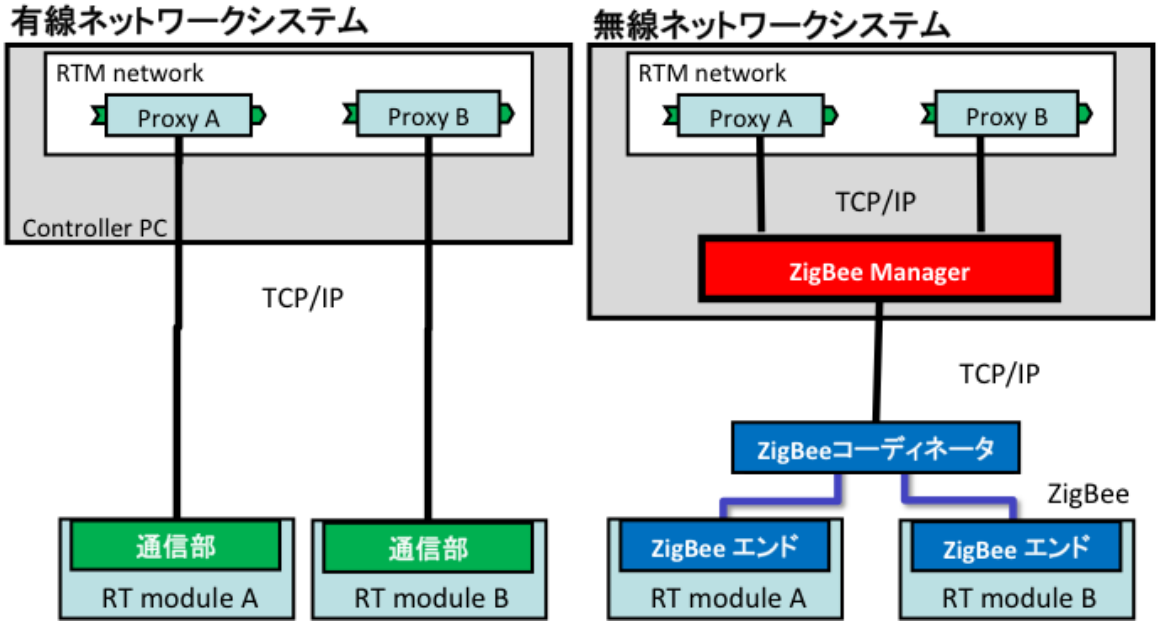


図 b-3-7 ZigBee Manager のコンセプト

ZigBee Manager では，新規に RT 要素部品が追加される場合において，自動的に RT コンポーネントが起動し，

RT ミドルウェアネットワークに参加できるように，プラグアンドプレイ機能が実装されている．

システム2として，玄関の施錠システムの構築を行った．従来の施錠システムでは，施錠ユニット，もしくはドア単体で閉じたシステムとなっていたために，他の住宅設備機器との間の連携が行えなかった．RT 要素部品として，玄関の施錠に関わる要素をモジュール化し，ネットワークを介して制御できるようになることで，室内の RT 要素部品との連携が実現できる．

玄関施錠システムでは，THK 株式会社の SEED PC を中心としたシステムを構築した．SEED PC は，OS を搭載し，OpenRTM-aist が単独で動作するだけでなく，CAN 通信も可能であるため，SEED との連携も可能である．SEED PC により制御される要素は，電動サムターン，ドアの開閉確認用の磁気スイッチ，認証用の RFID タグリーダから構成される．RFID タグリーダとして，DENSO WAVE 社の PR-550RKM を用いている．磁気スイッチとしてオムロン製近接センサ GLS を用いている．図 b-3-8 にシステム構成図を示す．

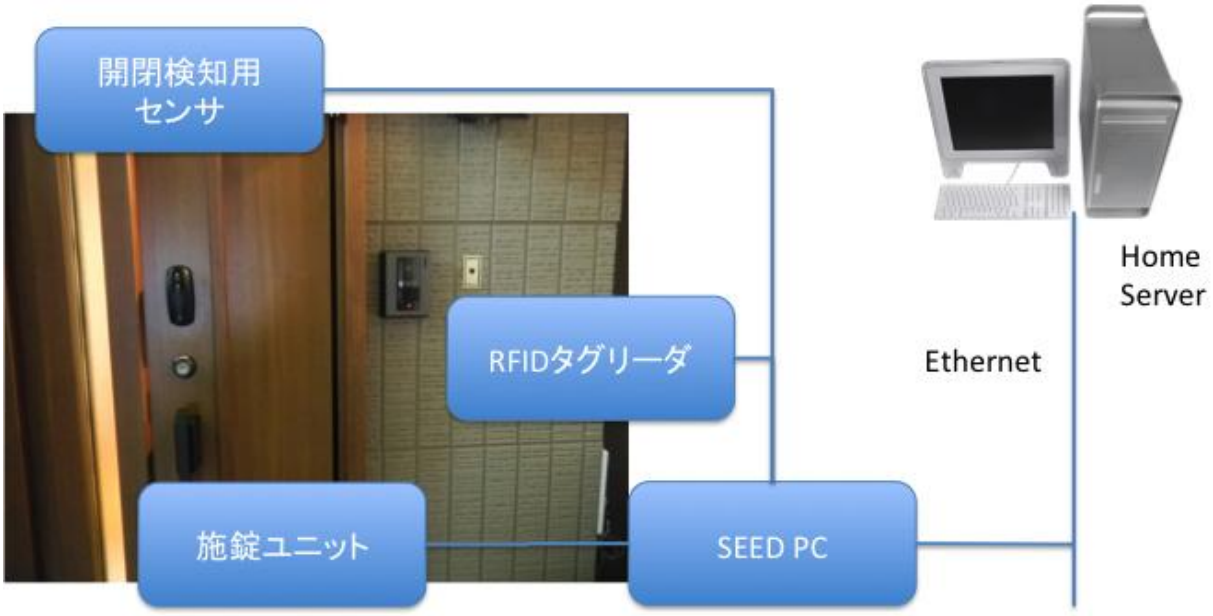


図 b-3-8 施錠システム構成図

こうして構築されたソフトウェア，ハードウェアの双方を用い，産業技術総合研究所実証住宅モデル（図 b-3-9 参照）において動作試験をおこなった．システム統合化については，c-1-6-4-c 節において詳細を報告する．

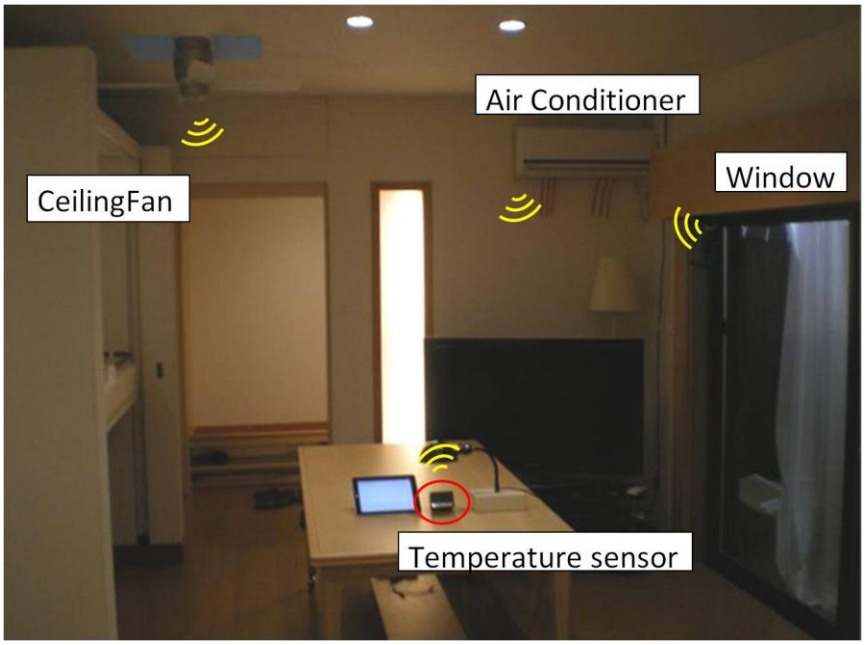


図 b-3-9 実験スペースの外観

b-3-4 RTC-Lite を利用しない直接的な RT ミドルウェアネットワークの開発

既存の CAN 対応製品を RT ミドルウェアの枠組みに参加させるためには，b-3-2 節で示したように直接制御コントローラのファームウェアを開発することが望ましい．しかしながら，プログラミング可能なシステムの場合はファームウェアの変更が可能であるが，製品としてブラッシュアップされている場合は，ファームウェア変更による手法は必ずしも適切ではなく，製品評価のプロセスまで含めると RT ミドルウェアに対応させるためのコストが大幅にアップする可能性がある．

そこで，既存の CAN 製品を RT ミドルウェアネットワークに参加させるための手法として，製品と RT ミドルウェアを媒介するブリッジユニットを用いる方法による開発を行った．ブリッジユニットを用いる方法の利点として，従来の製品に手を加える必要はないことが挙げられる．そのため，自社の CAN による通信プロトコルが既知という前提において，開発者に求められるものは RT ミドルウェアに関する知識となる．

こうした目的の検証のため，旭光電機において製品化が行われているエリア型人感センサを対象とし開発を行った．旭光電機のエリア型人感センサでは，従来の人感センサのようなスポット型の検出ではなく，光学系を改善することにより，多点計測を可能にしたセンサである．光学式のセンサは特に屋外環境での動作に不安があるが，旭光電機では，自動ドアの人感センサの開発を行ってきており，この問題点を克服し，多様な環境で用いることができることも本センサの特徴と言える．図 b-3-10 に本人感センサを示す．

図 b-3-10 エリア型人感センサ(旭光電機社製品)

本人感センサでは，外部インタフェースとして CAN 用通信線と電源(12V)が用意されている．ブリッジユニットでは，この CAN 通信をブリッジし，組み込み RT ミドルウェアである miniRTCs のネットワークに参加可能にする．図 b-3-11 にシステムの概要を示す．

図 b-3-11 開発システム概要

ブリッジユニットでは，基盤通信モジュールに対して，RT コンポーネントとして動作するための通信を，エリア型人感センサに対しては，動作コマンドの送信およびセンサからの獲得情報の送信が行われている．そのため，ブリッジユニットは，基盤通信モジュールとエリア型人感センサの間のプロトコル変換器として機能する．

ブリッジユニットを図 b-3-12 に示す．

図 b-3-12 ブリッジユニット

開発製品の実証として，本システムを産総研実証実験スペースに設置し，他のシステムとの連携について評価を行った．デモシナリオにおける位置づけとして，室内の人の有無の検出，また従来の人感センサと同様に人の存在するエリアの照明の点灯制御に対して用いた．設置状況を図 b-3-13 に示す．

図 b-3-13 産総研実証スペースへの人感センサの設置

実証実験を通じて，ブリッジユニットを用いる形態の実装においても，問題なくホームネットワークシステムに参加できること，またその下で動作することが確認できた．

ブリッジユニットを用いる形は，CAN 通信に関わらず，すべての通信プロトコルに対して適用可能なアプローチであり，既存製品を RT ミドルウェアネットワークに参加させる上で，有用なアプローチの一つであると言える．

本開発を通じて，開発元である旭光電機において，コスト試算を依頼した結果，RT ミドルウェアに関する知識がない場合において 600 万円，RT ミドルウェアに関する知識がある場合においては，300 万円のコストがかかると試算された．このことから，組み込み RT ミドルウェアに関するサポートを充実させていくことにより大きなコストをかけずに既存製品を組み込み RT ミドルウェアで構成されるホームネットワークへと参加させることが可能となり，多くの RT 要素部品を住宅用途に容易に応用できることが確認された．

(c)「RT 要素部品群による RT システムの開発・実証」
(c-1) 住宅環境における RT 実証システムの研究開発(委託先:株式会社ミサワホーム総合研究所)

c-1-1　概要

住宅への RT 普及の為にビジネスモデルと RT システムに要求される技術要件を整理し、それに基づき設備部品の開発と実証用建物の構築を行い、主に“インテリジェント空調システム”と”インテリジェント・ウィンドウ(窓)システム“の2つの機能を実現してデモンストレーションを行う事で実用性を確認した。以下に本課題の目標並びに研究開発成果および達成度をまとめる。

表 c-1-1　目標およびそれに対する研究開発成果ならびに達成度

目標	研究開発成果	達成度
1)分散型 RT 要素を利用した住宅用ホームオートメーションのビジネスモデルを検討.(出典:基本計画 p15)	(1)従来の住宅のビジネスモデルに基づき、RT が導入した場合のシステムの優位性を従来ある自動化の既製品と比較してまとめた。	(1)達成
(2)RT システムを住環境に応用する場合にハードウェアに求められる要件,そして RT の専門家以外の技術者或いは居住者によるアプリケーション開発を実現する為に開発環境に求められる要件をまとめる.(出典:基本計画 p15)	(2)RT システムに要求される仕様を、ユーザー企業となる住宅メーカーからの見地からまとめた。	(2) 達成
(3) RT システムを構築する際の仕様(住宅用設備開発フレームワーク等)を固める.(出典:基本計画 p16)	(3)住宅メーカーが顧客から要求されているニーズをまとめ、その中で RT システムで実現可能な機能を抽出した。	(3)達成
(4) 実証システムとして提示している「住宅用インテリジェント空調システム」「インテリジェント・ウィンドウ(窓)システム」のシステム設計仕様を固める。(出典:基本計画 p16)	(4)実証システムとして構築する「住宅用インテリジェント空調システム」「インテリジェント・ウィンドウ(窓)システム」について連携に必要な設備機器(RT 要素部品)をまとめ、システム設計を行った。	(4)達成
(5) RT 要素部品および開発環境を利用し,実証システムを構築し評価を行う。(出典:基本計画 p16)	(5)実証システムとして産業技術総合研究所実験室内のモデルルームおよびミサワホーム住宅展示場モデルハウスに導入し、住宅システム全体としての評価を行った。	(5)達成

c-1-2　住宅の現状と課題

ホームオートメーションに代表される様に、住宅への電動設備部品の導入とネットワーク化は以前から行われてきたが、単なる遠隔操作スイッチを寄せ集めた集中制御盤的な色彩が強く、ユーザーメリットが明確でない事から広く受け入れられていないのが現状である。その様な状況の中、住宅に導入される設備機器は年を追って増えて壁面に取り付けられる専用コントローラの数は片手に余る状況となり、操作が煩雑なだけでなく外観、意匠上も決して望ましい状態ではない。また、それぞれのコントローラは単独で動作し連携させる事ができないのが通常で、場合によっては暖房しながら冷房する様な矛盾する動作が発生する可能性もある。オーディオ・ビジュアルに関わるいわゆる黒モノ家電製品は、メーカーが独自のネットワーク化を進めた事もあり異なるメーカーの機器を接続して有機的に活用する事ができない状況が見られる他、同一メーカーであっても白物家電と呼ばれる機器とは連携しないの

が一般的であった。家電メーカーの多くの商品開発は残念ながら機器毎に細分化されているか連携範囲が極めて限定的である為に機器連携で実現できる使い勝手や機能について十分な検討を行えていないといえるだろう。空間全体、家全体を考慮した機能（以下、家の機能）は、家電の機能とは別個に考える必要がある。

太陽光発電システムや燃料電池システムやバッテリーシステム等の新しい電気設備は、複数が同時に設置されるケースが予想されるにも関わらず相変わらず個別の制御を前提としており最適なマネジメントは実現できそうにない。プラグインハイブリッド自動車や電気自動車等の連携についても議論が始まったばかりである。また、LED照明等の次世代の省エネ照明装置が急速に普及しつつあるが操作は相変わらずの壁スイッチや赤外線リモコンによるものが殆どで、他機器との連携は考慮されていないのが現状である。

住宅側に目を向けると、化学物質アレルギー（ホルムアルデヒド等）の問題に端を発した法改正による機械換気装置の設置義務化や火災報知器の設置義務化等、住宅を取り巻く環境は大きく変化しつつある。また、建物の高断熱化・高気密化によって窓ガラスは複層となり、防犯性向上の為にそれらの一方が合わせガラスとなるケースも増えてサッシの重量が大幅に増加しつつある事に加えて、枠材の強度向上によりサッシサイズの大型化が進んでいる事から、子供や高齢者による操作が困難となる様な新たな問題も発生している。外出時の戸締り確認や火の元確認等の手間は旧態依然で全く改善の兆しも見えず、日差しや通風を上手にコントロールする事でより省エネ空間を実現できるにも関わらず、そうした手法は殆どの家で相変わらず活用されていないのが現状である。

c-1-3 想定するビジネスモデルと対象

日本国内の新築住宅件数は年間100万戸を割り込んで久しく、既存住宅約4500万戸に比べると極めて少ない状況にある。普及を考えるならば既存住宅を対象に加えてシステム開発を行う必要がある。また、家電メーカーがネットワーク化に積極的に取り組んでいる情報家電を始めとした装置を対象とするのは、メーカー側に競争領域との認識もあり導入としては適当でなく、家電メーカーの関心は低いが住宅の基本機能を実現する為に必須な装置を対象とするのが効果的と考えられる。これにより“家電の機能”と“家の機能”が明確に区別され、家電製品が“家の機能”を利用する或いは家が“家電製品の機能”を利用するといったWinWinの関係が築ける事が期待される。

家の機能を実現する装置の一つ、開口部（窓やドア）の電動化した部品（電動シャッター等）には複数のサッシメーカーが共通に利用している電動化ユニットが見られる。即ち、RT対応の開口部品を個別に開発するのではなく、開発者が使いやすい電動化ユニットを供給する事ができれば多くのサッシメーカーに採用される可能性があると考えられる。更に現状では電動化ユニットはオリジナルの通信規格を採用しており、異なるメーカーでは全く互換性が無いという問題があるので、メーカーを問わずに接続できる、リーズナブルで高品質なRT電動化ユニットがあれば普及の可能性は更に高まるだろう。装置メーカーはある程度のスキルを有してはいるがRTに関する知識は殆どないのが一般的で、簡便な専用ツール化、キット化（例えば、電動窓開発キット等）を行い、開発が従前に比べて容易になる様な工夫も欲しい。

また、装置単体ではなく複数の装置を組み合わせて実現する機能（例えば外出時に玄関を施錠すると全ての窓が自動的に閉じて施錠される機能等）は、従来は殆ど見られないもので、住宅メーカーを始めとした様々なプレーヤーによって全く新しい機能が提供される可能性がある。通常、こうしたプレーヤーはRTやITが必ずしも専門ではないので、機能開発が素人にでも出来るような設計支援ツールやシミュレーションツールが必要となるだろう。

こうした検討を基に図 c-1-1 に示す様なビジネスモデルを想定した。スマートハウス等で検討されているビジネスモデルも同様のものであり、連携を図っていく事も容易であると思われる。図中の〇印に茶色と緑色の2色があるのは住宅用に開発されたRT関連機器とそれ以外の一般的RT関連機器を模式的に区別して示したものである。様々なチップやセンサを用いてオープンな要素部品管理モジュールや基盤通信モジュール等が開発され、設備機器メーカー等がそれらを用いてRTに対応した設備機器類を開発し市場に投入することを想定している。こうして提供される設備機器類は予めネットワーク対応しているので連携動作によって新しい機能を提供できる。そしてそれらはソフトウェアを変更することで改変可能なので居住者のライフステージやライフスタイルの変化に応じて柔軟にカスタマイズ可能な住宅を実現できる。

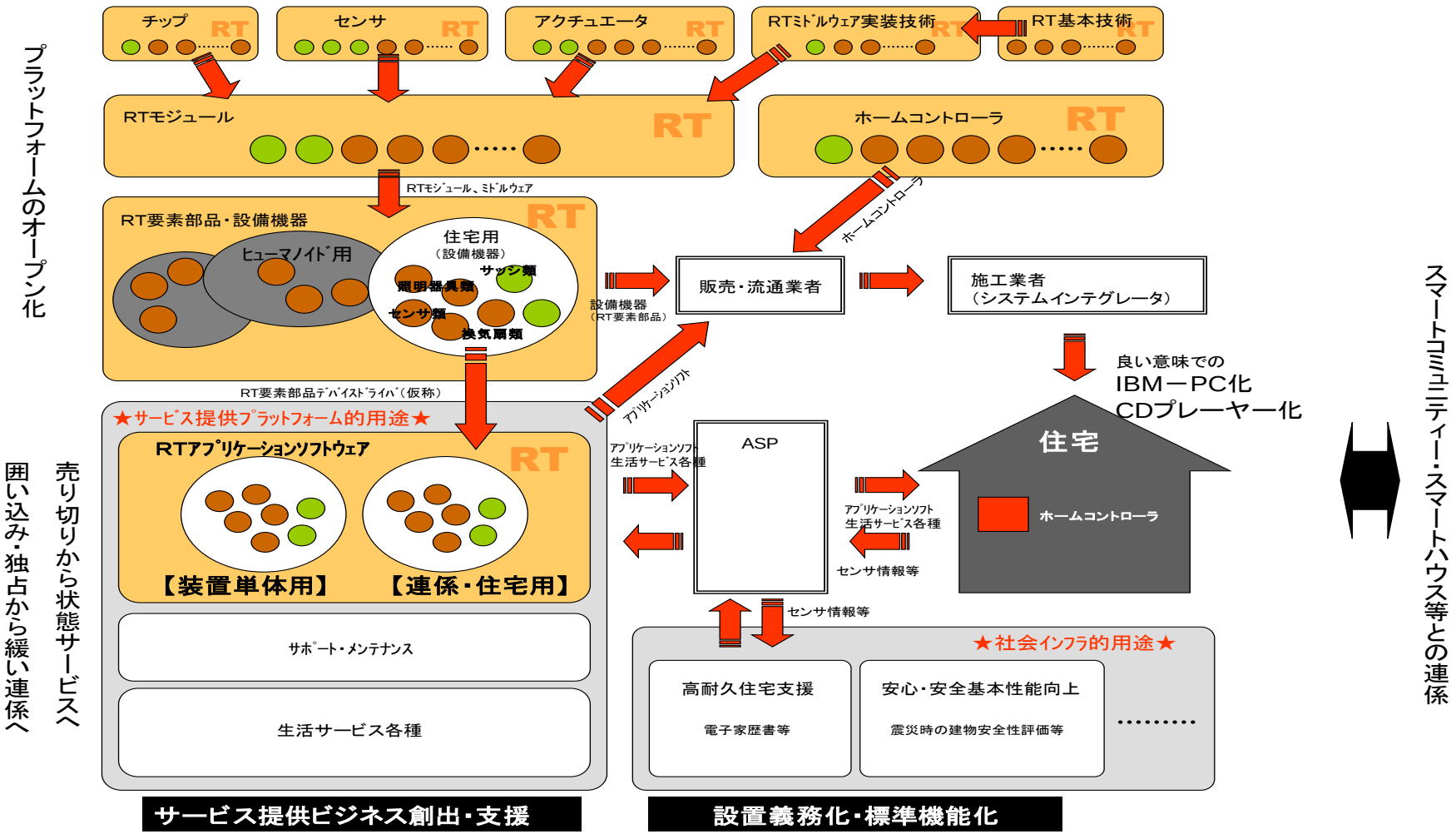


図 c-1-1　住宅におけるRTビジネスモデル

c-1-4 RT に求められる技術要件

RTが住宅に普及する為の技術要件の検討例を表 c-1-2 に示す。
性能が高い事だけでなく、開発が容易な事、施工が容易な事、運用が容易な事が必要で、システム開発に当たって関連するそれぞれの要件について具体的な検討を行い妥当性を評価する。

表 c-1-2 住宅へのRT普及の為の技術要件

	技術要件の例	分類
1	設備部品の開発が従来に比べて容易になる	開発
2	設備機器が複数連携する機能(家機能)を開発しやすい	開発
3	オープンなシステムである	開発・運用
4	部品の耐久性が高い(屋内環境で最低10年程度)	性能
5	低コスト(目的に対しリーズナブルなコストである)	性能・施工・運用
6	動作安定性に優れ、停電時復旧が確実に行われる	性能
7	起動時間が十分に短い(従来の家電製品並み)	性能
8	サイズ(目的に対して不都合のないサイズ、形状である)	性能
9	省エネ(目的に対しリーズナブルなエネルギー消費である)	性能
10	設備機器単体の不具合が他の機器に影響を与えない	性能
11	停電時には普通の家(窓が開かない等の不都合がない)	性能
12	設備部品の追加・撤去・交換が容易	施工
13	メンテナンスが容易	施工
14	設置に特別な工事が不要	施工
15	意匠に配慮した設置が容易	施工・運用
16	プラグアンドプレイ	施工・運用
17	責任分解点が明確で切り分けが容易	運用
18	動作のモニタリングやログ取得ができる	運用
19	メーカーによらずに接続可能(マルチベンダ)	運用
20	バグフィックスを含むバージョンアップ対応	運用
21	持続可能性(代替を含め将来的に入手可能な部品を用いる)	運用
22	〃 (ハードウェアが変更されても継続利用できる)	運用

c-1-5 住宅に求められる機能

住宅に求められる機能例を示したのが、表 c-1-3 である。大きく分けると、機器単体の動作による機能と複数の機器が連携することで実現される機能がある。主に前者は設備機器メーカーが自社製品開発の際に想定するもので、様々な商品が販売されている。後者は”家の機能”に該当するが、商品化されているものは殆どない。

表 c-1-3 住宅に求められる機能の例

住宅に求められる機能の例	分類
火災の検知・報知（音で火災を知らせる）	単体
不法侵入検知・報知（音で不法侵入を知らせる）	単体
防犯カメラ（不審者の映像を表示・記録する）	単体
自動開閉ドア（人を検知してドアを自動開閉する）	単体
パワーアシスト窓（窓の開け閉めを補助する）	単体
シーン照明コントローラ（複数の照明器具を予め設定した状態に制御する）	単体 （同種の装置の連携）
インテリジェント空調システム（屋外と屋内の温熱環境に応じて、通風とエアコンを適切に切り替えて省エネ空調を行う）	連携（家の機能）
目覚まし制御（目的の時間に快適に起きられる様に、照明や日射遮蔽装置等を適切にコントロールする）	連携（家の機能）
一括施錠（外出時に玄関錠の施錠のみで家中の戸締りが完了する）	連携（家の機能）

c-1-6　RT の住宅への応用と有効性評価

c-1-6-1　実証する機能の想定

タブレット型端末を用いた汎用リモコン機能、音声によるガイダンス機能、玄関電気錠の一括施錠機能に対応したセキュリティ機能等、様々な機能が考えられるが、今回の RT を用いた住宅機能の実証にあたり、”インテリジェント空調システム”と”インテリジェント・ウィンドウ（窓）システムを対象として選定した。前者は“家の機能”、後者の内パワーアシストは“単体の機能”、一括施錠は“家の機能”に該当する。

c-1-6-1-a　インテリジェント空調システム

インテリジェント空調システムとは電動トップライト、電動地窓、シーリングファン、エアコンを屋内と屋外の温熱環境に合わせて適切にコントロールして省エネな空調を実現するものである。コンセプトを図 c-1-2 に示す。

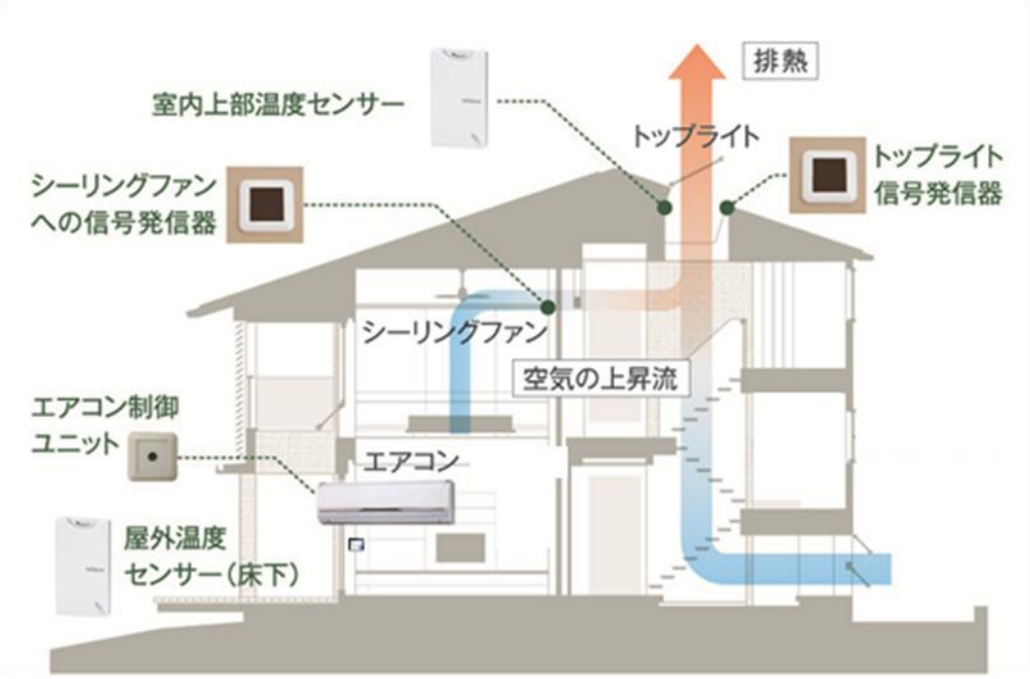


図 c-1-2　インテリジェント空調システムのコンセプト

今回想定した基本制御ロジックの概要は図 c-1-3 の通りで実住宅を想定した建物（以下、建物B）に採用した。トップライトについてはインテリジェント空調の動作に関わらず雨の吹き込みを防止する必要があるので、トップライトの機能の中に降雨センサとそれに連動した機能を組み込むのが適当である為に降雨センサの状態は表中には記載していない。

こうした装置を用いる事で夏期のエアコン利用時間が大幅に低減される（東京地域の試算で300時間程度）事が分かっており、消費エネルギー及び CO2 排出量の削減が期待できる。

下限温度	Tmin	寒さ防止（デフォルトは20℃ ※変更の可能性あり）
上限温度	Tmax	高温防止（デフォルトは30℃ ※変更の可能性あり）
設定温度	Tset	ユーザー設定値（エアコン設定温度より高い事、デフォルト値は28℃ ※変更の可能性あり、Tmin<Tset<Tmax）
		※設定は0．1℃刻み

パラメータ

tout	外気温度
troom	居室温度
ttop	屋内上部温度（トップライト近傍温度）

制御対象と状態

対象	状態	
電動トップライト	(OPEN,CLOSE)	
電動地窓	(OPEN,CLOSE)	
シーリングファン	(UP,STOP,DOWN)	※UPは上向き送風、DOWNは下向き送風、STOPは停止
エアコン	(ON,OFF)	※エアコンはH/A端子を用いたON,OFF制御のみ

状態遷移

状態	機能	過冷却防止	強制空冷	自然空冷	保冷	自然空冷	強制空冷	強制空冷	保冷	保冷	強制排熱	強制排熱	撹拌冷房
	状態番号	1	2	3	4	5	6	7	8	9	10	11	12
条件	tout	<Tmin	<Tmin	<Tmin	<Tset	<Tset	<Tset	<Tset	≧Tset	≧Tset	≧Tset	≧Tset	≧Tset
	troom	<Tset	≧Tset	≧Tset	<Tset	<Tset	≧Tset	≧Tset	<Tset	<Tset	≧Tset	≧Tset	≧Tmax
	ttop	－	<troom+3	≧troom+3	<troom+3	≧troom+3	<troom+3	≧troom+3	<troom+3	≧troom+3	<troom+3	≧troom+3	－
制御	トップライト	CLOSE	OPEN	OPEN	CLOSE	OPEN	OPEN	OPEN	CLOSE	CLOSE	OPEN	OPEN	CLOSE
	地窓	CLOSE	OPEN	OPEN	CLOSE	OPEN	OPEN	OPEN	CLOSE	CLOSE	OPEN	OPEN	CLOSE
	シーリングファン	STOP	UP	STOP	STOP	STOP	UP	UP	STOP	STOP	UP	UP	DOWN
	エアコン	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
備考													※状態12からは状態1,4,5にのみ遷移

※状態遷移のタイミングは温度変化が緩慢な事から1分毎（変更の可能性あり）とする

初期動作

対象	状態	
電動トップライト	CLOSE	
電動地窓	CLOSE	
シーリングファン	STOP	
エアコン	OFF	※窓開閉には一定の時間が必要なので初期動作から制御開始まで2分（変更の可能性あり）待つ

図 c-1-3 インテリジェント空調システムの制御ロジック例（実住宅を想定した建物：建物B）

全体実証用建物（以下、建物A）では、屋内設置である事と複数の機器の連携動作を効果的にデモンストレーションする事を目的とするので、図 c-1-4 に示す様な単純なロジックを実装した。

デモ用センサ温度	地窓	トップライト	シーリングファン	エアコン
20℃未満	CLOSE	CLOSE	OFF	OFF
20℃以上25℃未満	OPEN	OPEN	OFF	OFF
25℃以上30℃未満	OPEN	OPEN	ON	OFF
30℃以上	CLOSE	CLOSE	OFF	ON

※設定温度は変更の可能性あり

図 c-1-4　インテリジェント空調デモ用制御ロジック例（全体実証用建物：建物A）

c-1-6-1-b　インテリジェント・ウィンドウ（窓）システム

インテリジェント・ウィンドウ（窓）システムとはパワーアシスト機能と一括施錠機能を有するシステムである。窓の開閉方式には最も一般的な引き違い窓を選択し、新築ではなく既存の建物への装着を想定する。

自動ではなくパワーアシストとするのはホームオートメーションの黎明期に人の接近を検知して自動的に開閉するいわゆる自動ドアが好意的に受け止められなかった事を考慮した。一般の窓と同様に手動で開閉を行うが、その際に重量のある大型サッシでも指先一本で動かせる位の負荷で操作できる様にパワーアシストを行う。同時に、挟み込みや打撃による危険を防止する為に適切にブレーキがかけられる。

現在の住宅の多くは外出時等には全ての開口部の戸締りを目視で確認し、手作業で施錠を行う必要がある。一部の商品にクレセントの状態のモニタリングと遠隔操作が可能なものが登場しているが、サッシが閉じられていない状態では施錠は行えず従来と同様の対応が必要であった。パワーアシストのサッシ開閉機能と施錠機能を組み合わせる事でサッシの開閉を含めて完全な遠隔操作が可能となるので一括施錠機能として実装を目指す。

インテリジェント・ウィンドウ（窓）システムのこうした機能（遠隔開閉、遠隔施錠等）を活用する事で、前述のインテリジェント空調システム等の新しいアプリケーションが容易に提供できる様になる。また、例えば人感センサを組み合わせる事で不在室の窓は常に施錠した状態を保つ事が可能なので、侵入盗の侵入手口の3割を占める施錠忘れを防止できる等、防犯性を向上される新しい機能開発にも役立つと考えられる。

今回の実証では屋外に面した開口部を対象としているが、これらの機能は屋内の建具についても流用可能な事はいうまでもない。

c-1-6-2 実証に用いる機器の開発

設備機器類は現状ではRT対応していないので、実証に合わせて改造しなければならない。設備機器類を入手し、本プロジェクト関係機関にRT対応の為の改造を依頼し動作を確認した。

表 c-1-4 に全体実証用建物の実証に用いた主な機器を、表 c-1-4 に実住宅を想定した建物に用いた主な機器を示す。尚、開発欄にチェックの無い機器は改造前に設置されていた RT 設備の流用や改造が不要な機器であることを示している。

表 c-1-4 建物Aに用いる主な RT 対応設備機器

	設備機器、センサ	数量	開発	備考
1	インテリジェント・ウィンドウ（窓）システム	2	レ	複数の小型通信ドライバ及び基盤通信モジュールで構成
2	電動トップライト	1	レ	小型通信ドライバで直制御
3	電動地窓	2	レ	小型通信ドライバで直制御
4	電動シャッター	2	レ	小型通信ドライバで直制御
5	電動ブラインド	2	レ	小型通信ドライバで直制御
6	エアコン	1	レ	IR 或いは HA 制御
7	シーリングファン	1	レ	Zigbee 制御
8	玄関電気錠システム（カードリーダ、電気錠）	1	レ	小型 PC モジュール利用
9	照明コントローラ	1	レ	ホームコントローラ制御
10	サーキュレータ	1	レ	基盤通信モジュール（PLC）
11	電動昇降ベッド	1		無電圧接点（3 芯）制御
12	ダイニングテーブル（アクティブキャスタ付き）	1		Zigbee 制御
13	TV	1		IR 制御
14	開閉検知センサ	1		Zigbee 端末
15	人感センサ	6	レ	基盤通信モジュール（PLC）
16	温度センサ	5程度	レ	Zigbee 端末
17	ホームコントローラ（デスクトップPC）	1	レ	照明コントローラ対応

表 c-1-5 建物Bに用いる主な RT 対応設備機器

	設備機器、センサ	数量	開発	備考
1	電動トップライト	1	レ	IR 制御
2	電動地窓	2	レ	無電圧接点（3芯）制御
3	電動内部窓（電動地窓）	2	レ	無電圧接点（3芯）制御
4	エアコン	1	レ	HA 制御
5	シーリングファン	1	レ	IR 制御
6	温度センサ	（3）	（レ）	その他システムから Web-API で情報取得
7	ホームコントローラ(ノートPC)	1	レ	WEB－API対応
8	設定・操作端末（iPad）	1		WEBを介して操作

c-1-6-3 実証用建物の構築

c-1-6-3-a 全体実証用建物（建物A）

独立行政法人産業技術総合研究所（つくば）の研究施設屋内に建物Aを構築した。これは既存の住宅モデルルームを今回の実証用に改修、改良を加えたものである。概観を図 c-1-5 に、プランを図 c-1-6 に示す。実証建物は研究所建物の室内にあるが、配線等の便宜を考えて、既存壁との間に30cm程度の空間を設けている。防火上の制限で壁を天井面まで立ち上げられない壁面には、ブロック状の仮壁をそれぞれに用意して、デモンストレーション時等には塞ぐ事ができる構造とした。また、既存のシステム床を生かして建物が設置されているので床下空間を配線に利用することも可能である。天井面も同様にシステム天井となっていたが、実際の住宅に近づける為にシステム天井面をクロス仕上げした。

プラン（間取り）は標準的な住宅とは異なるが、LDKに相当する屋内を主要な実証スペースとして各種設備機器を配置した。

図 c-1-5 建物Aの概観

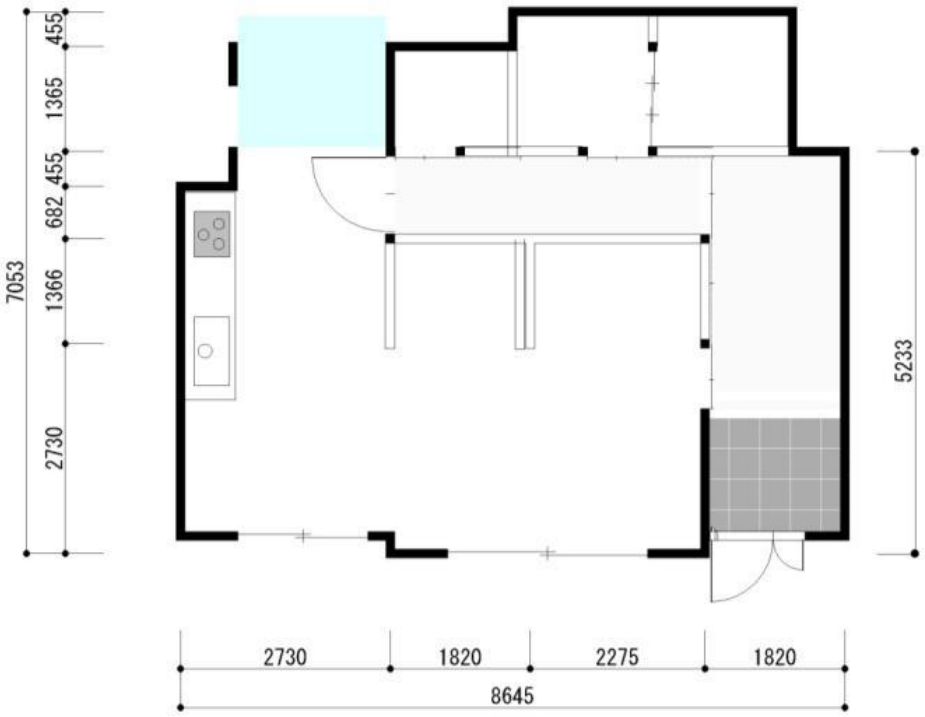


図 c-1-6 建物Aのプラン

c-1-6-3-b 実住宅を想定した建物（建物B）

実際の住宅により近い環境での動作を確認する目的でミサワホームの展示場を用いて“インテリジェント空調システム”を実装し、検証を行う事とした。外観を図 c-1-7、プランを図 c-1-8 に示す。展示場という性格上、装置を露出設置する事が許されず実証終了後に現状復帰が必要な事から制御機器類は地下機械室に配置する。参考に 2 階上部の電動屋内窓（地窓）からトップライトに通じる通風経路が分かる内観写真を図 c-1-9 に示す。

図 c-1-7 建物Bの外観（ミサワホーム展示場）

図 c-1-8 建物Bのプラン

図 c-1-9　インテリジェント空調システムの通風経路例

c-1-6-4　RT システムの構築

c-1-6-4-a　基本方針

既存の住宅を含めて普及を考えるので、ネットワークの構成としては設備機器間の通信には特別な配線工事が不要な電力線通信（PLC）と無線（Zigbee）の利用を原則とし、設備機器内部の通信にはCANを想定して検討を行った。図 c-1-10 にネットワーク構成のコンセプトを示す。図から分かる様に、商品提供の形態（複数の窓の連動等）によっては物理的に離れた部品間でもCANを利用する場合もあると考えられる。
また、ホームコントローラが停止しても個別の機器が動作を続けられる様にする為に、集中型ではなく分散型のシステム設計を行うのが望ましい。責任分解点を考慮すると提供する個別商品のレベルでメーカーが責任を持った機能開発を進めた上で、ネットワークからの動作要求に応えて動作するといったゆるい連携が望ましいだろう。

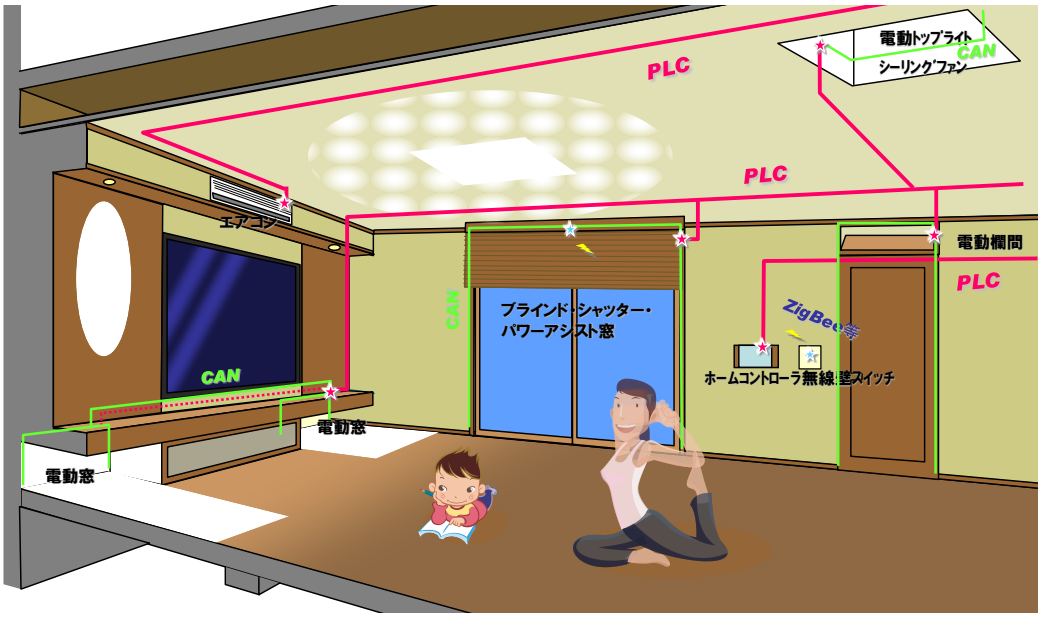
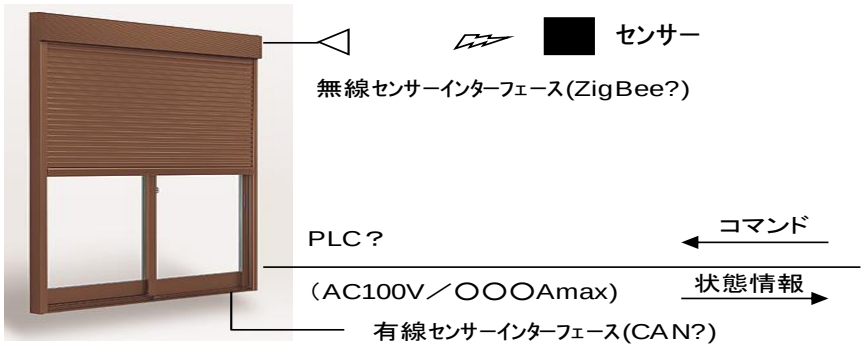


図 c-1-10　ネットワーク構成のコンセプト

RT関連モジュールの消費電力の現状値を考慮すると、現時点では特に高い能力を有する通信モジュールの利用台数は 1 部屋に 1 台程度に制限するのが現実的と考えられる。すなわち、分散型でかつ、低消費電力という両面の機能を実現するには、高機能型の通信モジュールである要素部品管理モジュールを部屋に一台程度、低コスト版な基盤通信モジュールを各機器に組み込むという構成にし、全体のコスト、消費電力を押さえながら、分散型システムとする構成が望ましい。加えて、居室には開口部が設けられ、そこには可動部品が集中するのが一般的であるので、電源が容易に確保出来ることから、開口部に PLC 接続された基盤通信モジュールを設置し、そこに小型通信ドライバやZigbeeエンドデバイスが繋がる様なネットワーク構成とするのが効果的といえる。参考に図 c-1-11 に開口部に関連する設備部品やセンサを示す。
ホームコントローラは分電盤内部等にインターネット接続用の機器等と同様に設置するのが望ましく、それらを考慮した形状及び熱対策が望まれる。盤内部は一般に密閉されるので発熱量の大きい機器を設置は避ける必要がある。また、省エネを考慮すると、いたずらに高機能を追求するのではなく、バランスのとれたスペックのハードウェアを選定しなければならず、耐久性を考慮するならファンやハードディスク等の可動部品の無い機器の方が適当である。



【関連設備部品】
・雨戸
・シャッター
・オーニング
・ルーバー
・網戸
・ドア
●窓
・ブラインド
・カーテン
・ロールスクリーン
等

【補助部品】
●電気錠
・カメラ
・個人認証装置
・太陽電池
・セルフクリーナ
等

【関連センサー】
●開閉センサー
●施錠センサー
●挟み込み検知センサー
●人感センサー
・ガラス破壊検知センサー
・ピッキング検知センサー
・温度センサー
・湿度センサー
・風速センサー
・ほこりセンサー
・CO2センサー
・煙センサー
・VOCセンサー
・降雨検知センサー
・照度センサー
・日射センサー
・結露センサー
・汚れ検知センサー（ガラス等）
・可視光通信用センサー
・赤外線通信用センサー
・電力センサー
等

図 c-1-11 開口部に関連する設備機器やセンサの例

c-1-6-4-b 住宅プランと機器の配置

RT関連部品を住宅内部に設置する場合、それらは剥き出しではなく、既存の機器に内蔵されるか、収納等の一部に設置してデザインへの影響を最小限に抑えるのが望ましい。既存住宅の開口部を想定すると、カーテンボックスの利用が効果的で、今回の建物Aの開口部 2 箇所で採用した。外観は図 c-1-12-1 の通りである。

図 c-1-12-1　カーテンボックスの外観

特徴として,制御用機器の取り付けや現場調整を考慮し,極力取り外し可能な構造とした.正面の平板カバーを外すと内部は上下段の 2 部構成になっている(図 c-1-12-2 参照).

上段奥は制御機器を収納する.メンテナンスの作業性を考え取り外し可能で左右可動式の板を制御機器配置用に設置した.上段手前は内部制御機器の配線を収納する.仕切りの設置は配線が機器の邪魔にならない簡単な構造なので最適と考えた. また,仕切り板は照明架台としても機能する.中間板の一部は明かり取りの為に切り欠きをほどこした(図 c-1-12-3、c-1-12-4 参照).カーテンボックス外部下方より制御機器へ配線の引き回しがあることから,中間板の左右には開口を設けた(図 c-1-12-5 参照).電源スイッチはアクセスが安易な場所に設置する(図 c-1-12-2 参照).将来設備機器の追加や交換の際,簡単に RT システムネットワークへ読み込みが可能なプラグアンドプレイに対応させる為,中間板にコネクタジャックを設けた.設備機器のコネクタを差すのみで設備機器を RT ネットワークシステム化できる.

デモンストレーションの計画当初はプラグアンドプレイにコネクタジャックを用いる予定だったが、通常のコンセントを用いたデモンストレーションに変更された.下段奥の壁面にはインテリジェント・ウィンドウ駆動用機器が配置(図 c-1-12-4 参照),手前は中間板に取り付けたブラインドが位置する(図 c-1-12-2、c-1-12-3 参照)

図 c-1-12-2　カーテンボックス内部（その1）

図 c-1-12-3　カーテンボックス内部（その2）

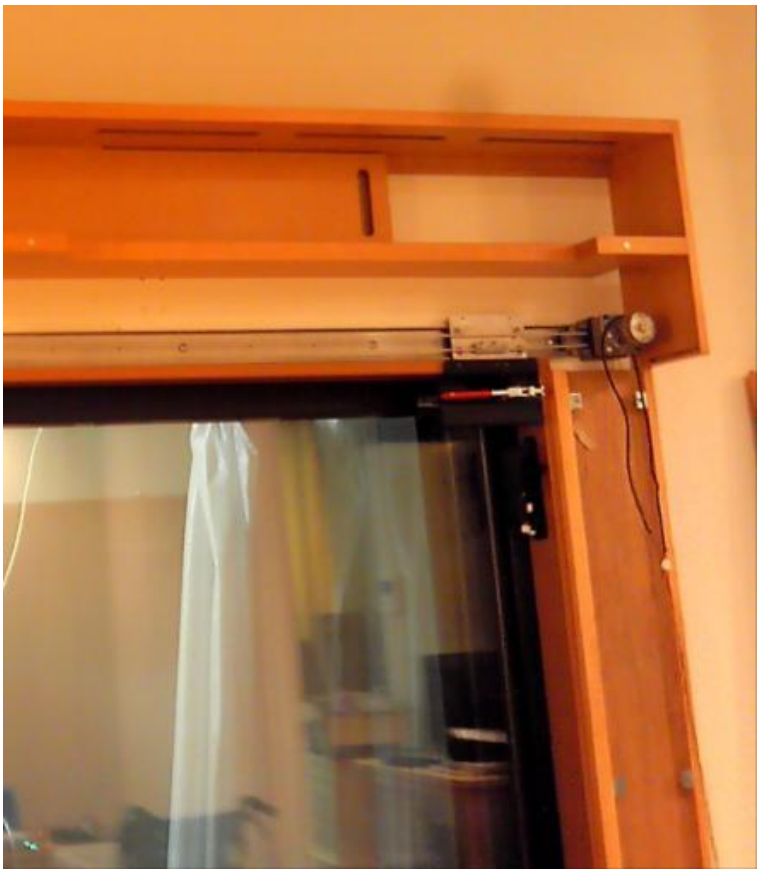


図 c-1-12-4　カーテンボックス内部・配線ボックス内部

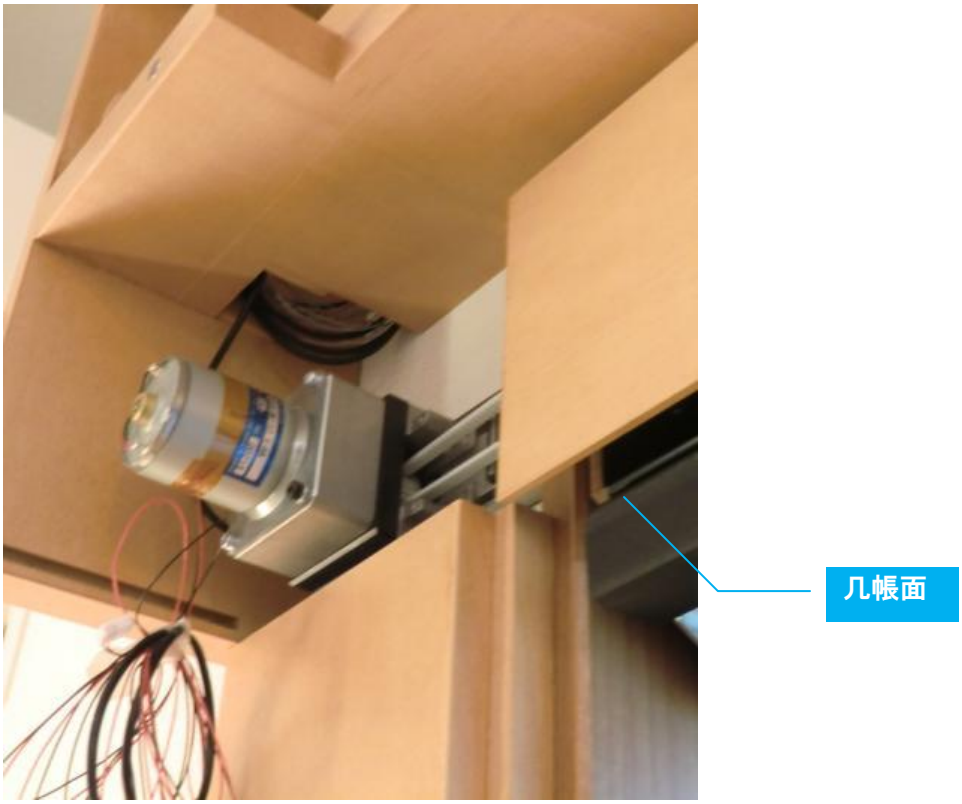


図 c-1-12-5　カーテンボックス内部と柱状カバーの取り合い部

建物Aでは電力線通信（PLC）を用いたネットワークを構築する。窓下部よりカーテンボックス制御機器までの配線は,配線スペースを窓枠外側に設けた（図 c-1-12-4, c-1-12-5 参照）.普通,電気配線は新築であれば壁内に収納される.しかし,新築であっても窓周りには芯材を使用する為,壁内収納は難しい.また,既存住宅も対象とする事から,追加取り付けが可能な構造とした.従来は馴染みのない建具だが,インテリアに調和するようカーテンボックスと合わせた意匠とした（図 c-1-12-1, c-1-12-5 参照）.配線のスペースのカバーはマグネットにより着脱が可能である.また,カーテンボックス正面カバーも同様だが几帳面細工（角そのものは残すように,両側に段をつけたもの）を施すことによりカバーの存在を目立ちにくくした.（図 c-1-12-5 参照）本プロジェクトでは窓枠外側のみに留まったが,幅木部も同様の加工をすることにより宅内配線を目立たなくさせるとともにコンセントの配置も容易になる等,汎用性が高い手法である。こうした配慮を行いながら、関連する設備機器類、RTモジュール類の設置を行った。

照明器具及び主要設備機器の配置は図 c-1-13、図 c-1-14 の通りである。RT 関連モジュールは図 c-1-15 の位置に配置されている。見かけ上基盤通信モジュールの台数が多いのは用いる人感センサがエリア型で情報処理量が多いのと、PLC による接続を志向した為に各センサに基盤通信モジュールを用いた事によるもので、実質2台（小型 PC を基盤通信モジュールで代用するなら3台）が必要数である。

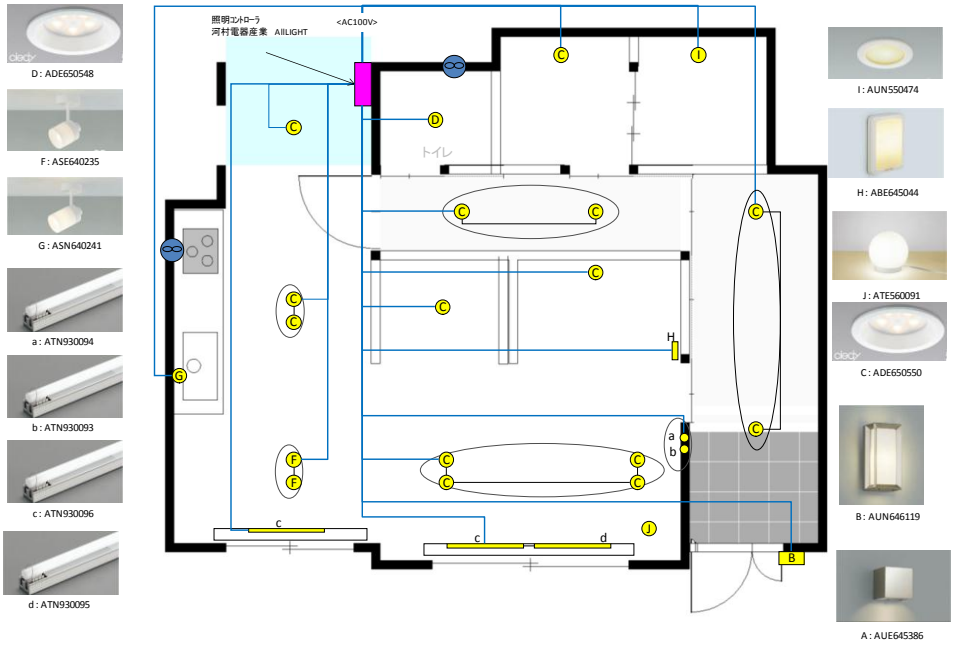


図 c-1-13 建物Aの照明器具配置図

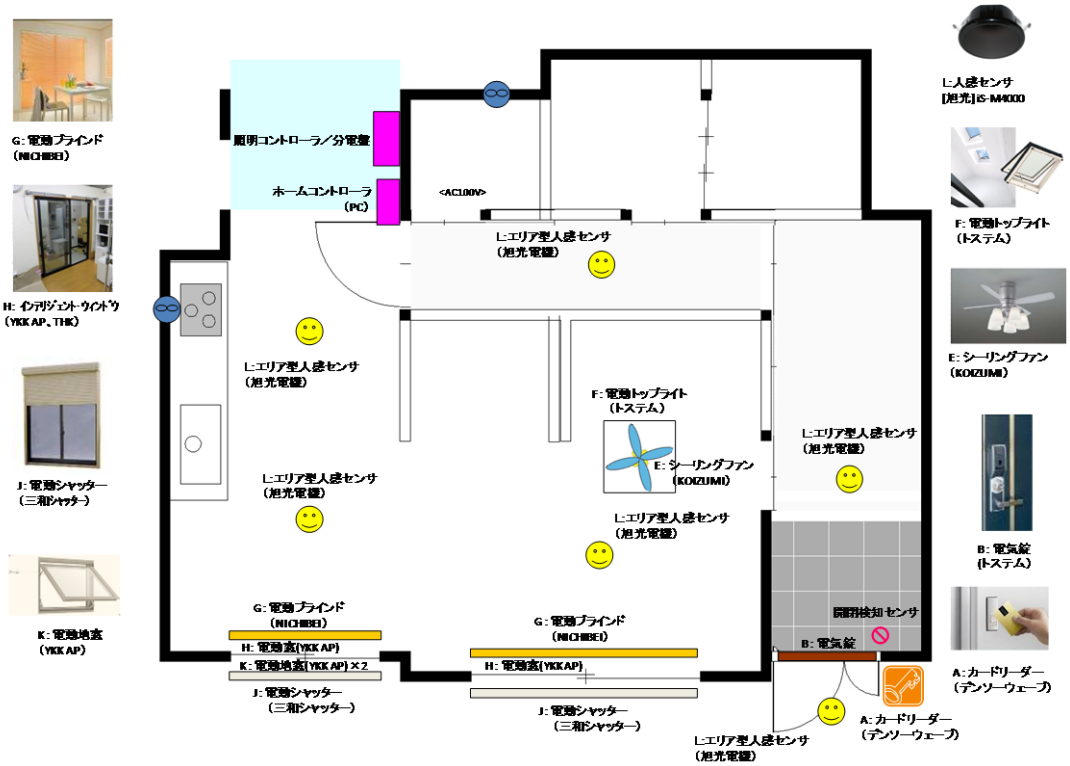


図 c-1-14　建物Aの主要設備機器配置図

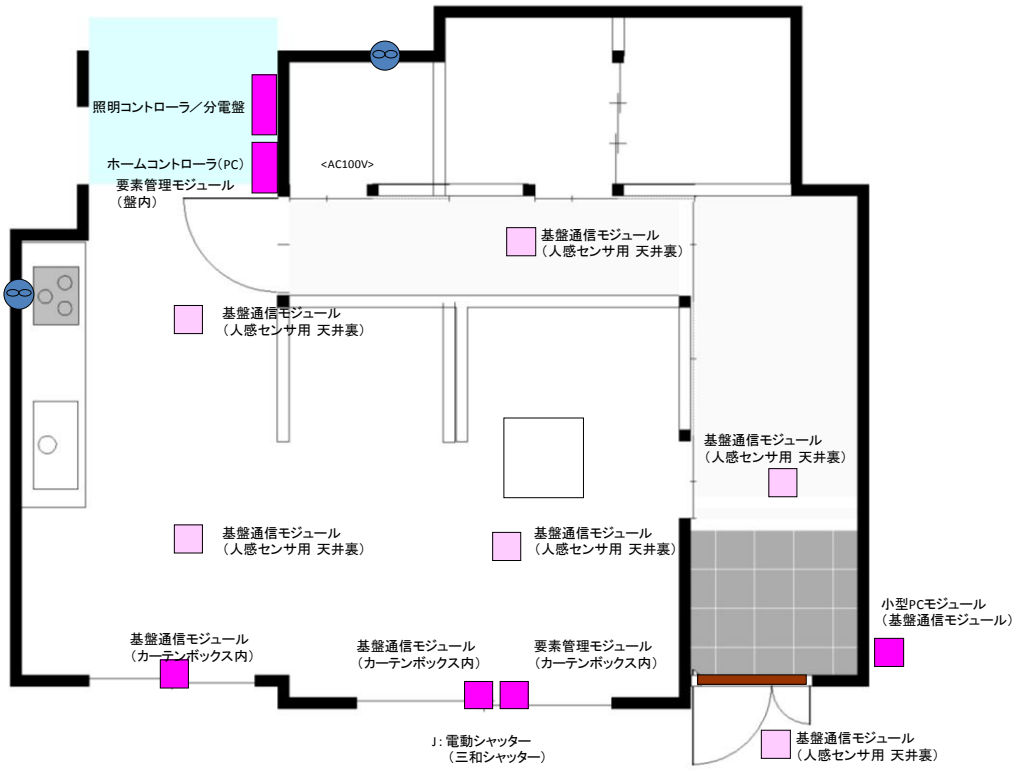


図 c-1-15　建物Aの主要 RT モジュールの配置

インテリジェント・ウィンドウシステムに必要な RT 関連部材はカーテンボックスの内部及び関連設備表面に設置されている。尚、今回の実証に用いたインテリジェント・ウィンドウシステムは後付けを想定したもので、施錠は通常のクレセントではなく戸先に設けた施錠ユニットで行う方式となっている。これは稼動するサッシ部分への電源供給を避ける目的もある。パワーアシストに関連する操作部は戸先部に設けられており、電源を確保できない事から無線信号（Zigbee）を用いてシステムに接続されている。Zigbee 受信用のモジュールはカーテンボックス内部に設置され基盤通信モジュールと接続されている。

建物Bの設備機器の配置は図 c-1-16、それらの接続は図 c-1-17 の通りである。撤去を想定している為に、RT 関連モジュールは全て1つのケースに収められて地下機械室内に設置されている。（平面図には機械室の記述はない）

図 c-1-16 建物BのRT関連設備機器配置

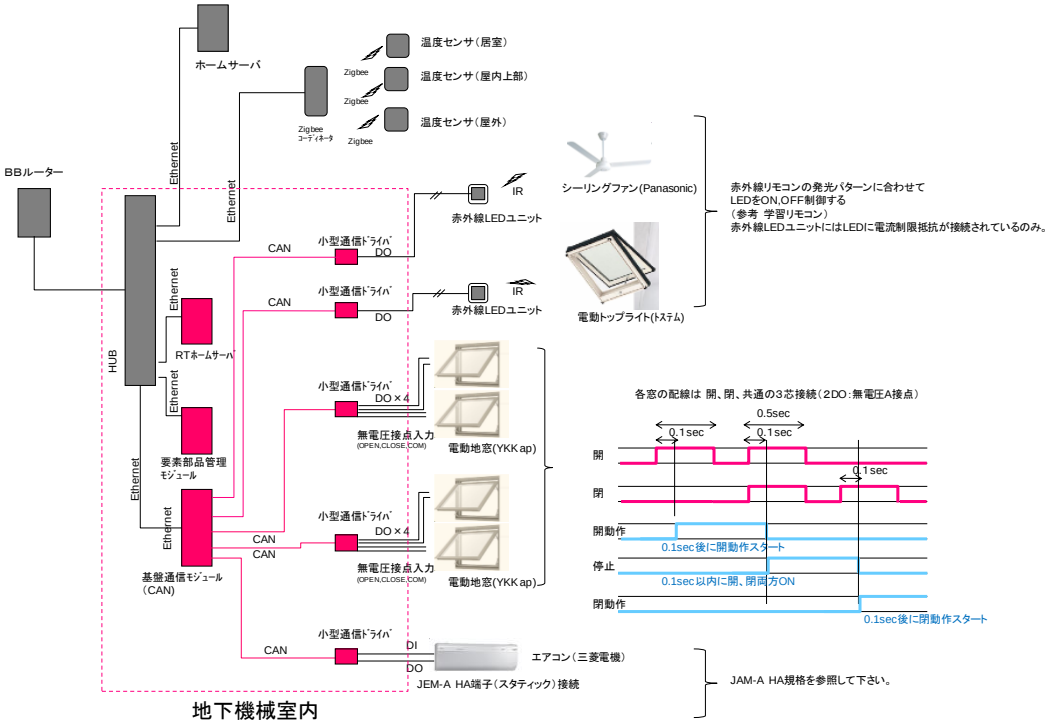


図 c-1-17　建物BのRT関連設備の接続

図 c-1-18　建物Bの地下機械室のRT部材写真

c-1-6-4-c システム構成

建物AのPLCを用いたシステムの構成（上位層）を図 c-1-19 の通りである。

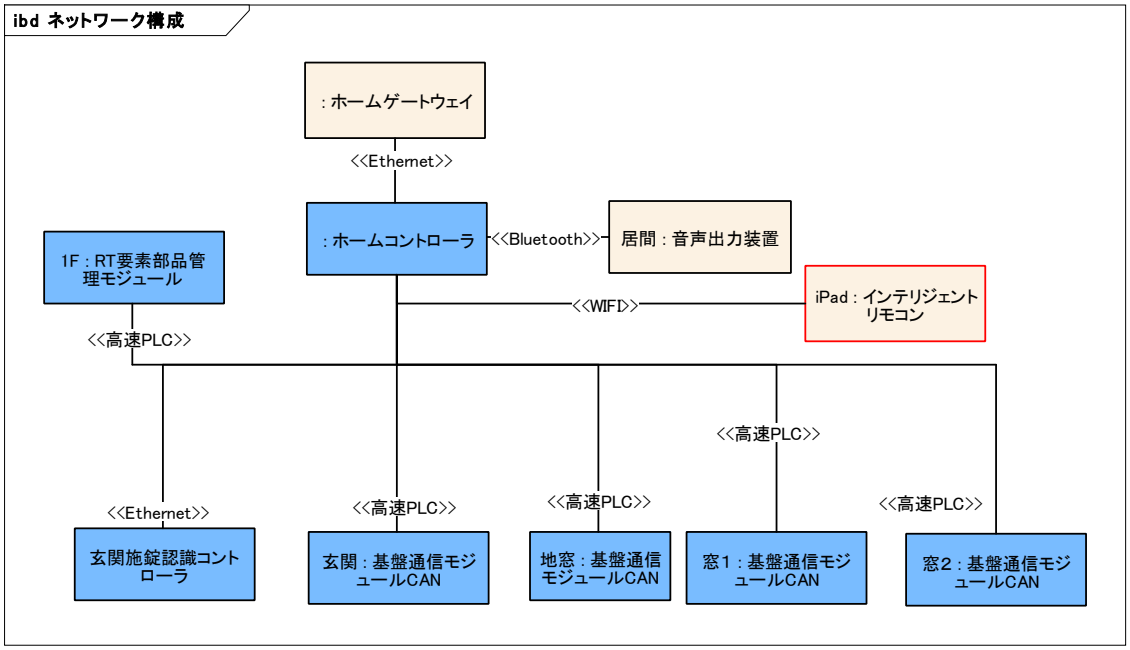


図 c-1-19　建物Aのシステム構成図（上位層）

外部インターネットはホームゲートウェイを経由してホームコントローラに接続される。ホームコントローラは住宅アプリケーション向けの RT コンポーネントが動作し、各機器の状態信号ならびに制御指令といった住宅システム全体の連携動作に係るマネージメントを行っている。ここで、ホームコントローラは、従来の RT ミドルウェア（OpenRTM-aist）で動作しており、内部の RT コンポーネントは OpenRTM-aist ベースで構築されている。一方、低コスト版である基盤通信モジュールはスペック上 OpenRTM-aist は組み込めないため、軽量版 RT ミドルウェアである miniRTC（株式会社セック開発）が組み込まれている。すなわち、OpenRTM-aist と miniRTC の相互通信を可能とする機能が相互の間には必要となる。そこで、要素部品管理モジュールは OpenRTM-aist を組込、基盤通信モジュールで組み込まれている miniRTC とのブリッジ機能を有することで、OpenRTM-aist と miniRTC 間の相互通信を実現している。このような 2 種類の階層をつくることで、システム全体のコストを抑えている。また、基盤通信モジュール以下のシステム構成を窓部について、図 c-1-20 に示す。

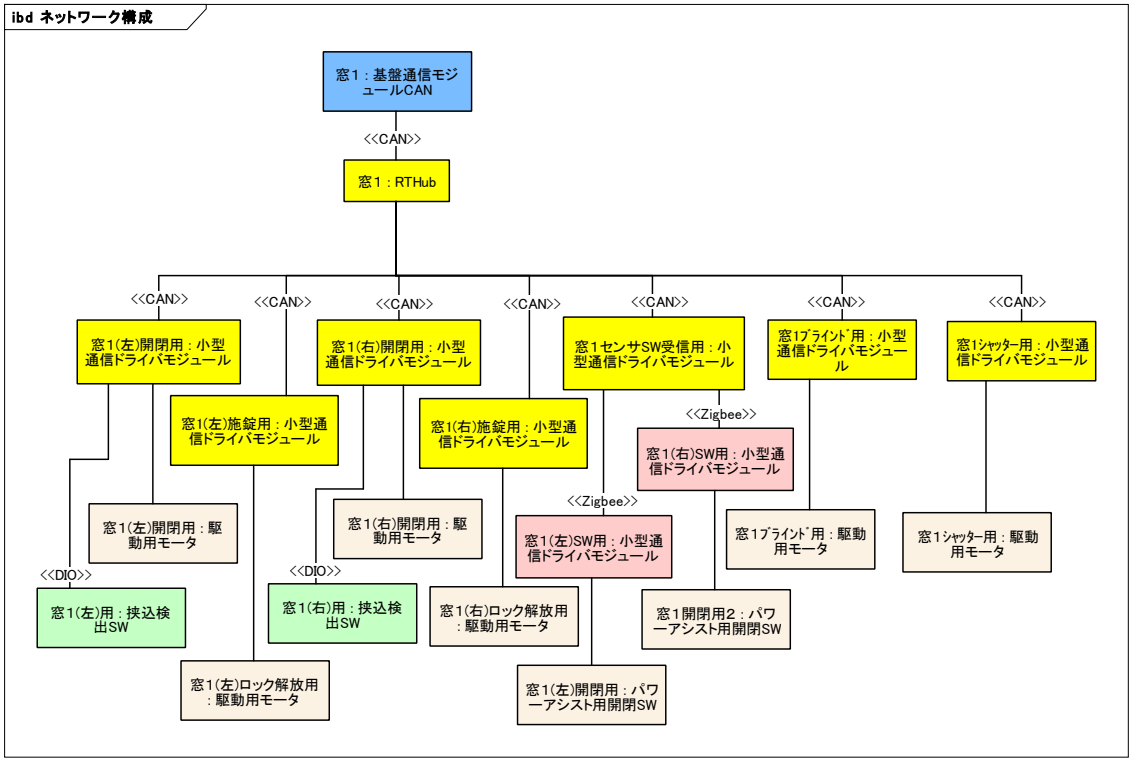


図 c-1-20　建物Aの窓部システム構成図（下位層）

窓部における基盤通信モジュールより下位システムを例に取ると、基盤通信モジュールの CAN のネットワーク上に小型通信ドライバモジュール（THK 株式会社開発）が接続される。小型通信ドライバモジュールは各窓やブライド、シャッター等を駆動するモータに接続され、基盤通信モジュールからの指令値によって動作する。特に、モータといった物理的動作を有する設備を接続する際は、ネットワークの安定性から有線である CAN ネットワークを利用するべきである。一方、安定性よりも設置の利便性を必要とされる場合は、Zigbee といった無線ネットワークを利用することも可能である。

全体システムとしては、ホームコントローラから、RT 要素部品管理モジュールを経由して基盤通信モジュールに制御指令値が流れ、最終的に各設備が小型通信ドライバモジュールによって動作し、その状態情報を同様の経路でホームコントローラに返すと言ったフィードバック系を形成している。住宅システムとしての各設備の連携動作はホームコントローラによる集中管理になっているが、基盤通信モジュール以下の小型通信ドライバモジュール間同士ならば OpenRTM-aist である上位層を介さなくとも miniRTC の中だけで連携動作可能となっている。例えばホームコントローラといった上位層がダウンしたとしても、挟み込み検知スイッチは機能する仕組みとなっているために、安全に係る機能は維持することになる。すなわち、安全に係る機器は、その関連する設備を制御する基盤通信モジュールの下部に設置している。以上のように、各設備の連携動作と安全上必要な動作機能は分けると言った分散処理系を構築していることで、システム全体のコストを抑えると共に、システムの安全性向上も実現している。図 c-1-21 に、その他のシステムを含んだ建物Aにおけるシステム全体の構成図を示す。

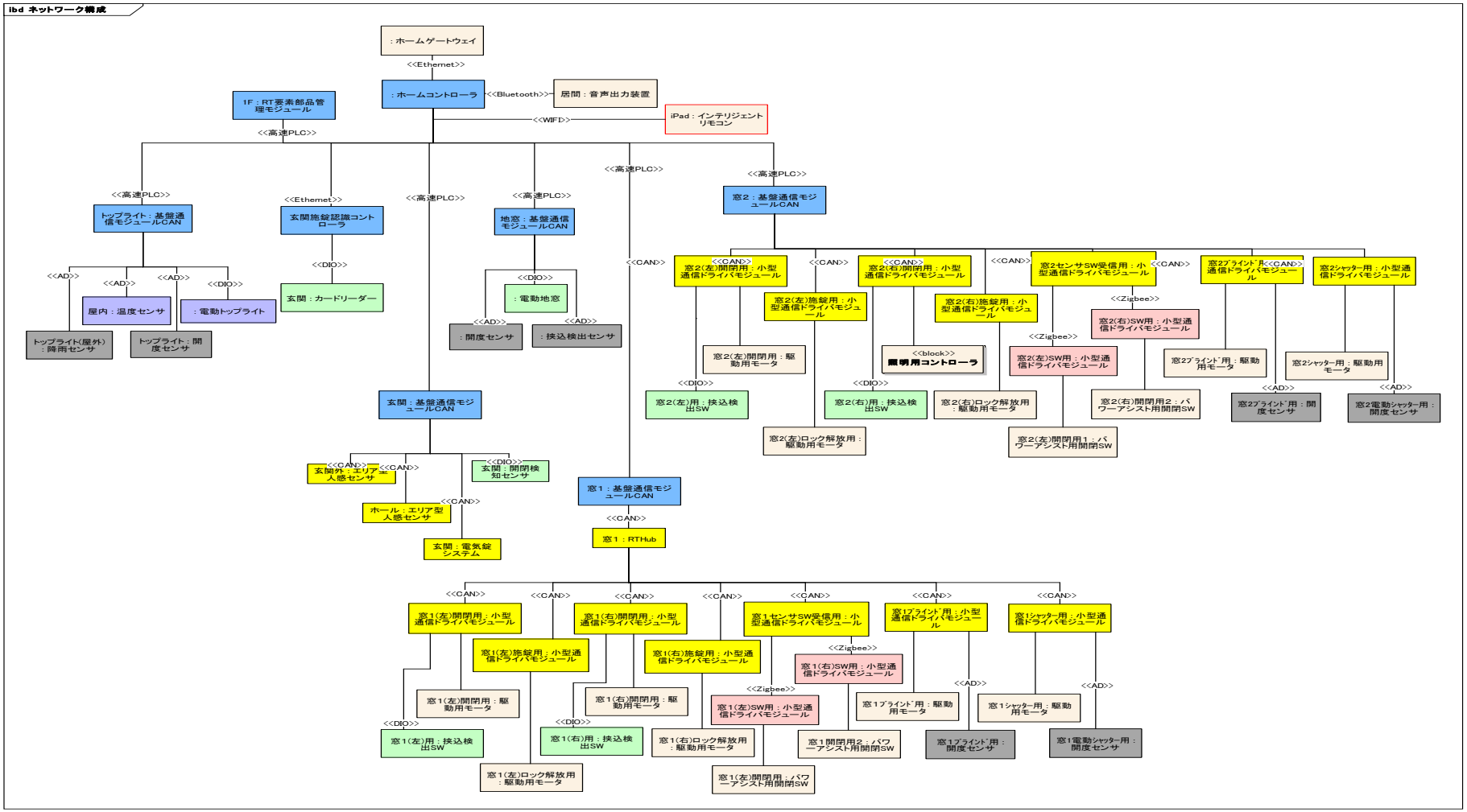
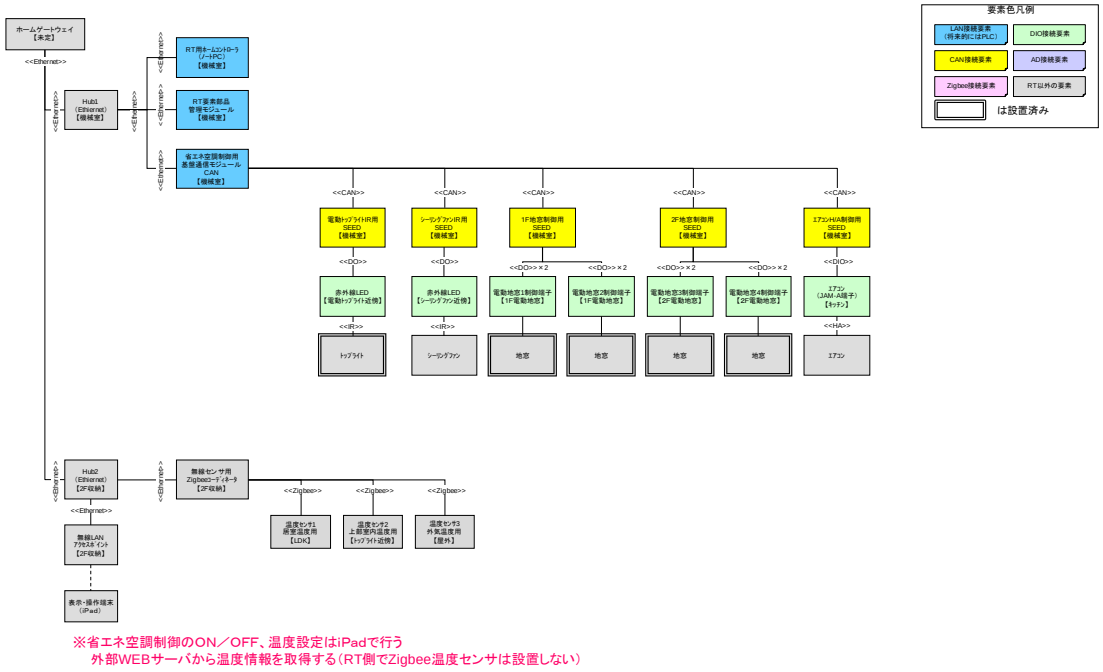


図 c-1-21　建物Aのシステム全体構成図

建物Bのインテリジェント空調システムの構成は図c－1-22 の通りである。撤去を前提とした設計の為、機械室内部で配線が完結するので PLC ではなく Ethernet による接続を行った。



※省エネ空調制御のON／OFF、温度設定はiPadで行う
外部WEBサーバから温度情報を取得する（RT側でZigbee温度センサは設置しない）

図c－1-22 建物Bのシステム構成図

c-1-6-4-d　結果と評価

住宅システムとして導入した RT システムの有効性を評価するために以下の評価項目を確認した。

① RT システムとしての動作確認

自動化する住宅設備が、設備ごとにクローズなシステムであることから家全体の制御を連携動作できないといった従来のシステムと異なり、本システムの優位性は RT ミドルウェアといった共通ネットワークプロトコルを使うことで、各設備がネットワーク化され、連携動作できるという点である。そこで、空調に係る連携動作（インテリジェント空調システム）と安心安全に係る連携動作（インテリジェント窓システム）の2つの連携動作を確認した。

インテリジェント空調システムについては、部屋に設置されている温度センサの温度に応じて、窓（地窓、トップライト）の開閉、シーリングファンの稼働およびエアコンの動作を連係して制御するシステムである。温度毎の連携動作は c-1-6-1-a 節の図 c-1-4 で示した通りに動作するロジックを組み込んだ。この動作を建物 A に組み込んだ各 RT 要素部品が連携して動作することを確認した。

デモ用センサ温度	地窓	トップライト	シーリングファン	エアコン
20℃未満	CLOSE	CLOSE	OFF	OFF
20℃以上25℃未満	OPEN	OPEN	OFF	OFF
25℃以上30℃未満	OPEN	OPEN	ON	OFF
30℃以上	CLOSE	CLOSE	OFF	ON

※設定温度は変更の可能性あり

図 c-1-4（再掲）　インテリジェント空調デモ用制御ロジック例（全体実証用建物：建物A）

インテリジェント窓システムにおいては、安心・安全に係る連携動作として、外出等における一括施錠の連携動作が挙げられる。今回は、玄関に電動化された施錠システムを組込み、それを RT コンポーネント化することで、各機器との連携を可能とした。すなわち、玄関の施錠を IC カードリーダーで行うと、室内における開けっ放しのドアが自動で閉められ施錠される。さらに消し忘れの照明が全て消灯し、エアコンや TV といった家電機器も自動的に電源を落とすといった連携動作を確認した。更に、本システムでは人感センサが設置されていることから、もし、居住者が中にいる場合は，その部分の家電製品や照明はそのまま稼働するといった制御も含まれている。以上の連携動作について確認することができた。

これらの連携動作は、ソフトウェアレベルで変更が可能であるために、住宅メーカー独自のアレンジや居住者のライフスタイルに応じて適宜修正することが可能となっている。

②RT 要素部品のプラグアンドプレイ機能確認

住宅システムでは、生活のライフスタイルに応じて、適宜住宅設備が変更、追加されることが一般的であるため、システムの可変を容易にする仕組みが必要となる。今回の RT ミドルウェアの仕組みとして、機器の RT コンポーネントがネットワーク上に立ち上がった段階で、システム上でつながるプラグアンドプレイの機能を開発している。この機能の確認を行った。今回、インテリジェント窓は窓以外にシャッター、ブラインドの 3 種類で構成されている。そこで、ブラインドを外した状態から、組付けを行い、他のシステムが稼働状態で RT コンポーネントを起動したとしても、自動的にシステムがブラインドを認識し、遠隔操作できる状態に移ることを確認した。加えて、今回ネットワークに PLC を利用していることから、一般の家電機器である扇風機に基盤通信モジュールを組込み、電源線を経由して扇風機を制御することを可能とした。これに対しても、扇風機が電源線に接続された段階でシステムが自動認識し、扇風機を遠隔操作できる状態にすることが可能であった。加えて、インテリジェント空調のシステムにも、新規で組み込まれた扇風機が連携動作することも確認された。

③PLCにおける基盤通信モジュール間のネットワーク動作

今回、住宅向けのシステムであることから、設置コストの少ない電力線通信：Power Line Communications（PLC）を、要素基板管理モジュールや基盤通信モジュール間のネットワークに利用している。通常は PC で利用されているようにネットワークでは Ethernet を利用するが、既存の住宅に組

み込むことや、設置コストを考えると、なるべく屋内配線は少ない方が望ましい。そのため、電力線を通して通信パケットを送信する PLC ネットワークを利用した。PLC には、HD-PLC 方式と HomePlugAV 方式の通信規格が代表的である。HD-PLC 方式はパナソニックが中心として国内で積極的に普及しようとしているが、米国のスマートグリットの流れから、米国における電力線搬送通信の業界団体 HomePlug Powerline Alliance（HomePlug）が積極的に HomePlugAV 方式を進めていることから国際的には、HomePlugAV 方式が標準に近いとされる。また、仕様がオープンになっているので、本事業では HomePlugAV 方式を採用した。HomePlug AV は最高 200 M ビット/秒の半二重であり、ほぼ理論上は Ethernet と差がない通信帯域を持っているが、他の家電機器、特にドライヤーや扇風機といったモーターが電力線と直接的に組み込まれているとの併用をすると、ノイズ源となり、実効値は住宅設備および屋内の配線経路によって大きく影響をうける。今回はその影響も踏まえ、実効値としてどのくらいの通信帯域を有するかを確認した。

建物 A である産業技術総合研究所内の実証住宅システムにおいて、ホームコントローラと基盤通信モジュール間、および要素基板管理モジュールと基盤通信モジュール間のスループットを評価した。双方共に、5M ビット/秒～4M ビット/秒の間のスループットであった。Fast EtherNet 規格（100BASE-TX）の場合、１００Mビット/秒であるが、実効値としては、20Mビット/秒程度である。よって、機器ノイズや配線経路によって、かなり通信速度は遅くなっているという結果となった。住宅の制御という面では、問題ない通信速度であったが、今後、ロボット等のシステムと環境側のセンサとのフィードバック系をつかってサービスするアプリケーションを検討する場合は、注意する必要がある。。また、ブレーカーを介すると通信自体が困難な状態になることも確認されており、利用する場合は、前もって通信環境を把握する必要がある。

④住宅システム内の RT 要素部品の安定動作

住宅システムは、通常の PC と異なり、24 時間、365 日稼働し続けることが要求される。それだけの安定動作を実現する機能を有することが必要である。LonWorks など、ビルのオートメーションに利用されているシステムは、トラフィックマネージャとされる通信状態を管理するソフトウェアが稼働し、常に通信トラフィックを 40%以下に抑えるように制御されている。本システムではトラフィックマネージャといったソフトウェアの監視機能を加える前段階として、ネットワークが高負荷で通信環境が悪い状態においても、以下に動作を安定化させるかについて検討した。すなわち、常に安定に動作する保証は現時点では通信トラフィックを下げておくといった定性的な処理によって行うだけであり、確実に長期に渡って動作を保証できるものではない。しかし、動作しなくなった状態を検知し、再接続することで、システム自らが自己復帰する機能を加えることで、安定性を向上させることは可能である。各 RT コンポーネントは、それぞれハートビートといった存在確認情報を定期的にホームコントローラに送っており、システムの健全性を常に監視している。この技術はプラグアンドプレイにも利用されており、動作が不安定になりハートビートが届かなくなると、その状態を報告する、もし不都合のある機器の RT コンポーネントが復帰し、ハートビートが再開されると、プラグアンドプレイの機能と同様に、システムが自己復帰する機能を有している。このため、既存のネットワーク負荷だけを制御している機能に、上記の自己復帰機能を併用することでより安定した動作を可能とする。本システムではその機能を組込、動作を確認した。

⑤通信モジュールの待機消費電力の評価

本システムは多くの組み込み系 MPU の小型コンピュータが通信モジュールの上で利用されている。それらが連携動作することで、エコを意識した空調システムが実現出来るが、各通信モジュールの待機電力と、削減が期待される消費電力量のバランスは意識する必要がある。そこで、本システムが稼働している状態で、ホームコントローラ、要素基板管理モジュール、基盤通信モジュール、小型通信ドライバモジュールの消費電力を計測した。表 c-1-6 に各機器の待機電力をまとめる。

表 c-1-6 各機器の待機電力

機器名	消費電力（W）
ホームコントローラ（EPSON EndeavorNP15）	16～18
要素基板管理モジュール	4
基盤通信モジュール	4
小型通信ドライバモジュール	0.6

それぞれの個数は住宅一棟に対して、ホームコントローラー台、階数毎に要素基板管理モジュール 1 台、設備毎に基盤通信モジュール 1 台、アクチュエータ毎に小型通信ドライバモジュール 1 台ということで見積もられる。今回の建物 A の住宅モデルにおいては、ホームコントローラ、要素基板管理モジュールが各一台、窓が 2 組あるため基盤通信モジュールは 2 台、小型通信ドライバモジュールについては、各窓に複数利用されているが、窓の構成として、図 c-1-15 における右側の窓については、窓開閉2台＋施錠2台＋トップライト1台＋ブラインド1台＋シャッター1台＋パワーアシスト用1台＝計8台利用されている。また、左側の窓については、窓開閉2台＋施錠2台＋ブラインド1台＋シャッター1台＋地窓2台＝計8台利用されている。そこで、合計 16 台と言うことで、3.6W となる。さらに、基盤通信モジュールは屋内に利用されている人感センサ毎に一台利用していることから 6 台使用している。以上から、本システムにおける通信機器の待機電力は、最大 57.6W として見積もられ、白熱電球60W 一個程度の電力消費ということで見積もられた。設備を増やすことで、待機消費電力は増加するであろうが、その全体消費電力と、見積もられる削減消費電力を比較し、バランスを考慮したシステム構成にすることが必要と言える。

c-1-6-4-e 住宅メーカーから見た今後の検討事項

(a) インテリジェント空調

建物 A と同様、建物Bでも複数の温度センサ情報に基づいた制御ロジックを搭載し、目論見通りの動作を実現した。建物BではRTシステムとは別のシステムが動作しており、それらとの連携を実現した事も既存建物への普及のテストケースとして効果的だった。具体的には、温度情報は本来RTシステムに接続されたセンサから取得するところを、別システムの情報をWEB経由で取得してシームレスに結合を図っている点、WEBインターフェースを介してシステムの発停、温度設定、個別機器のマニュアル操作を実現している点が挙げられる。図c－1-23 に操作画面例を示す。

図c－1-23 建物 B の涼風システム操作画面例

今回の実証において、実用性に関しての可能性を示すことができたが、今後、具体的なマーケットに合わせた改善点が必要とされる。表 c-1-7 にインテリジェント空調の検討課題をまとめる。

表 c-1-7 試作インテリジェント空調システムの改善課題例

	課題等	備考
1	動作の安定性改善	要素部品管理モジュールと基盤通信モジュール間の TCP/IP 通信（PLCではなくEthernet接続においても）異常により設備機器動作の停止が発生する事があり、自己復帰も起きないケースが依然存在する。再現性も低いため、ネットワーク負荷等の関係性も考慮し、引き続き詳しく調べていく必要がある。
2	ユーザインターフェースが不十分	温度設定や発停を操作する適当な端末がなく、タブレット型デバイスを用いてWEB画面操作で対応している。WEB画面デザインが単なるスイッチの羅列となっていて新味がなく改善が必要。

（b）インテリジェント・ウィンドウ（窓）システム

建物Aの 2 カ所の引き違い窓をインテリジェント・ウィンドウシステム化して評価を行った。その結果、操作感等の細かい調整の必要はあるものの、ほぼ目論見通りの機能を実現できることを確認した。表 c-1-8 に主な検討課題を示す。

表 c-1-8 試作インテリジェント・ウィンドウシステムの検討課題

	課題等	備考
1	サッシの開閉時のパワーアシスト操作感（ユーザインターフェース）	Zigbee通信を用いた無線スイッチを用いた事もありスイッチ操作に近い操作感で改良の余地あり。
2	停電時に窓の開閉が困難	サッシとアクチュエータをハードにリンクさせる構造の為、停電するとモータが負荷となり開閉が困難。手動でリンクを外す事で対応可能だが、構造を含めて改善の余地あり。
3	動作音が大きい	住宅で用いるにはモータの動作音が大きく改善を要する。
4	窓の閉め方	全閉時にサッシが戸当たりに衝突し大きな音が発生するケースが見受けられた。直線でスピードダウンをする等して衝撃音を無くす等の調整が必要。
5	施錠部の防犯性が低い	実証では戸先の突起部を窓枠に取り付けたアルミの長尺の板材を引っかけて固定する方式だが、強度が不足している。バールによるこじ開けで簡単に破壊してしまうと思われる。施錠方法の根本的見直しも含めて改善の余地あり。

（c）その他

建物AではPLC（高速）を用いてネットワークを構成し目的の機能を実現したが、一部の回路で通信品質が悪化する等の現象も見られた。また、PLCで従前から課題とされているノイズの影響も考えられ、インバータタイプの照明器具や一部のACアダプタ等の接続により動作が不安定になるケースも見受けられた。PLCについては継続的な改善が必要な他、通信が正常に行えない場合の原因の切り分けやそれらへの対策をケーズバイケースで具体的に実施できる様な運用上の準備を並行して考える必要があるだろう。こうしたノイズに強い中低速のPLCの利用、その場合のトラフィック削減の為のプロトコルの見直しも今後の改善として必要と考える。

使用時の課題として停電から電源が復旧した場合の動作について改善が必要である。要素部品管理モジュール、基盤通信モジュール、小型通信ドライバモジュールの電源投入の順序に制限は無い設計になっているが、不定期に認識が出来ないことがある。このため、より安定した停電からのシステム再稼働へのプロセスを構築する必要がある。

設備機器のRT化には開発支援ソフトを用いるが、簡易化したとはいえ汎用の開発ツールとなっている為に、設備機器メーカーの開発担当者が利用するにはまだ敷居が高いレベルと考えられる。例え

ば開口部部品（サッシ、シャッター、ブラインド、カーテン、etc.）に絞り込んで、各機器に対応した雛形を事前準備する事で、開発者はパラメータの幾つかを操作するだけで開発が終了する様なツールやキットがあると採用が進むと考えられる。

同様に設備機器連携で実現する家の機能開発に用いるツールも住宅メーカー等の担当者には訓練が必要である。現在は目的の機能が本当に実現できているのかを実際の建物に設置してみなければ確認できない状態だが、将来的にはCADや様々なシミュレーションツールとの連携を図って、机上である程度の作り込みが出来る環境構築が望ましい。

バグ対応やアップデートに関しても改善が必要である。ビジネスモデルで示した通り、様々なアプリケーションソフトウェアをネットワークを介して提供する事を想定している為である。ホームコントローラ上のアプリケーションの更新には対応すべきであり、更に下位のモジュールに搭載されたソフトウェアのアップデートの手法についても対応を検討する必要がある。加えて、これらが実現されると複数のアプリケーションソフトウェアが矛盾する動作を行わない仕組みも構築していく必要があると考えられる。

(c-2) プラグアンドプレイ機能を実現する統合ミドルウェアの開発(委託先:株式会社セック)

c-2-1 概要

プラグアンドプレイ機能を実現する統合ミドルウェアの開発では、OpenRTM-aist とアプリケーションの中間に位置し、RT システムの開発機関に、RT システムを容易かつ短期間で開発するための枠組みを提供することを目的としている。

統合ミドルウェアとして、基盤通信モジュール RT ミドルウェア上で動作するコンポーネントを管理し、プラグアンドプレイを実現する「RTC-Lite Manager」と、様々な RT ミドルウェアを RT システムとして連携させる「RTC HUB」を開発した。「RTC-Lite Manager」は、株式会社アルゴシステムが開発した基盤通信モジュール(CAN 版/ZigBee 版)上で、「RTC HUB」は、同じく株式会社アルゴシステムが開発した RT 要素部品管理モジュール上で実装・評価をおこなった。

本件での目標、並びに達成度は表 c-2-1 の通りである。

表 c-2-1 研究項目に対する目標ならびに達成度

研究項目	目標	達成度
プラグアンドプレイ機能を実現する統合ミドルウェアの開発	プラグアンドプレイ機能や RT 要素部品のステータス管理機能などを有する汎用 PC もしくは組み込み MPU を利用したホームコントローラ上で動作する基盤通信モジュールの統合ミドルウェアを実現する。また、基盤通信モジュール上の RTC-Lite フレームワークが、OpenRMT-aist と相互運用可能なように、SH4 / ARM9 200MHz 相当の RT 要素部品管理モジュール上に ProxyRTC 機能を実現する。統合ミドルウェアの動作環境として、Linux および Windows の 2 種類の OS をサポートする。	本目標は達成した。 プラグアンドプレイや RTC のステータス管理機能を持つ統合ミドルウェアを開発した。
	統合ミドルウェアおよび ProxyRTC の実証評価版は、平成 21 年度中に RT システム開発機関に無償で提供し、RT システムの開発に供する。この際、統合ミドルウェアと ProxyRTC だけではなく、仕様書および取扱説明書もあわせて提供する。	本目標は達成した。 統合ミドルウェアのソフトウェア、ドキュメントをプロジェクト実施者に提供し評価した。
	実証評価の結果をフィードバックし、統合ミドルウェアと ProxyRTC の最終版をプロジェクト終了時までにリリースする。	本目標は達成した。 フィードバックした最終版をリリースした。

研究項目	目標	達成度
	22 年度に構築する実証 RT システムについては、株式会社ミサワホーム総合研究所で有する顧客からの要求仕様を元に、株式会社セックが必要な要素部品を構成するシステムの仕様書を作成し、事業主体として各委託機関からの RT 要素部品や関連ソフトウェアをとりまとめて構築する。	本目標は達成した。 産総研およびミサワホーム展示場に実証フィールドを構築し、実証 RT システムを評価した。
	住宅価格に対して RT システムの価格が 5%以下、例えば 2400 万円（床面積 40 坪、坪単価 60 万円）の住宅の場合、RT システムの価格を 120 万円以下とする事を目標とする。	本目標は実現可能と判断する。 ただし、導入する RT 要素部品や RT システムによる。

c-2-2　RT 要素部品群による RT システム

本システムでは、住宅に設置される窓や家電、各種センサなどを、それぞれ一つの RT 要素部品とし、それらを RT ミドルウェア技術により連携させることで、住宅向けの環境分散型 RT システムを実現している。本システムでは、ホームコントローラによる集中管理ではなく、モジュールごとの分散管理のアーキテクチャを採用している。それぞれの RT 要素部品は、基盤通信モジュール、要素部品管理モジュール、ホームコントローラにより、階層的に管理がおこなわれる。各モジュールは独立して動作が可能であり、システムの一部を変更した場合やモジュールに異常があった場合でも、他のモジュールに影響を及ぼすことなく動作することが可能である。

c-2-2-1　ハードウェア構成

本システムのハードウェア構成を図 c-2-1 に示す。本システムは、ホームコントローラ、RT 要素部品管理モジュール、基盤通信モジュール、小型通信ドライバモジュール、及び各種デバイスによって構成される。

ホームコントローラは1軒の住宅内に 1 台程度配置するもので、タッチパネルを有した高性能な組み込み型 MPU である。住宅内の全ての設備を統合しており、ユーザとのインタフェースやインターネット上のデータセンターとの通信などをおこなう。住宅内に配置される全ての設備機器の全てをホームコントローラで管理することが難しいため、複数の設備機器を統合する機器として、RT 要素備品管理モジュールを配置する。RT 要素部品管理モジュールは部屋毎もしくはブレーカー毎に配置される。基盤通信モジュールは端末のセンサやアクチュエータを制御するものであり、窓や屋外センサなどの設備機器ごとに配置される。小型通信ドライバモジュールもデバイスを制御するものであり、住宅内に大量に配置される。

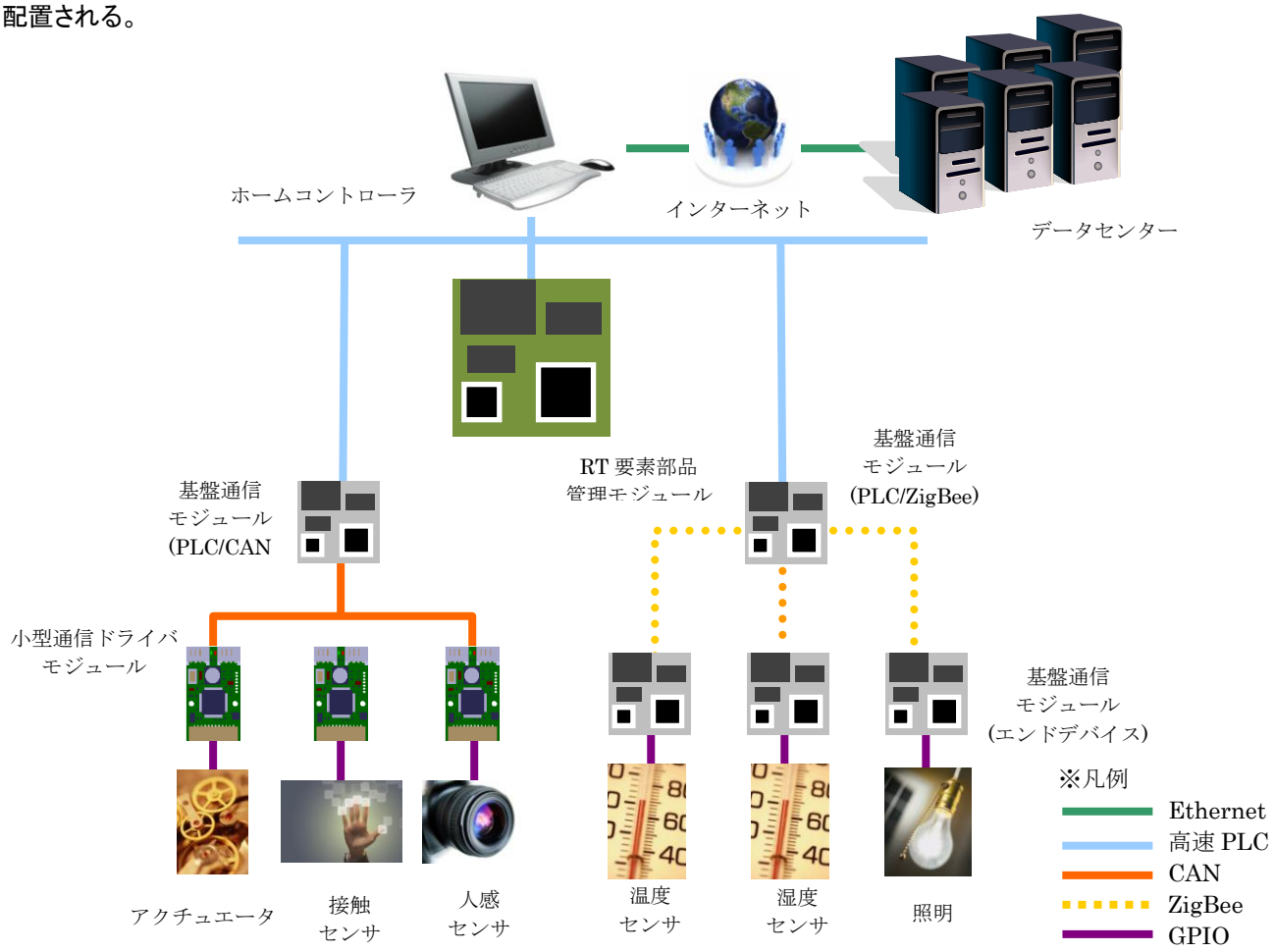


図 c-2-1　ハードウェア構成

図 c-2-1 に示したハードウェアの詳細を表 c-2-2 に示す。

表 c-2-2 ハードウェア詳細

名称	仕様	備考
ホームコントローラ	汎用 PC 並み	OS :Windows RTM :OpenRTM.NET
RT 要素部品管理モジュール	CPU :AM1808(375MHz) ROM :2GB RAM :256MB	OS :Linux RTM :OpenRTM-aist
基盤通信モジュール (PLC/CAN 版)	CPU :SH7214(100MHz) ROM :1MB RAM :128KB I/F :CAN	OS :TOPPERS / ASP RTM :RTC-Lite Manager + miniRTCs
基盤通信モジュール (PLC/ZigBee 版)	CPU :SH7214(100MHz) ROM :1MB RAM :128KB I/F :ZigBee	OS :TOPPERS / ASP RTM :RTC-Lite Manager + microRTCs
小型通信ドライバモジュール	CPU :Cortex(72MHz) Flash :256KB RAM :64KB	OS :なし RTM :miniRTCs
基通信通信モジュール (エンドデバイス)	CPU :CC2530(16MHz) ROM :256KB RAM :8KB	OS :なし RTM :microRTCs
センサ/アクチュエータ		

図 c-2-1 に示した通信プロトコルの通信帯域について表 c-2-3 に示す。

表 c-2-3 通信帯域

名称	通信帯域	備考
Ethernet	数 Mbps 以上	
高速 PLC	数 Mbps 以上	
CAN	1Mbps	
ZigBee	250Kbps	

c-2-2-2　ソフトウェア構成

本システムのソフトウェア構成を図　c-2-2　に示す。本システムはモジュールの構成に応じて、RTC-Lite をベースとした高速制御用 RTC-Lite、低速制御用 RTC-Lite、及び OpenRTM-aist をベースとした統合ミドルウェアの 2 種類の RT ミドルウェアから構成される。

高速制御用 RTC-Lite、及び低速制御用 RTC-Lite は、省資源なマイコンでも動作するよう機能を制限し、軽量な実装となっている。この RT ミドルウェアは、基盤通信モジュール、及び小型通信モジュール上で動作し、通信プロトコルとして CAN と ZibBee を対象としている。

統合ミドルウェアは、基盤通信モジュールの統合機能や、アプリケーションを構築するための機能を有する。この統合ミドルウェアは、RT 要素部品管理モジュール上で動作する。

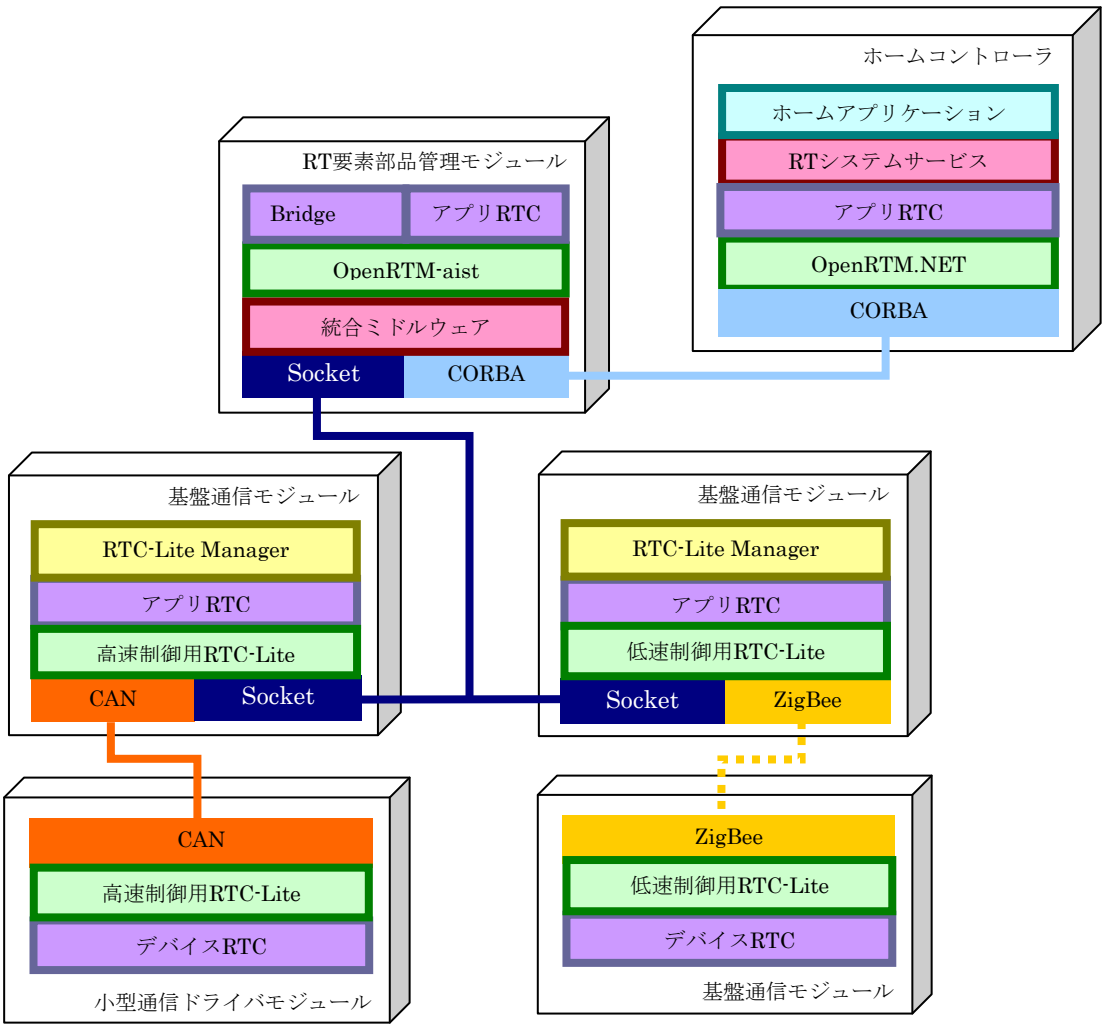


図 c-2-2　ソフトウェア構成

図 c-2-2 に示したソフトウェアの詳細について表 c-2-4、及び表 c-2-5 に示す。

表 c-2-4 ソフトウェア詳細(1/2)

ハードウェア	ソフトウェア	処理内容	備考
ホームコントローラ	ホームアプリケーション	タッチパネルなどを使用し、システム全体のコントロールをおこなう。 住宅のモードの変更や各部屋の状況の表示をおこなう。	
	RT システムサービス	RT システムのデプロイメントサービス・ネーミングサービス・ヘルスモニタリングサービス・ロギングサービス・ビルディングサービスを提供する。	
	アプリ RTC	ホームネットワークシステム全体を統括する RTC。	
	OpenRTM.NET	Microsoft .NET Framework 上で動作する RT ミドルウェア実装。RT ミドルウェアとしての全機能を持つ。	開発対象外
RT 要素部品管理モジュール	Bridge RTC	基盤通信モジュール上で動作する複合コンポーネントの代理コンポーネント。	
	アプリ RTC	「ブレーカー」や「部屋」毎に配置される RTC。	
	OpenRTM-aist	産業技術総合研究所が開発する RT ミドルウェア実装。RT ミドルウェアとしての全機能を持つ。	開発対象外
	統合ミドルウェア (RTC HUB)	プラグアンドプレイ機能を実現し、起動した基盤通信モジュール上のコンポーネントの Bridge RTC を自動生成する。 また、RT システム支援ツールと連携し、基盤通信モジュール配下のコンポーネントの情報取得や、コンポーネントに対するコマンド送信を RTC-Lite Manager との間でおこなう。 統合ミドルウェアと RT システム支援ツール、及び RTC-Lite Manager との通信は、OpenRTM とは別の独自のプロトコルを用いる。	

表 c-2-5 ソフトウェア詳細(2/2)

ハードウェア	ソフトウェア	処理内容	備考
基盤通信モジュール	アプリ RTC	「窓」や「屋外センサ」などを制御する RTC。	
	RTC-Lite Manager	プラグアンドプレイ機能を実現し、起動したデバイス RTC を自動認識し、RTS プロファイルを基に、デバイス RTC の活性化や接続をおこなう。 また、基盤通信モジュール配下のコンポーネントを複合化し、統合ミドルウェアから一つのコンポーネントとして扱うための仕組みを持つ。 通信に関して、PLC と CAN/ZigBee のプロトコル変換をおこなう Gateway 機能を有する。	
小型通信ドライバモジュール	高速制御用 RTC-Lite (miniRTCs)	OS は搭載せず、メインルーチンと割り込みルーチンにより動作する RT ミドルウェア。デバイス RTC 同士の通信を可能とする。 通信プロトコルとして CAN を扱う。	
	デバイス RTC	デバイスを制御する RTC。miniRTCs 環境で動作するコンポーネントであり、RT ミドルウェアとしての機能は制限される。	
基盤通信モジュール	低速制御用 RTC-Lite (microRTCs)	M3T-MR30/4 上で動作する RT ミドルウェア。デバイス RTC 同士の通信はできず、RTC-Lite Manager を経由する必要がある。 通信プロトコルとして ZigBee を扱う。	
	デバイス RTC	デバイスを制御する RTC。microRTCs 環境で動作するコンポーネントであり、RT ミドルウェアとしての機能は制限される。	

c-2-2-3　データフロー

データフローを図 c-2-3 に示す。

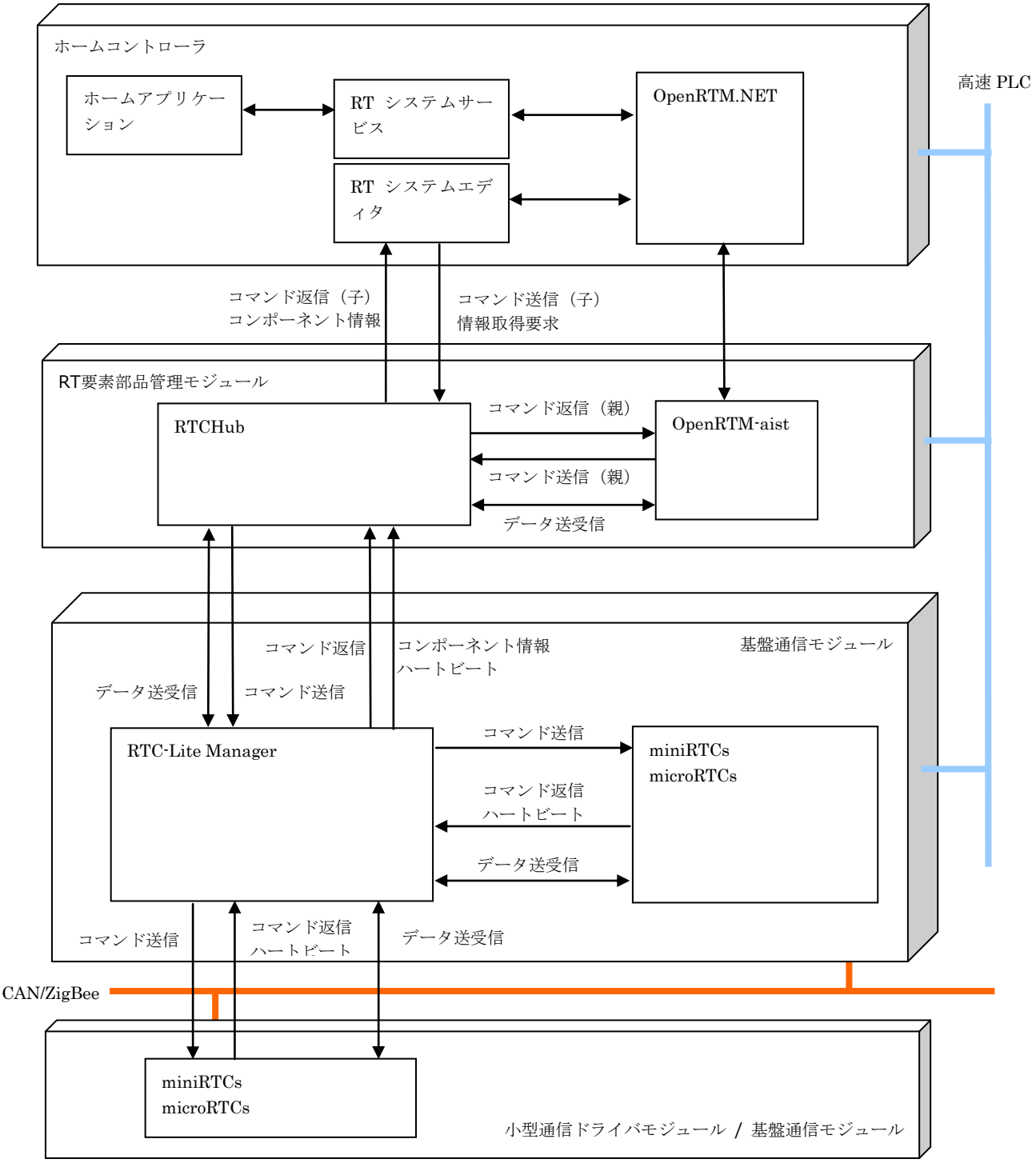


図 c-2-3　データフロー

c-2-2-4 コンポーネント構成

本システムにおける RTC 構成例を図 c-2-4 に示す。

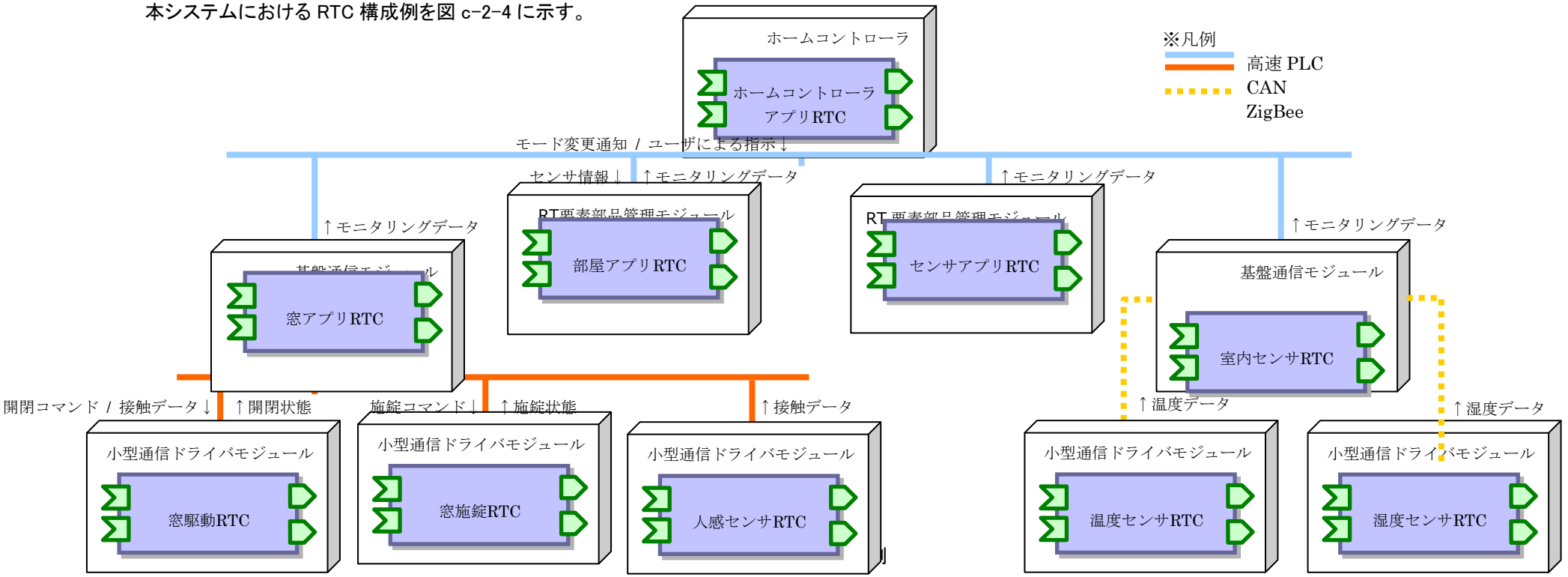


図 c-2-4 に示した RTC 構成について、コンポーネントを扱う際は基盤通信モジュール配下のコンポーネントを複合化し、基盤通信モジュール上の RTC を Bridge RTC として要素部品管理モジュール上へ配置するようにする。このように扱う RTC 構成を図 c-2-5 に示す。

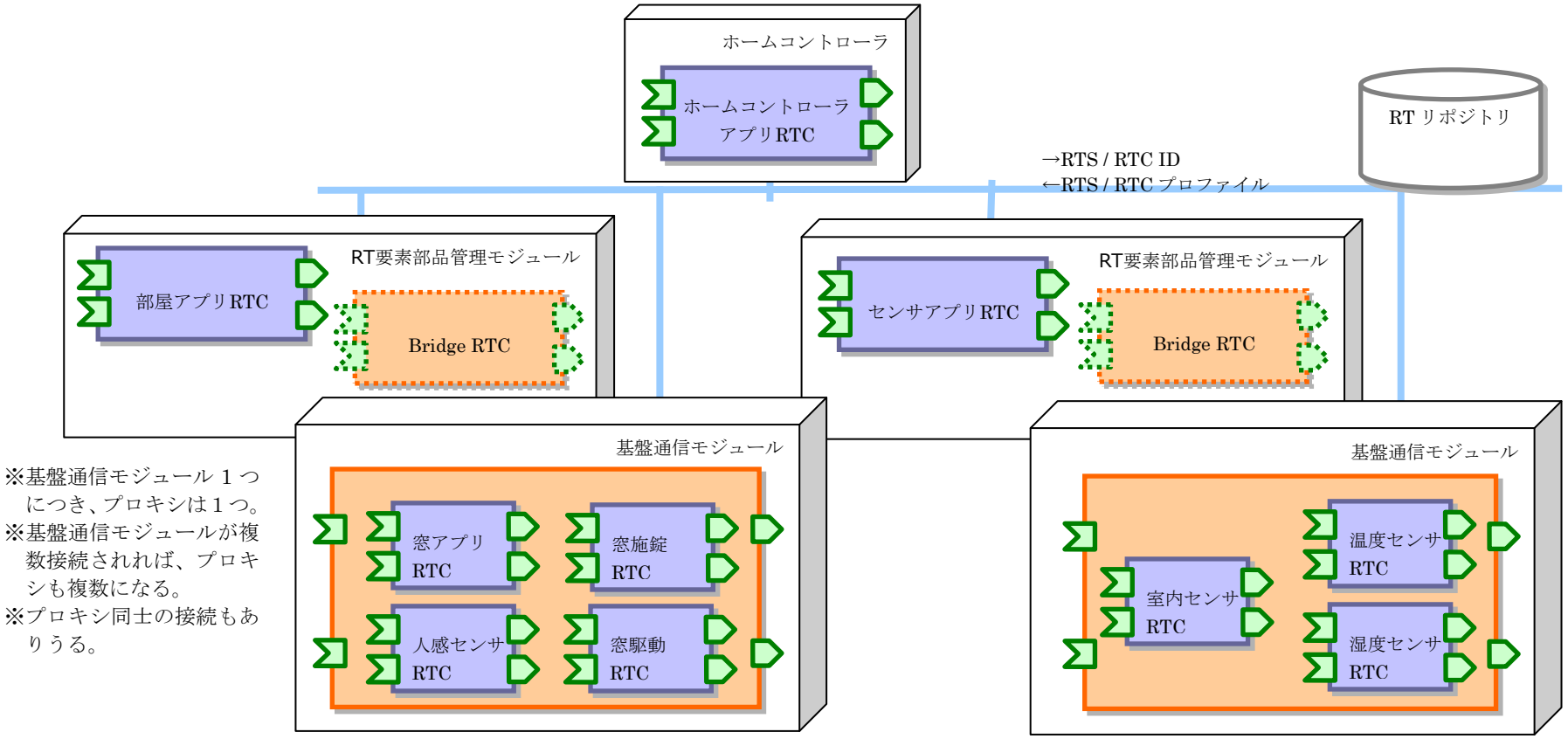


図 c-2-5 複合コンポーネント及び Bridge RTC 化した RTC 構成

複合コンポーネント

基盤通信モジュール上にて、以下の状況を勘案し、基盤通信モジュール以下の複数のコンポーネントをあたかも 1 つのコンポーネントであるかのように見せることとした。

・下位の系は下位の系として閉じた状態にしたい。
・末端のデバイスコンポーネントをプロキシコンポーネントとして見せるためには、プロキシのプロキシを生成する必要があり、処理が複雑になる。

しかし、RTC の開発中においては、複合コンポーネント内の子コンポーネントを操作する必要がある。そのため、子コンポーネントの状態や接続状況を RTSystemEditor にて表示できる必要がある

複合コンポーネントの仕様について以下に示す。

(1) コンポーネントの起動

RTS プロファイルにて、<isRequire/>と設定されているコンポーネントが全て起動した時点で、「起動」として扱う。

(2) コンポーネントの状態遷移コマンド受信

複合コンポーネントに対して状態遷移コマンドが送信された際は、全ての子コンポーネントに対してコマンドを送信する。

(3) 異常ケース

① コンポーネントの状態が不一致

Active／Inactive が混在している際は、Active として RTSystemEditor へ通知する。Error のコンポーネントが 1 つでもあれば、Error として扱う。

② コンポーネントがエラー状態に遷移

子コンポーネントが Error 状態に遷移した場合、RESET コマンドの自動発行はしない。

c-2-3 RTC-Lite Manager

c-2-3-1 RTC-Lite Manager 構成

RTC-Lite Manager には通信プロトコルに応じて miniRTCs、又は microRTCs を含み、RTC-Lite Manager 上でも RTC を動作させることが可能である。また、RTC HUB 上のプロキシコンポーネントを軽減するため、デバイス基盤通信モジュール配下の RTC を複合化させる。

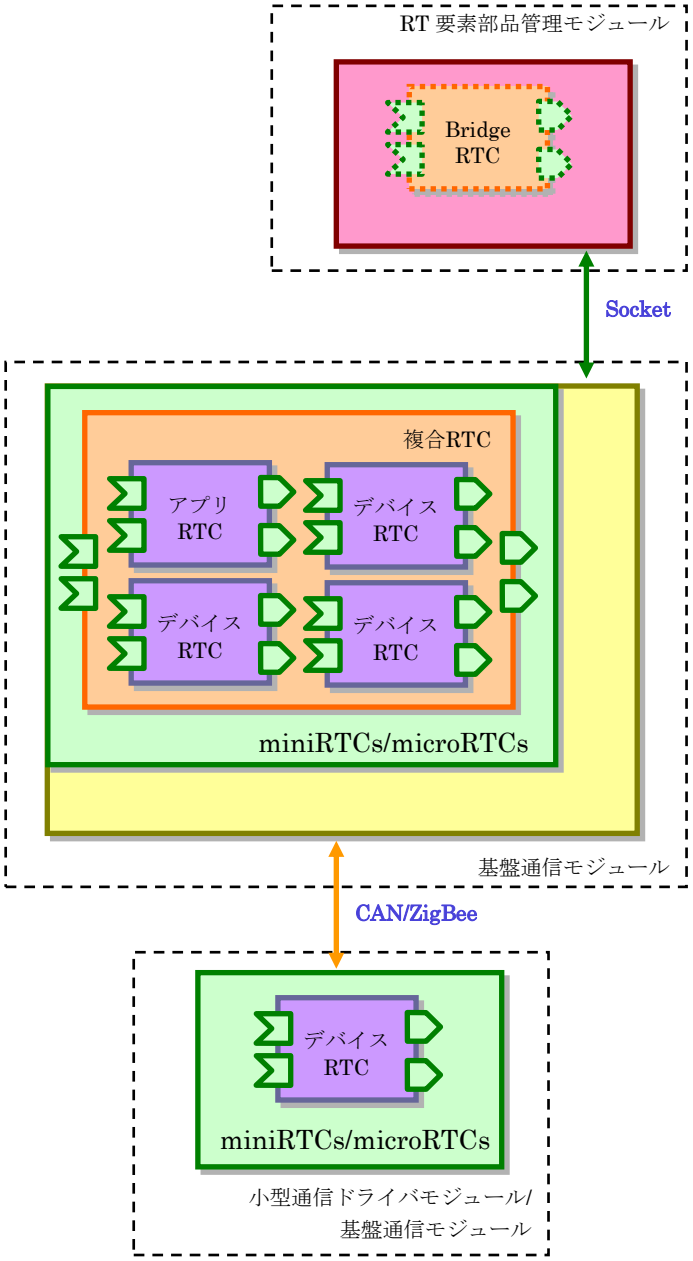


図 c-2-6 RTCLite Manager 構成

Bridge RTC

基盤通信モジュール上で動作する複合コンポーネントの代理コンポーネント。

複合 RTC

RTC-Lite Manager により、miniRTCs、及び microRTCs 上で動作する RT コンポーネントを複合化したもの。

アプリ RTC

RTC-Lite Manager 内の miniRTCs、又は microRTCs 上で動作するコンポーネント。

デバイス RTC

miniRTCs/microRTCs 環境で動作するコンポーネントであり、RT ミドルウェアとしての機能は制限される。デバイスと 1 対 1 で対応し、センサやモータの制御をおこなう。

c-2-3-2 機能

RTC-Lite Manager の機能一覧を表 c-2-8 に示す。

表 c-2-8 機能一覧

機能	機能対応	備考
プラグアンドプレイ	対応する	
生存情報の定期送信	対応する	1 秒周期で送信する
複合 RTC の状態遷移	表 c-2-7 参照	
複合 RTC ノード数	CAN 版　:最大 15 個 ZigBee 版 : 最大 64 個	複合化した仮想 RTC のノード ID は 0 固定
公開データポート	最大 128 ポート	実体は RTC のデータポート
接続可能ポート数	最大 128 個	複合化した RTC 同士の接続、及び公開ポートによる外部の RTC との接続を含む
サービスポート	持たない	データポートで代用する
通信タイプ （対 RT 要素部品管理モジュール）	Ethernet	
通信タイプ （対基盤通信モジュール）	CAN/ZigBee	RTC-Lite Manager と同じ基盤通信モジュール上に載っている RTC に対してはメソッド呼び出しにより通信をおこなう
miniRTCs 機能	対応する	通信タイプとして CAN を持つ場合のみ
microRTCs 機能	対応する	通信タイプとして ZigBee を持つ場合のみ

RTC-Lite Manager の RTC 状態複合化対応表を表 c-2-7 に、個々の RTC の取りうる状態一覧を表 c-2-8 に示す。

表 c-2-7　RTC 状態複合化対応表

RTC 状態 / RTC 状態	Inactive	Active	Error / Timeout
Inactive	Inactive	Active	Error
Active		Active	Error
Error / Timeout			Error

表 c-2-8　RTC の状態一覧

ID	RTC の状態
0x00	未定義状態
0x01	Created 状態
0x02	Inactive 状態
0x03	Active 状態
0x04	Error 状態
0x10	ハートビート受信タイムアウト

c-2-3-3 インタフェース仕様

c-2-3-3-1 TCP/IP 通信プロトコル

本システムでは、独自の TCP/IP 通信プロトコルを利用して、RTC HUB と RTC-Lite Manager 間のコマンド及びデータの送受信をおこなう。その詳細について以下に説明する。

c-2-3-3-2 TCP/IP 通信共通メッセージ定義

TCP/IP 通信で送受信するコマンドやデータなどのメッセージに共通のメッセージ構造を表 c-2-9 に示す。

表 c-2-9 TCP/IP 通信共通メッセージの構造

項目名	内容	サイズ
データ長	データ長	4byte
メッセージ ID	メッセージ種別を示す ID	1byte
送信元ノード ID	メッセージの送信元ノード ID	1byte
送信元オブジェクト ID	メッセージの送信元オブジェクト ID	1byte
送信先ノード ID	メッセージの送信先ノード ID	1byte
送信先オブジェクト ID	メッセージの送信先オブジェクト ID	1byte
データ	メッセージ種別毎のデータ内容	n byte

メッセージ ID を表 c-2-10 に示す。

表 c-2-10 メッセージ ID 一覧

ID	名称	内容	備考
0x00	特権通信	RTC-Lite Manager のシステム外のメッセージ	今後の拡張
0x01	コマンド送信	コンポーネントに対して実行させるコマンドのメッセージ	
0x02	コマンド返信	コンポーネントが実行したコマンドの応答メッセージ	
0x03	ショートデータ送信	8byte 以下の送信データメッセージ	
0x04	ロングデータ送信	分割された送信データメッセージ	今後の拡張
0x05	ハートビート	RTC から周期的に送信される生存情報	
0x10	RTS-ID	RTS-Profile の ID	今後の拡張
0x20	RTC 情報取得	RTC-Lite Manager が管理している RTC の情報を取得するメッセージ	

c-2-3-3-3 メッセージ種別
c-2-3-3 インタフェース仕様

(1) コマンド送信メッセージ
コマンド送信メッセージにて送信するデータを表 c-2-11 に示す。

表 c-2-11 コマンド送信メッセージの構造

コマンド種別	データ部(各 1byte)		
Activate	コマンド ID:0x00	なし	なし
Deactivate	コマンド ID:0x01	なし	なし
Reset	コマンド ID:0x02	なし	なし
Exit	コマンド ID:0x03	なし	なし
Connect	コマンド ID:0x20	接続先ノード ID	接続先ポート ID
Disconnect	コマンド ID:0x21	接続先ノード ID	接続先ポート ID

(2) コマンド返信メッセージ
コマンド返信メッセージにて送信するデータを表 c-2-12 に示す。

表 c-2-12 コマンド返信メッセージの構造

コマンド種別	データ部(各 1byte)		
Activate	コマンド ID:0x00	戻り値(エラーコード)	なし
Deactivate	コマンド ID:0x01	戻り値(エラーコード)	なし
Reset	コマンド ID:0x02	戻り値(エラーコード)	なし
Exit	コマンド ID:0x03	戻り値(エラーコード)	なし
Connect	コマンド ID:0x20	戻り値(エラーコード)	なし
Disconnect	コマンド ID:0x21	戻り値(エラーコード)	なし

(3) ハートビートメッセージ
ハートビートメッセージにて送信するデータを表 c-2-13 に示す。

表 c-2-13 ハートビートメッセージの構造

項目名	内容	サイズ
カウンタ	複合 RTC の起動時からハートビートを送信する毎にカウントアップする値	4byte
状態	複合 RTC の状態	1byte

(4) データメッセージ

データメッセージにて送信するデータを表 c-2-14 に示す。

表 c-2-14 データメッセージの構造

項目名	内容	サイズ
データ	マーシャリングされたデータ	n byte

(5) RTS ID メッセージ

RTS ID メッセージにて送信するデータを表 c-2-15 に示す。

表 c-2-15 RTS ID メッセージの構造

項目名	内容	サイズ
RTS ID	RTS-Profile の ID	nbyte

(6) RTC 情報取得メッセージ

RTC 情報取得メッセージにて送信するデータを表 c-2-16、RTC 情報取得メッセージに対して返信するデータを表 c-2-17、データ種別 ID と返信するデータの対応表を表 c-2-18 に示す。

表 c-2-16 RTC 情報取得メッセージの構造(統合ミドル→RTC-LiteManager)

項目名	内容	サイズ
データ種別 ID	データ種別を表す ID	1byte

表 c-2-17 RTC 情報取得メッセージの構造(RTC-LiteManager→RTC HUB)

項目名	内容	サイズ
データ種別 ID	データ種別を表す ID	1byte
データ	マーシャリングされたデータ	nbyte

表 c-2-18 データ種別 IDーデータ部内容対応表

データ種別 ID	データ内容	サイズ
0x00	RTC の生存(1:生存している、0:生存していない) 1 : Inactive/Active/Error 0 : Undefined/Created/Timeout	1byte
0x01	エラーコード	1byte
	RTC の状態	1byte
0x02	エラーコード	1byte
	getRate	nbyte
0x03	エラーコード	1byte
	setRate	nbyte
0x04	エラーコード	1byte
	getKind	nbyte

c-2-3-3-4 UDP 通信プロトコル

RTC HUB と RTC-Lite Manager 間の TCP 通信を確立するため、UDP 通信にて接続に必要なデータを本システム起動時にブロードキャスト送信する。UDP 通信のメッセージ構造を表 c-2-19 に示す。

表 c-2-19 UDP 通信メッセージの構造

項目名	内容	サイズ
データ長	データ長	4byte
RTS ID	RTS-Profile の ID	1byte
IP アドレス	送信元ノードの IP アドレス	4byte
ポート番号	送信元ポート番号	2byte
グループ ID	複合 RTC のグループ ID	1byte
複合コンポーネント名	複合コンポーネントの名前	nbyte

c-2-3-3-5 CAN 通信プロトコル

a-2-3-3 インタフェース仕様参照。

c-2-3-3-6 ZigBee 通信プロトコル

a-2-4-3 インタフェース仕様参照。

c-2-3-4 制約事項

・RTS-Profile に定義する複合 RTC のノード ID は 0 固定とする

・コマンド送信にて、Connect/Disconnect を送信する際、送信先は接続情報の IN ポートとする
（Connect コマンドなら、送信先が IN ポートで、接続先が OUT ポートとなる）

c-2-4 統合ミドルウェア

本システムでは、複数の窓や家電、センサなどを設置するため、数 10～100 個の RT 要素部品を用意する必要がある。従来の RTC-Lite 方式を採用した場合、要素部品管理モジュール上で、デバイス RTC と同数のプロキシ RTC が実行されることになり、システム全体のパフォーマンスのボトルネックとなっていた。そこで我々は、miniRTCs と microRTCs は、OpenRTM-aist を必要せず、それぞれ独立して動作可能な組込みモジュール向けの RT ミドルウェアとして実装した。ただし、RT ミドルウェア同士を接続するためのブリッジ RT コンポーネント(ブリッジ RTC)を用意することで、RT ミドルウェア間の相互接続を実現している。このように、RT ミドルウェア同士を相互接続する方式を、RTC HUB 方式と呼称する(図 c-2-7、図 c-2-8)。

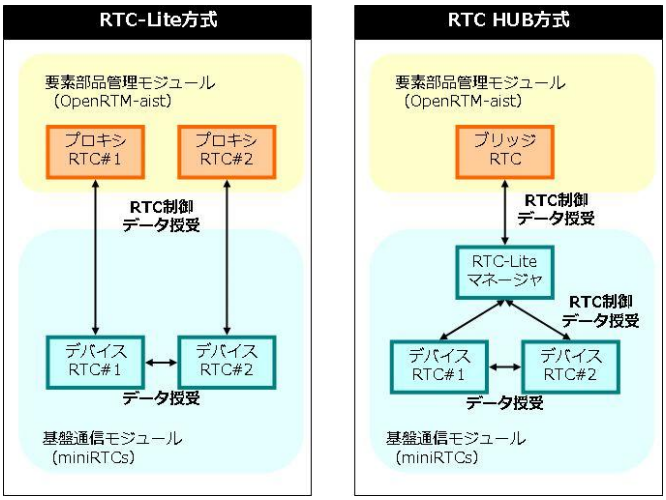


図 c-2-7 RTC-Lite 方式と RTC HUB 方式

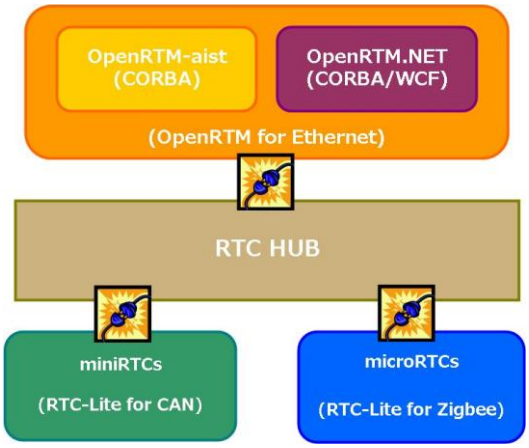


図 c-2-8 RTC HUB による RT ミドルウェアの連携

c-2-4-1　RTC HUB 構成

RTC HUB は、OpenRTM-aist の機能を使用し、RTC-Lite Manager 上のコンポーネントの代理コンポーネントとなる Bridge RTC を生成する。Bridge RTC は OpenRTM-aist 上で動作し、RT ミドルウェアとしての全機能を持つ。

RTSystemEditor から、基盤通信モジュールが複合化している各コンポーネントへのアクセスは、独自の CORBA インタフェースによりおこなう。

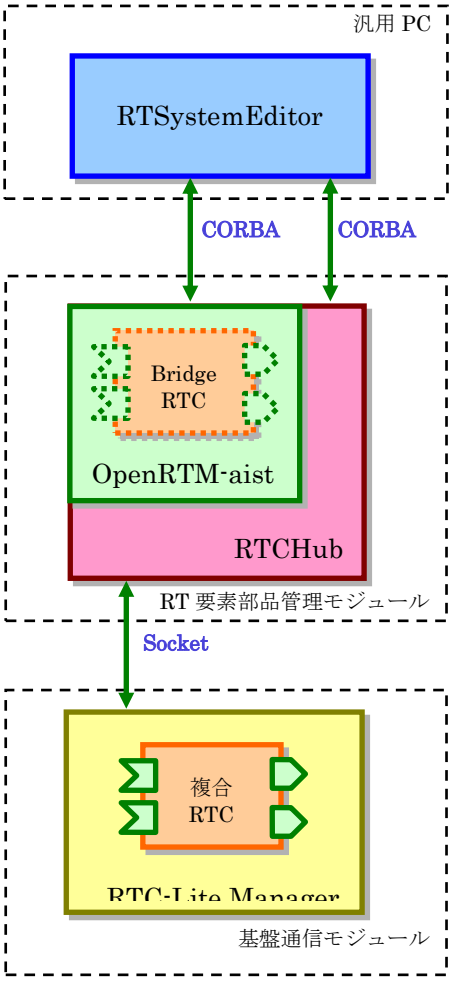


図 c-2-9RTC HUB 構成

Bridge RTC

基盤通信モジュール上で動作する複合コンポーネントの代理コンポーネント。

複合 RTC

RTC-Lite Manager により、miniRTCs、及び microRTCs 上で動作する RT コンポーネントを複合化したもの。

c-2-4-2 機能

RTC HUB の機能一覧を表 c-2-20 に示す。

表 c-2-20 機能一覧

機能	機能対応	備考
プラグアンドプレイ	対応する	RTC-Lite Manager から UDP による起動通知を受信し、TCP 接続を確立する。
プロキシの自動生成	対応する	未登録のコンポーネントから起動通知を受信すると、RTS プロファイル及び RTC プロファイルに基づき Bridge RTC を生成する。
プロキシ機能	Bridge RTC にて以下を対応する ・状態遷移 ・ポート接続 ・データポート送受信	
生存情報監視	RTC-Lite Manager から一定時間生存情報の通知がなければ Bridge RTC をエラー状態に遷移させる	タイムアウト時間：5 秒
RT システム支援ツール連携	独自の CORBA インタフェースにて対応する	
Socket 通信	RTC-Lite Manager と独自の Socket インタフェースにて通信する	通信タイムアウト設定 ・TCP メッセージ送信 ：2 秒 ・TCP メッセージ受信 ：2 秒 ・UDP メッセージ受信 ：5 秒

c-2-4-3 インタフェース仕様

c-2-4-3-1 TCP/IP 通信プロトコル

a-2-3-3 インタフェース仕様を参照のこと。

c-2-4-3-2 UDP 通信プロファイル

a-2-4-3 インタフェース仕様を参照のこと。

c-2-4-3-3 プロファイル

(1) プロファイルの参照

各種プロファイル情報の参照関係を図 c-2-10 に示す。

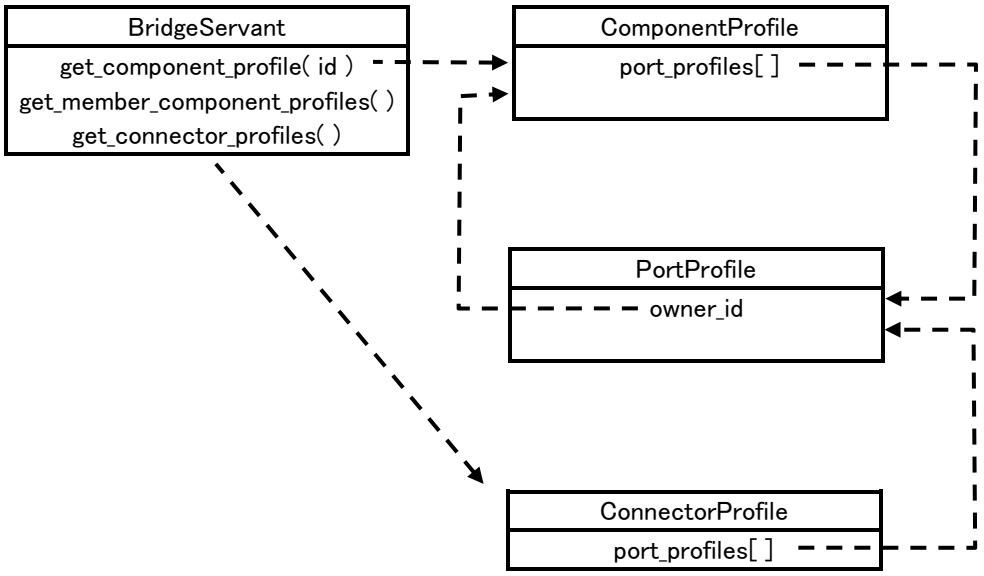


図 c-2-10 各種プロファイル情報の参照関係

(2) RTS プロファイル

RTC HUB 特有の RTS プロファイルの設定項目を以下に示す。

ファイル名

複合コンポーネント名.xml

component

component		
attributes		
pathUri	1	対応する RTC プロファイルのパス

dataPort

dataPort		
attributes		
name	1	インスタンス名.ポート名

(3) RTC プロファイル

RTC HUB 特有の RTC プロファイルの設定項目を以下に示す。

basic_info_ext

Properties		
attributes		
value	1	デバイス毎に指定するノード ID miniRTCs：1～15 microRTCs：1～255 ※任意の ID を指定可能。
name	1	node_id

Properties		
attributes		
value	1	CAN のバス ID や ZigBee の PAN ID miniRTCs：1～255 microRTCs：1～255 ※任意の ID を指定可能。
name	1	bus_id

dataport_ext

Properties		
attributes		
value	1	コンポーネントのポート毎に付与するオブジェクト ID miniRTCs：1～15 microRTCs：1～15 ※ID は連番のみしか指定できない。
name	1	object_id

c-2-4-3-4　CORBA インタフェース

(1) RTSystemEditor インタフェース

RTSystemEditor が複合コンポーネントの内部を表示、及び操作するためには、子コンポーネントに対してのインタフェースが必要である。RTSystemEditor と RTC HUB とのインタフェースは独自 IDL を利用することとし、図 c-2-11 に示す構成とする。

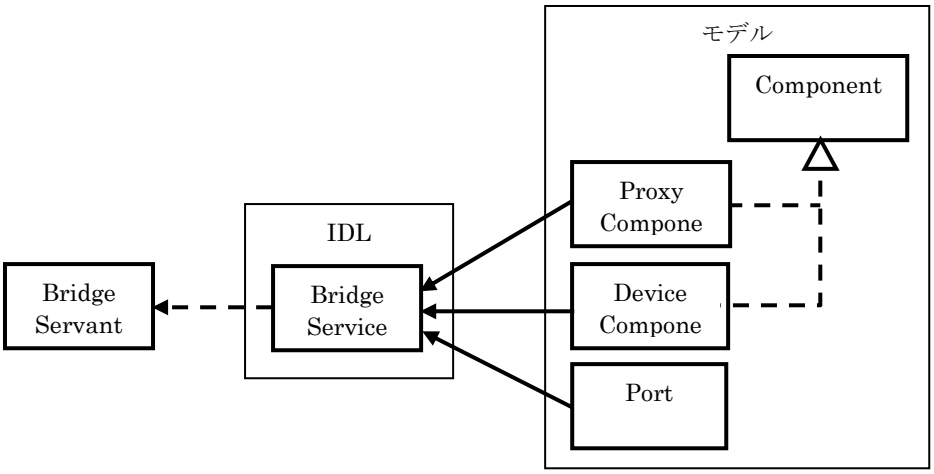


図 c-2-11 RTSystemEditor インタフェース

(2) IDL 構成

IDL で定義するクラスは図 c-2-12 に示す構成とする。

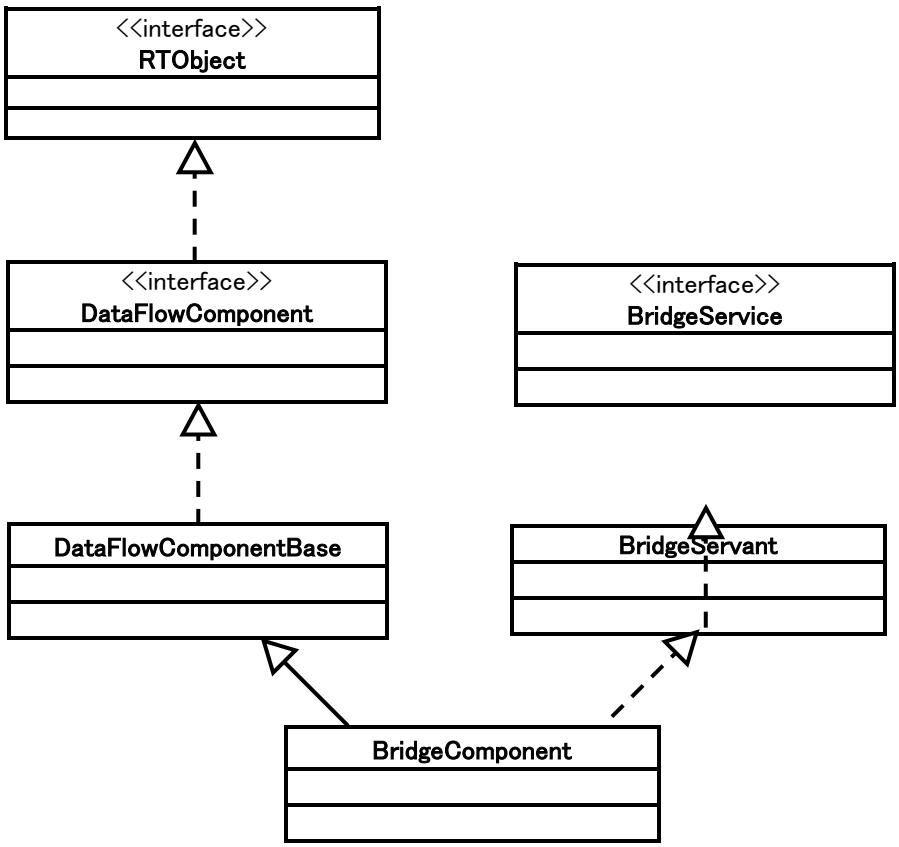


図 c-2-12 IDL のクラス構成

(3) IDL インタフェース

IDL を図 c-2-13 及び図 c-2-14 に示す。

```
// RTCHub.idl

#include "SDOPackage.idl"
#include "RTC.idl"

#pragma prefix "sec.co.jp"

//#define EXECUTION_HANDLE_TYPE_NATIVE long

module RTCHub
{
  typedef SDOPackage::UniqueIdentifier UniqueIdentifier;
  typedef SDOPackage::NVList NVList;

  typedef EXECUTION_HANDLE_TYPE_NATIVE DeviceComponentId_t;

  typedef RTC::ReturnCode_t ReturnCode_t;

  typedef RTC::LifeCycleState LifeCycleState;
  typedef RTC::ExecutionKind ExecutionKind;


  interface BridgeService;


  struct PortProfile
  {
    string name;
    DeviceComponentId_t owner_id;
    NVList properties;
  };
  typedef sequence<PortProfile> PortProfileList;

  struct ConnectorProfile
  {
    string name;
    UniqueIdentifier connector_id;
    PortProfileList port_profiles;
    NVList properties;
  };
  typedef sequence<ConnectorProfile> ConnectorProfileList;
```

図 c-2-13 IDL(1/2)

```
  struct ComponentProfile
  {
    DeviceComponentId_t id;
    string instance_name;
    string type_name;
    string description;
    string version;
    string vendor;
    string category;

    ExecutionKind kind;
    double rate;

    PortProfileList port_profiles;
    NVList properties;
  };
  typedef sequence<ComponentProfile> ComponentProfileList;



  interface BridgeService : SDOPackage::SDOService
  {
    boolean is_running(in DeviceComponentId_t id);

    ReturnCode_t activate_component(in DeviceComponentId_t id);
    ReturnCode_t deactivate_component(in DeviceComponentId_t id);
    ReturnCode_t reset_component(in DeviceComponentId_t id);

    LifeCycleState get_component_state(in DeviceComponentId_t id);

    double get_rate(in DeviceComponentId_t id);
    ReturnCode_t set_rate(in DeviceComponentId_t id, in double rate);
    ExecutionKind get_kind(in DeviceComponentId_t id);

    ReturnCode_t exit(in DeviceComponentId_t id);

    ComponentProfile get_component_profile(in DeviceComponentId_t id);
    ComponentProfileList get_member_component_profiles();

    ReturnCode_t connect(inout ConnectorProfile connector_profile);
    ReturnCode_t disconnect(inout ConnectorProfile connector_profile);

    ConnectorProfileList get_connector_profiles();
  };
};
```

図 c-2-14 IDL(2/2)

(4) IDL と各種プロファイルとのマッピング

PortProfile

データ		入力情報
name		Instance.Port Instance : /rts:RtsProfile/rts:Components/@rts: instanceName Port : /rtc:RtcProfile/rtc:DataPorts/@rtc:name
owner_id		/rtc:RtcProfile/rtc:BasicInfo/rtcExt:Properties[@rtcExt:name="node_id"]
properties	port.port_type	/rtc:RtcProfile/rtc:DataPorts/@rtc:portType
	dataport.data_type	/rtc:RtcProfile/rtc:DataPorts/@rtc:type
	dataport.dataflow_type	/rtc:RtcProfile/rtc:DataPorts/@rtc:dataflowType
	dataport.interface_type	/rtc:RtcProfile/rtc:DataPorts/@rtc:interfaceType
	dataport.subscription_type	/rtc:RtcProfile/rtc:DataPorts/@subscriptionType
	object_id	/rtc:RtcProfile/rtc:DataPorts/rtcExt:Properties[@rtcExt:name="object_id"]

ComponentProfile

データ		入力情報
id		/rtc:RtcProfile/rtc:BasicInfo/rtcExt:Properties[@rtcExt:name="node_id"]
instance_name		/rtc:RtcProfile/rtc:BasicInfo/@rtc:name
type_name		/rtc:RtcProfile/rtc:BasicInfo/@rtc:name
description		/rtc:RtcProfile/rtc:BasicInfo/@rtc:description
version		/rtc:RtcProfile/rtc:BasicInfo/@rtc:version
vendor		/rtc:RtcProfile/rtc:BasicInfo/@rtc:vendor
category		/rtc:RtcProfile/rtc:BasicInfo/@rtc:category
kind		/rtc:RtcProfile/rtc:BasicInfo/@rtc:activityType
rate		/rtc:RtcProfile/rtc:BasicInfo/@rtc:executionRate
port_profiles		/rtc:RtcProfile/rtc:DataPorts
properties	node_id	/rtc:RtcProfile/rtc:BasicInfo/rtcExt:Properties[@rtcExt:name="node_id"]
	bus_id	/rtc:RtcProfile/rtc:BasicInfo/rtcExt:Properties[@rtcExt:name="bus_id"]

ConnectorProfile

データ		入力情報
name		/rts:RtsProfile/rts:DataPortConnectors/@rts:name
connector_id		/rts:RtsProfile/rts:DataPortConnectors/@connectorId
port_profiles		/rts:RtsProfile/rts:DataPortConnectors/rts:sourceDataPort /rts:RtsProfile/rts:DataPortConnectors/rts:targetDataPort
properties	PortProfile.name	Instance.Port Instance : /rts:RtsProfile/rts:Components/@rts: instanceName Port : /rtc:RtcProfile/rtc:DataPorts/@rtc:name
	dataport.data_type	/rts:RtsProfile/rts:DataPortConnectors/@rts:dataType
	dataport.dataflow_type	/rts:RtsProfile/rts:DataPortConnectors/@dataflowType
	dataport.interface_type	/rts:RtsProfile/rts:DataPortConnectors/@interfaceType
	dataport.subscription_type	/rts:RtsProfile/rts:DataPortConnectors/@subscriptionType

c-2-4-3-5 ネームサービス

(1) ネームサービス

全体で1つであり、デバイス RTC はネームサービスに登録しない。プロキシ内の各デバイス RTC を操作するために、プロキシの中にサーバ処理を持たせ、デバイス RTC の ID を指定する。

(2) クライアントオブジェクトの取得

get_sdo_service(const char *id)の id には、"BridgeService"を指定する。

c-2-4-3-6 CORBA メソッド

is_running

is_running

[概要]

指定されたコンポーネントの起動状況を取得する。

[C++記述形式]

CORBA::Boolean is_running(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
TRUE	起動中
FALSE	停止中

[補足説明]

以下の場合も停止中とする。

- 指定されたデバイスコンポーネントが存在しない場合
- RTC-Lite Manager とネットワークの接続が切れている場合

activate_component

activate_component

[概要]

指定されたデバイスコンポーネントが INACTIVE 状態の場合に、ACTIVE 状態に遷移させる。

[C++記述形式]

RTC::ReturnCode_t activate_component(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
RTC::RTC_OK	ACTIVATE 成功
RTC::RTC_ERROR	ACTIVATE 失敗

[補足説明]

以下の場合も ACTIVATE 失敗とする。

- 指定されたデバイスコンポーネントが存在しない場合
- RTC-Lite Manager とネットワークの接続が切れている場合

deactivate_component

deactivate_component

[概要]

指定されたデバイスコンポーネントが ACTIVE 状態の場合に、INACTIVE 状態に遷移させる。

[C++記述形式]

RTC::ReturnCode_t deactivate_component(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
RTC::RTC_OK	DEACTIVATE 成功
RTC::RTC_ERROR	DEACTIVATE 失敗

[補足説明]

以下の場合も DEACTIVATE 失敗とする。

- 指定されたデバイスコンポーネントが存在しない場合
- RTC-Lite Manager とネットワークの接続が切れている場合

reset_component

reset_component

[概要]

指定されたデバイスコンポーネントが ERROR 状態の場合に、INACTIVE 状態に遷移させる。

[C++記述形式]

RTC::ReturnCode_t reset_component(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
RTC::RTC_OK	RESET 成功
RTC::RTC_ERROR	RESET 失敗

[補足説明]

以下の場合も RESET 失敗とする。

- 指定されたデバイスコンポーネントが存在しない場合
- RTC-Lite Manager とネットワークの接続が切れている場合

get_component_state

get_component_state

[概要]

指定されたコンポーネントのステータスを取得する。

[C++記述形式]

RTCHub::LifeCycleState get_component_state(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
RTC::ERROR_STATE	ERROR 状態
RTC::CREATED_STATE	生成状態
RTC::INACTIVE_STATE	INACTIVE 状態
RTC::ACTIVE_STATE	ACTIVE 状態

[補足説明]

以下の場合もエラー状態とする。

- 指定されたデバイスコンポーネントが存在しない場合
- RTC-Lite Manager とネットワークの接続が切れている場合

get_rate

get_rate

[概要]

指定されたデバイスコンポーネントのアクションの実行周期を取得する。

[C++記述形式]

CORBA::Double get_rate(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
rate	指定されたデバイスコンポーネントのアクションの実行周期

[補足説明]

今後の拡張

set_rate

set_rate

[概要]

指定されたデバイスコンポーネントのアクションの実行周期を、指定された値に設定する。

[C++記述形式]

RTC::ReturnCode_t set_rate(RTCHub::DeviceComponentId_t id, CORBA::Double rate)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID
I	rate	onExecute 実行周期[ms]

[復帰値]

シンボル / 値	説明
RTC::RTC_OK	設定成功
RTC::RTC_ERROR	設定失敗

[補足説明]

今後の拡張

get_kind

get_kind

[概要]

指定されたデバイスコンポーネントのアクティビティのタイプを取得する。

[C++記述形式]

RTCHub::ExecutionKind get_kind(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
RTC::PERIODIC	PERIODIC
RTC::EVENT_DRIVEN	EVENT_DRIVEN
RTC::OTHER	その他

[補足説明]

今後の拡張

exit

exit

[概要]

指定されたデバイスコンポーネントを終了させる。

[C++記述形式]

RTC::ReturnCode_t exit(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
RTC::RTC_OK	EXIT 成功
RTC::RTC_ERROR	EXIT 失敗

[補足説明]

以下の場合も EXIT 失敗とする。

- 指定されたデバイスコンポーネントが存在しない場合
- RTC-Lite Manager とネットワークの接続が切れている場合

get_component_profile

get_component_profile

[概要]

指定されたデバイスコンポーネントの、コンポーネント情報を取得する。

[C++記述形式]

RTCHub::ComponentProfile* get_component_profile(RTCHub::DeviceComponentId_t id)

[パラメタ]

I/O	パラメタ	説明
I	id	デバイスコンポーネント ID

[復帰値]

シンボル / 値	説明
ComponentProfile	指定されたデバイスコンポーネントの、コンポーネント情報

[補足説明]

指定されたコンポーネントが存在しない場合、設定値が初期値のコンポーネントプロファイルを返す。

get_member_component_profiles

get_member_component_profiles

[概要]

ブリッジコンポーネント内で複合化されている全てのデバイスコンポーネントの、コンポーネント情報リストを取得する。

[C++記述形式]

RTCHub::ComponentProfileList* get_member_component_profiles()

[パラメタ]

I/O	パラメタ	説明
―	―	―

[復帰値]

シンボル / 値	説明
ComponentProfileList	ブリッジコンポーネント内の全てのコンポーネント情報

[補足説明]

コンポーネントが一つも存在しない場合、コンポーネントプロファイル数が 0 のコンポーネントプロファイルリストを返す。

connect

connect

[概要]

指定されたコネクタプロファイルの内容でポートを接続する。
接続に成功した場合は、コネクタプロファイルに追加される。

[C++記述形式]

RTC::ReturnCode_t connect(RTCHub::ConnectorProfile& connector_profile)

[パラメタ]

I/O	パラメタ	説明
I	connector_profile	コネクタプロファイル

[復帰値]

シンボル / 値	説明
RTC::RTC_OK	CONNECT 成功
RTC::RTC_ERROR	CONNECT 失敗

[補足説明]

以下の場合も CONNECT 失敗とする。

- RTC-Lite Manager とネットワークの接続が切れている場合
- コネクタプロファイルに、接続する 2 種類のポート以外のポート情報が含まれていた場合
- コネクタプロファイルに設定された object_id が不正だった場合

disconnect

disconnect

[概要]

指定されたポート接続を切断する。
切断に成功した場合、コネクタプロファイルリストから削除される。

[C++記述形式]

RTC::ReturnCode_t disconnect(RTCHub::ConnectorProfile& connector_profile)

[パラメタ]

I/O	パラメタ	説明
I	connector_profile	コネクタプロファイル

[復帰値]

シンボル / 値	説明
RTC::RTC_OK	DISCONNECT 成功
RTC::RTC_ERROR	DISCONNECT 失敗

[補足説明]

以下の場合も DISCONNECT 失敗とする。

- 指定されたポート接続が存在しない場合
- RTC-Lite Manager とネットワークの接続が切れている場合

get_connector_profiles

get_connector_profiles

[概要]

ブリッジコンポーネント内で接続されているデバイスコンポーネントの接続情報リストを取得する。

[C++記述形式]

RTCHub::ConnectorProfileList* get_connector_profiles()

[パラメタ]

I/O	パラメタ	説明
―	―	―

[復帰値]

シンボル / 値	説明
ConnectorProfileList	ブリッジコンポーネント内の全ての接続情報

[補足説明]

コネクタプロファイルが一つも存在しない場合、コネクタプロファイル数が 0 のコネクタプロファイルリストを返す。

c-2-4-4 制約事項

Bridge RTC と OpenRTM-aist のコンポーネントとの相違点

・RTSystemEditor の RTS プロファイル読込による、コンポーネントの自動起動には対応していない。
・動的なポートの公開／非公開切り替えに対応していない。
・複合コンポーネントとして扱うが、Bridge RTC とデバイス RTC の ExcutionContext の参加関係は、compositeType に依らず常に独立とする。

その他制約事項

・BridgeServant は、複数から同時に接続することができない。
・Bridge コンポーネントが EXIT コマンドを受信すると、「未定義状態」になる場合がある。

c-2-5 住宅における実証評価

c-2-5-1 システム構成

これまで開発してきた本システムの実証評価をおこなうため、産業技術総合研究所内にモデルルームを建設した。そして、モデルルーム内に窓やブラインドなどの住宅設備と、それらを制御するモジュールやソフトウェアを導入した(図 c-2-15)。今回設置したシステムの構成を図 c-2-16 に、システムを構成する各モジュールの仕様を表 c-2-21 に示す。

図 c-2-15 モデルルーム

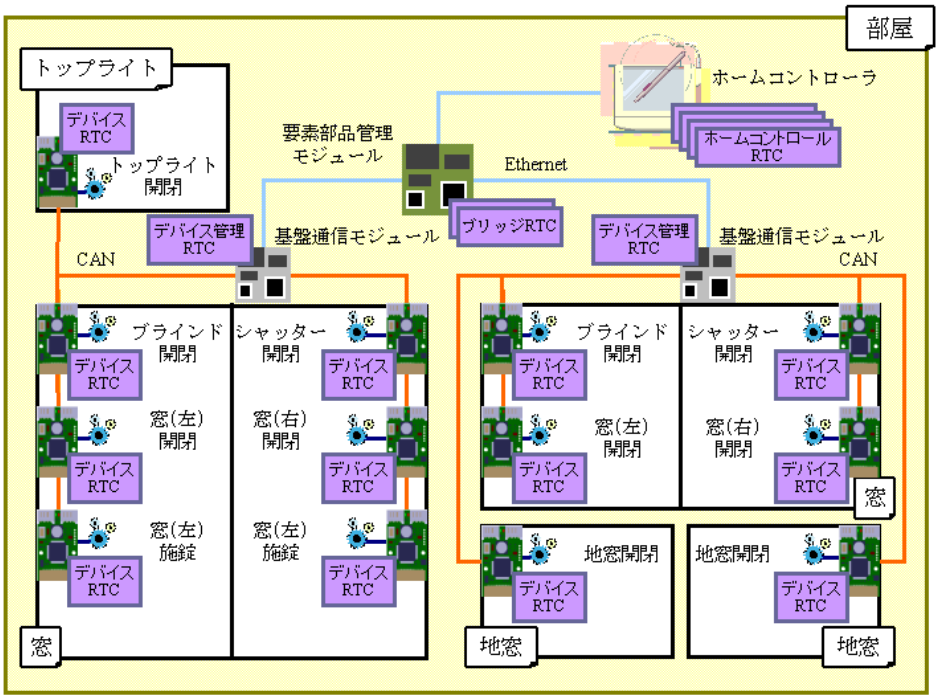


図 c-2-16 システム構成

表 c-2-21　RT モジュール仕様

ホーム コントローラ	CPU RAM OS	:Intel Core2Duo 2.00GHz :1GB :Windows XP
要素部品管理 モジュール	CPU Flash RAM OS	:SH7764 / 324MHz :32MB :128MB :Linux
基盤通信 モジュール (CAN)	CPU Flash RAM OS	:SH7216 / 200MHz :1MB :128KB :TOPPERS / ASP
ドライバ モジュール (CAN)	CPU Flash RAM OS	:ARM7 / 60MHz :256KB :16KB :なし

今回の実証では、インテリジェント空調システムの事例として、図 c-2-17 に示すホームコントローラを iPad から操作し、モデルルームに設置された窓、シャッター、ブラインド、トップライト、及び地窓を動作させた。また、設定したモードにより、それぞれの窓に設置されたブラインドとシャッターを連携動作させた。

図 c-2-17 ホームコントローラ

本システムの RTC 構成を図 c-2-18 に示す。

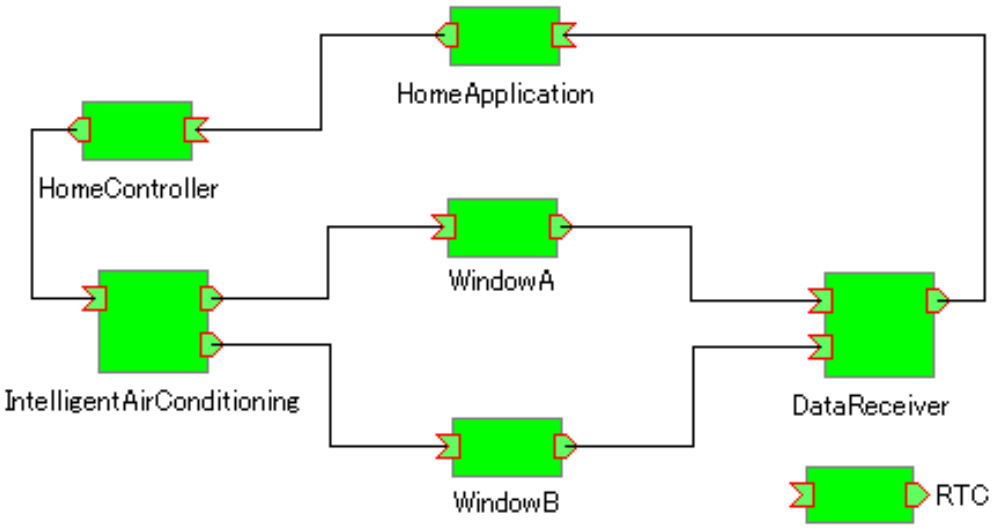


図 c-2-18RT コンポーネント構成

HomeApplication、HomeController、DataReceiver、及び IntelligentAirConditioning はホームコントローラ上で動作し、GUI のアプリケーションとの連携やコマンドの振分け、デバイスのステータス収集・管理をおこなう RTC である。窓やブラインドを制御するデバイス RTC は数が多いため、窓ごとの単位で WindowA、WindowB として仮想的に複合化し、1 つの RTC として扱う。これにより、省資源マイコンの処理負荷を軽減させることができる。また、WindowA、WindowB は RTC HUB により、ブリッジ RTC として生成される。このブリッジ RTC により、OpenRTM.NET 上で動作する RTC と、miniRTCs 上で動作する RTC との相互接続が可能になる。これらの RTC により、本システムを実現する。

c-2-5-2　機能評価

今回、本システムをモデルルームに導入し、これまで開発した RT ミドルウェアについて以下の評価をおこなった。

(1) 数の RT ミドルウェア間のデータ入出力
(2) miniRTCs による分散管理
(3) プラグアンドプレイ

これらの評価結果について、以降説明する。

(1) 複数の RT ミドルウェア間のデータ入出力

今回の実証により、先に述べた本システムの構成において住宅設備を制御し、複数の RT ミドルウェアが機能的に動作することを確認することができた。

住宅設備を制御する際のデータフローを図 c-2-17 に示す。ホームコントローラのボタンを押すと、ホームコントロール RTC のデータポートからデバイス制御コマンドが出力される。出力されたコマンドはブリッジ RTC、デバイス管理 RTC を経由し、デバイス RTC のデータポートに入力される。そして、デバイス RTC がコマンドに従い、モータを動作させる(図 c-2-19 データフロー①、②、③)。また、デバイス RTC からはデバイスのステータスが送信され、最終的にホームコントロール RTC がステータスを受信する。そして、ホームコントローラの画面上にデバイスのステータスを表示する(図 c-2-19 データフロー④、⑤、⑥)。

これら一連のデータの入出力を確認することにより、複数の RT ミドルウェア間でのデータポートによるシームレスなデータの入出力が可能であることを確認することができた。

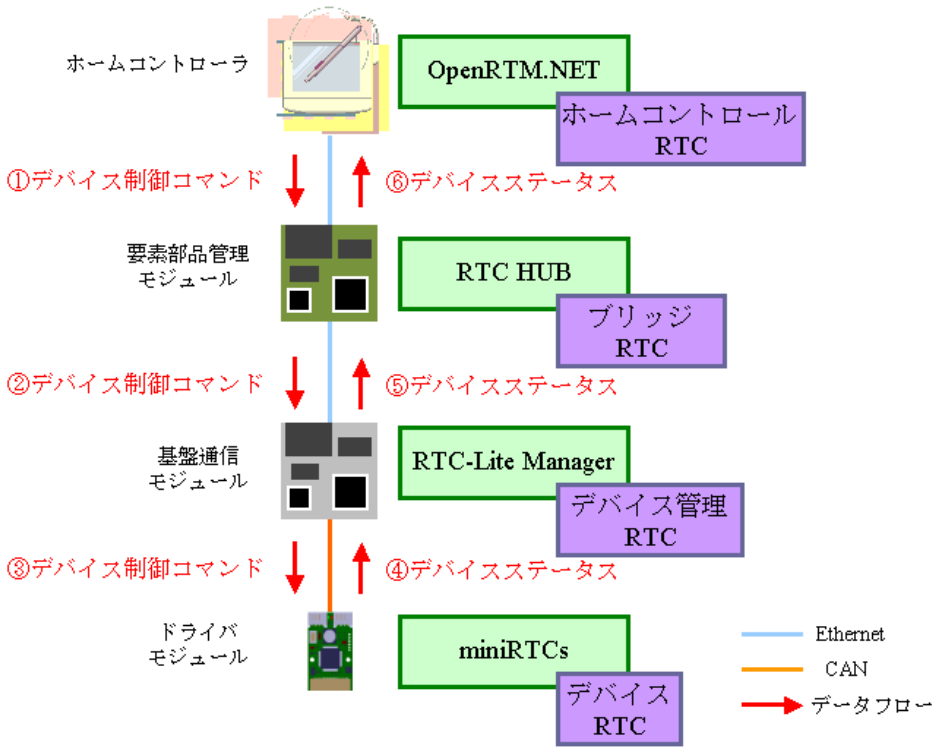


図 c-2-19 データフロー

(2) miniRTCs による分散管理

本システムのように、多数のモジュールにより構成されるシステムでは、一部のモジュールが故障した際にシステム全体を停止させない仕組みが必要である。そのため我々は、分散管理のアーキテクチャを採用し、各モジュール単位で独立した動作が可能となるようにしている。

今回の実証では、分散管理されたモジュールとして、窓開閉のデバイス RTC と窓施錠のデバイス RTCとの間の独立した動作を確認した。窓開閉のデバイスRTCは、「Open」というコマンドを受信すると、窓施錠デバイス RTC へ開錠のコマンドを直接送信し、開錠させることができた。
このように、ホームコントローラから RTC HUB を介したデバイス RTC の制御だけではなく、miniRTCs による RTC の分散管理が可能であることを確認した。

(3) プラグアンドプレイ

多数のモジュールによって構成されるシステムでは、これまで述べてきた点以外に、ユーザビリティの向上についても考慮する必要がある。システムを構成する大量の RTC の活性化やデータポートの接続を手動でおこなうのは煩雑であり、居住者も用意にシステムを導入することができない。ユーザビリティの向上のためには、各機器の電源を投入しただけでシステムが利用可能になることが望まれる。

本システムでは、RTC のプラグアンドプレイ機能を実装し、ユーザビリティを向上させている。プラグアンドプレイは、ハートビートと呼ばれるメッセージの送信によって実現している。ハートビートには機器の情報が含まれており、各モジュールは電源を投入されるとハートビートを送信し、自動的に機器の存在を通知する。

今回の実証では、このハートビートを利用し、RTC を認識することで、RTC を自動的に活性化することができた。また、あらかじめ用意したRTSプロファイルの情報を基に、各データポートの接続を自動的におこなうことができた。

このように、各モジュールの電源を投入するだけでシステムとして稼動可能な状況になることを確認することができた。

c-2-5-3 性能評価

今回、性能評価として、ホームコントロール RTC からデバイス RTC へコマンドを送信した際の RTC HUB におけるスループットを測定した。このとき送信したデータサイズを表 c-2-22、測定結果を表 c-2-23 に示す。なお、表 c-2-24 内の番号(②、③、④、⑤)は、図 c-2-19 に示したデータフローの番号に対応する。

表 c-2-22 データサイズ

データ	サイズ
TCP コマンド送信	30byte
TCP コマンド返信	31byte
CAN コマンド送信	60bit
CAN コマンド返信	68bit

表 c-2-23 性能測定結果

測定対象	測定時間 [ms]
スループット (②+③+④+⑤+ RTC-Lite Manager 処理時間+ miniRTCs 処理時間)	8.32
TCP 通信時間(②+⑤)	7.54
RTC-Lite Manager 処理時間	0.06
CAN 通信時間(③+④)+ miniRTCs 処理時間	0.72

測定の結果、RTC HUB におけるスループットは、ホームコントローラから制御する際に問題とならない時間であることが確認できた。また、リアルタイム制御が必要である RTC-Lite Manager、miniRTCs 処理時間と CAN 通信時間については、合計 1ms 以内に収まることが分かった。これらの処理はモータの制御が必要なため、処理時間の目標を 1ms としていたが、オーバーヘッドがなく目標範囲内に収まることが分かった。これらの結果から、これまで開発した RT ミドルウェアがモータの制御に適用可能であることを示すことができた。

(c-3) 安全性の検討(委託先:産業技術総合研究所)

c-3-1 概要

RT システムを導入する際、その危険性は従来の駆動源のない住宅設備と比較して非常に高い物となる。近年では自動ドアや自動シャッター等、駆動源を有する住宅設備も普及が進んでいるが、基本的にシステムとしてクローズであり、製作するメーカー内において、その住宅設備のみの安全性を評価することで、危険性は低減できる。しかし、本事業で構築する RT システムは、各機器がネットワークで連携動作することになるため、システムとして複雑であり、通信状態によっては、事故発生につながる可能性は、既存のシステムより大きい。そこで、本件では、RT システムを導入した住宅システム全体ににおける安全性を評価する手順を構築する。本件での目標、並びに達成度は表 c-3-1 の通りである。

表 c-3-1 研究項目に対する目標ならびに達成度

目標	研究開発成果	達成度
RT システムに対する安全性検討手順の構築		
(1)本実証 RT システムに対しての安全性の検討手順を構築する.	(1)SysML を利用し、システムを見える化し、構造図に基づくリスク評価手順を示した。	(1)達成
(2)プロジェクト終了後, 開発成果の一つとして, 様式を含めた安全性チェック手順を公開する.	(2)報告書で公開を行う。	(2)達成見込み(2011 年 8 月)

c-3-2 安全性評価の流れ

本事業において構築される実証 RT システムとして住宅内の設備機器と各種センサがネットワークを介してつながった RT 住宅システムを対象としている。すなわち、各 RT 要素部品は、住宅設備に基盤通信モジュールが付加された部品であり、それぞれがネットワークに並列に接続されている構成で構築されている。そのため、各機器の接続は並列分散型の構成となり、一見してそのシステム構成が複雑に見える。本開発項目は、RT 住宅システムの安全性を評価するための手順を構築することである。安全性を評価するためには、システムの構成を見える化することが重要となる。ここで一般にシステムの全体を表現する手法として、SysML(OMG Systems Modeling Language)という「モデリング言語」が利用されている。SysML は、「UML(Unified Modeling Language)」と同様、OMG(Object Management Group)によって仕様が策定されている。また、UML をベースに、UML でソフトウェアアーキテクチャを構築するソフトウェアエンジニアリング手法に基づき構築されている。SysML では UML と同様、各種設計図であるダイアグラムが利用される。また、代表的なダイアグラムとして、振る舞い図、要求図、構造図の 3 つで構成される。特に今回のシステムを表現するためには構造図を作成することで、システムの構成を見える化し、その構造図に基づいて、安全性の検討であるリスクアセスメントを行う。リスクアセスメントに利用する構造図として、ハードウェア構造図、RT コンポーネント構造図、ネットワーク構造図の 3 種類の構造図を構築し、これらに基づいて安全性を評価した。評価の全体の流れを図 c-3-1 に示す。

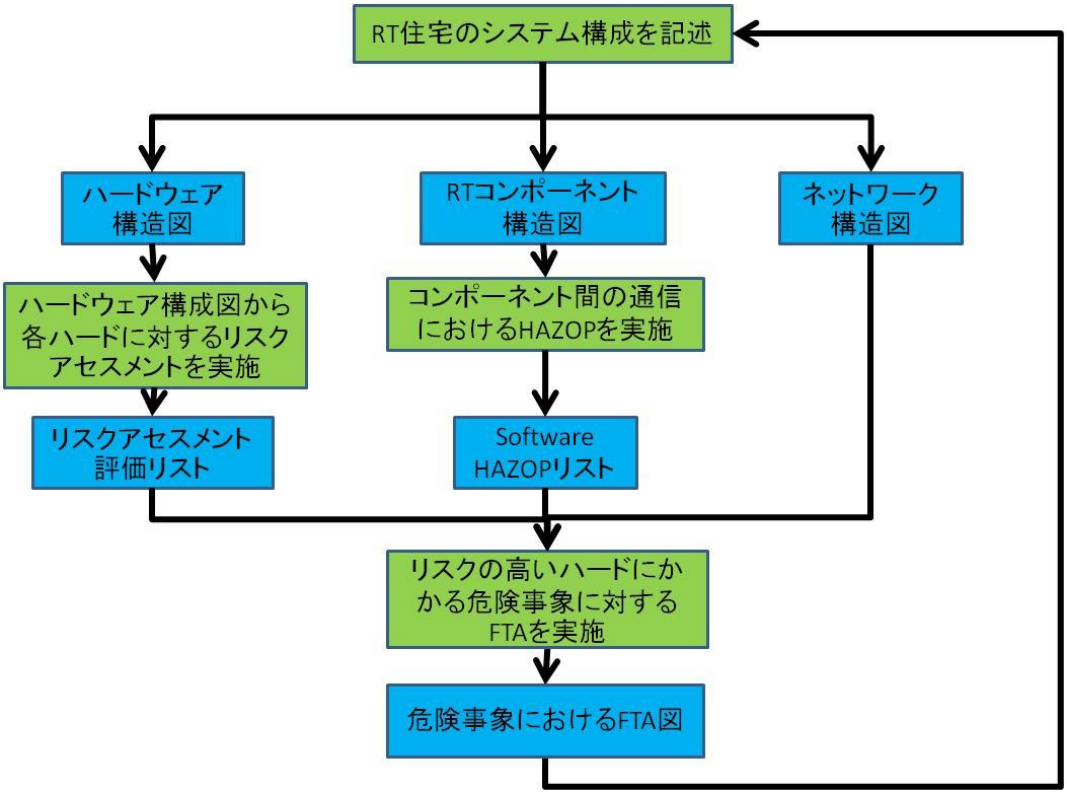


図 c-3-1　安全性評価手法の流れ

大まかな流れとして、住宅システムのハードウェア構成を表現する「ハードウェア構造図」を構築し、各機器に対するリスクアセスメントを実施する。これにより、各機器のリスク項目の大小を評価し、「リスクアセスメント評価リスト」を作成する。

ソフトウェア構成については、本住宅システムは RT ミドルウェアで構成されていることから、各機器の制御用ソフトウェアは RT コンポーネントとして、モジュール化されている。各 RT コンポーネント内の動作に関しては、そのコンポーネントを開発した企業が責任を有することになるため、そのコンポーネント内の動作は保障されているものとする。すなわち、システム化の際に不都合が発生するリスクは、コンポーネント間の通信に係る部分となる。そこで、コンポーネント間の関係図を「RT コンポーネント構造図」として図示する。これに対し、各 RT コンポーネント間の通信状態に関して Hazard and Operability Studies for Software（Software HAZOP）を実施し、送信側の RT コンポーネントから受信側の RT コンポーネントに対して、信号が来ない場合、予期しない時に信号が来る場合、信号伝達が早すぎる場合、信号伝達が遅すぎる場合、信号情報が誤っている場合の以上6パターンについて、原因、影響、リスクを評価し、「Software HAZOP リスト」を作成する。

本システムは基本的にネットワークによる各機器の通信で制御を行っている。そのため、各機器同士の通信形態が無線なのか有線なのかで、その通信の安定性のリスクが異なる。そのため、各機器のネットワークがどのプロトコルを利用しているかを示した「ネットワーク構造図」をつくることが要求される。

以上の「リスクアセスメント評価リスト」、「Software HAZOP リスト」、「ネットワーク構造図」の 3 種類の資料から、リスク評価の高い項目に対して、Fault Tree Analysis を実施し、「危険事象における FTA」を作成する。この結果から、そのリスクに対しての改善を行い、再度システムに反映することで、最終的に「リスクアセスメント評価リスト」、「Software HAZOP リスト」のリスク項目が許容値になるまで繰り返す。以上の流れで安全性の評価を行うことで、漏れのないリスクアセスメントが実現可能となる。

c-3-3　ハードウェア構造図を利用したリスクアセスメント

具体的に、今回構築された住宅システムに対して、以上で示した安全性の評価を進めることとする。まず、システム全体を表現するため、図 c-3-2 に本実証における住宅システムの全体構造図を示す。

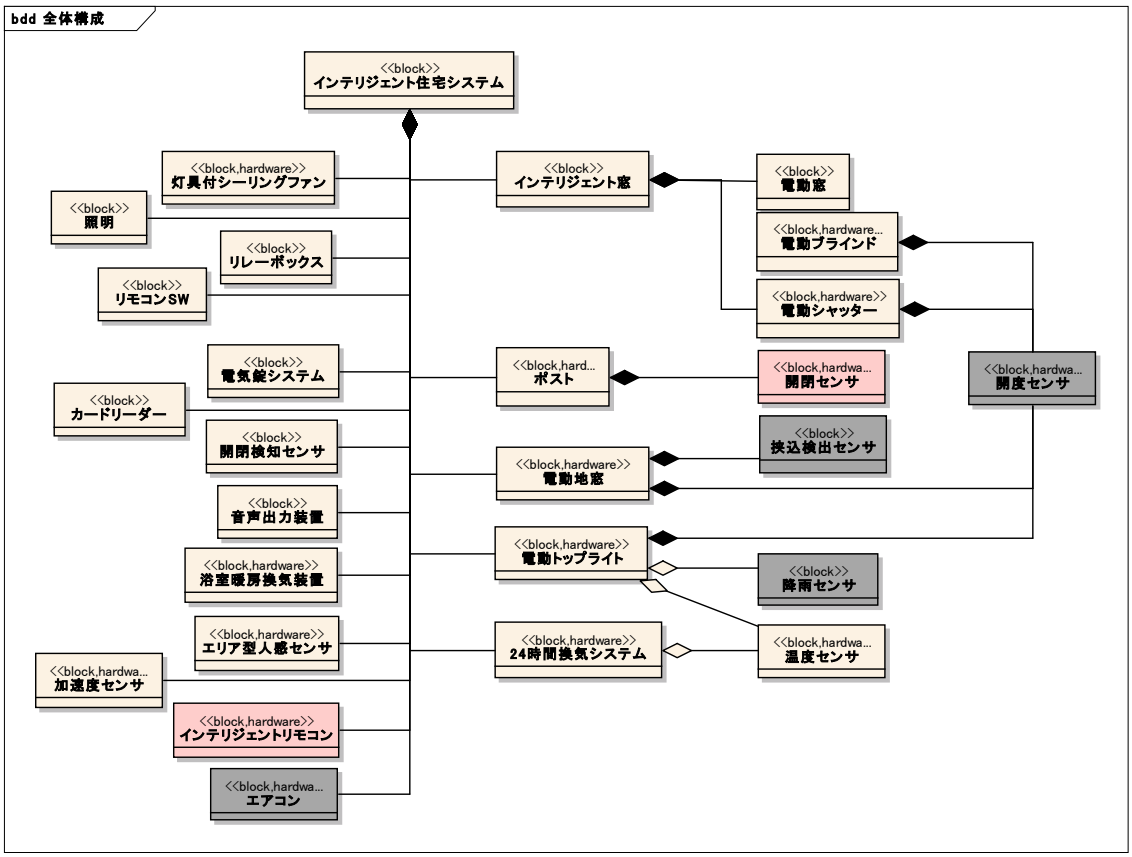


図 c-3-2　全体構造図

インテリジェント住宅システムという統合システムの構成は、インテリジェント窓や照明等の各種サブシステムが図 3-2 に示されるように含まれている。また、インテリジェント窓に関しては、さらにその中のサブシステムとして、電動窓、電動シャッターで構成されていることがわかる。

図 c-3-3 にインテリジェント窓のハードウェア構造図を示す。

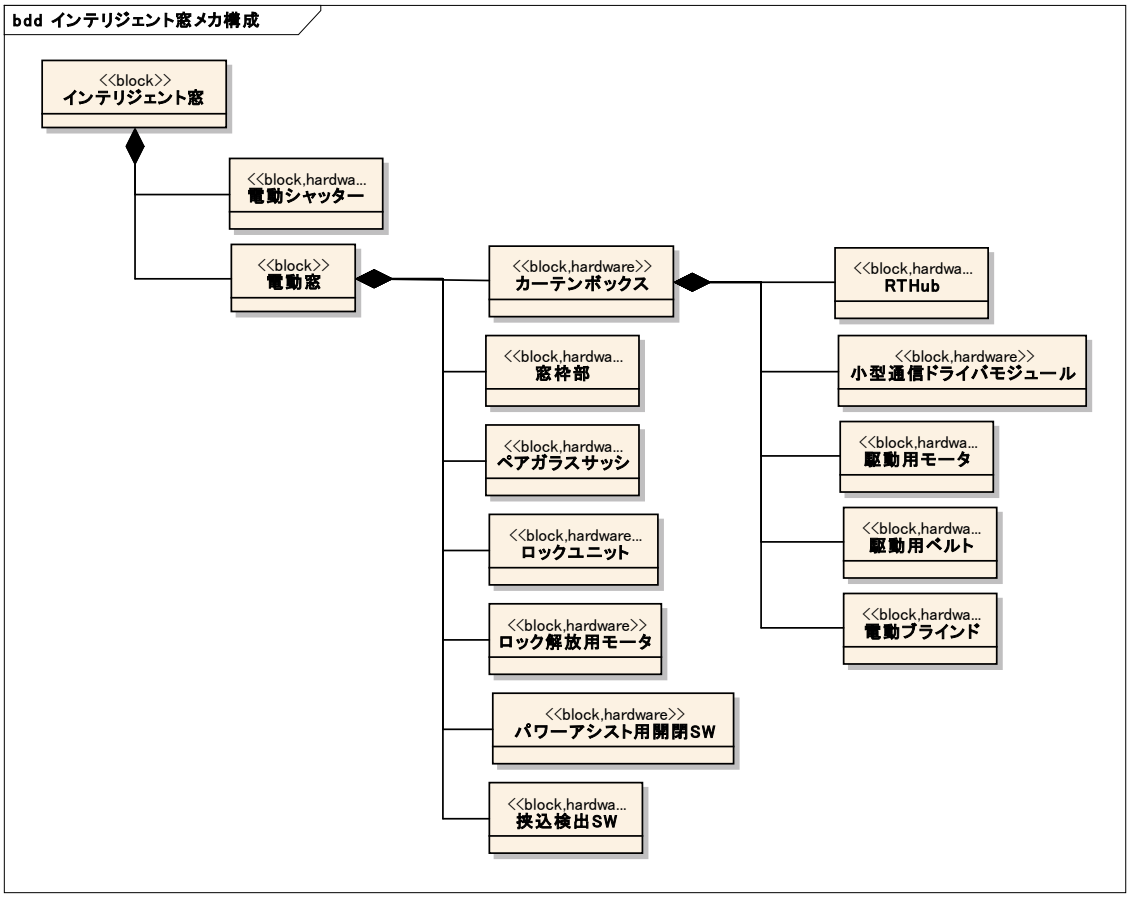


図 c-3-3　インテリジェント窓ハードウェア構造図

このハードウェア構造図は、インテリジェント窓の構成されるハードウェアの設置箇所の観点で図示したものである。電動窓はカーテンボックスや窓枠部等の部材で構成されており、さらにカーテンボックス内に RTHub やドライバモジュールといった電装部品が組み込まれていることを示している。

別の構造図として、インテリジェント窓のハードウェアの関連性を示した構造図を図 c-3-4 に示す。

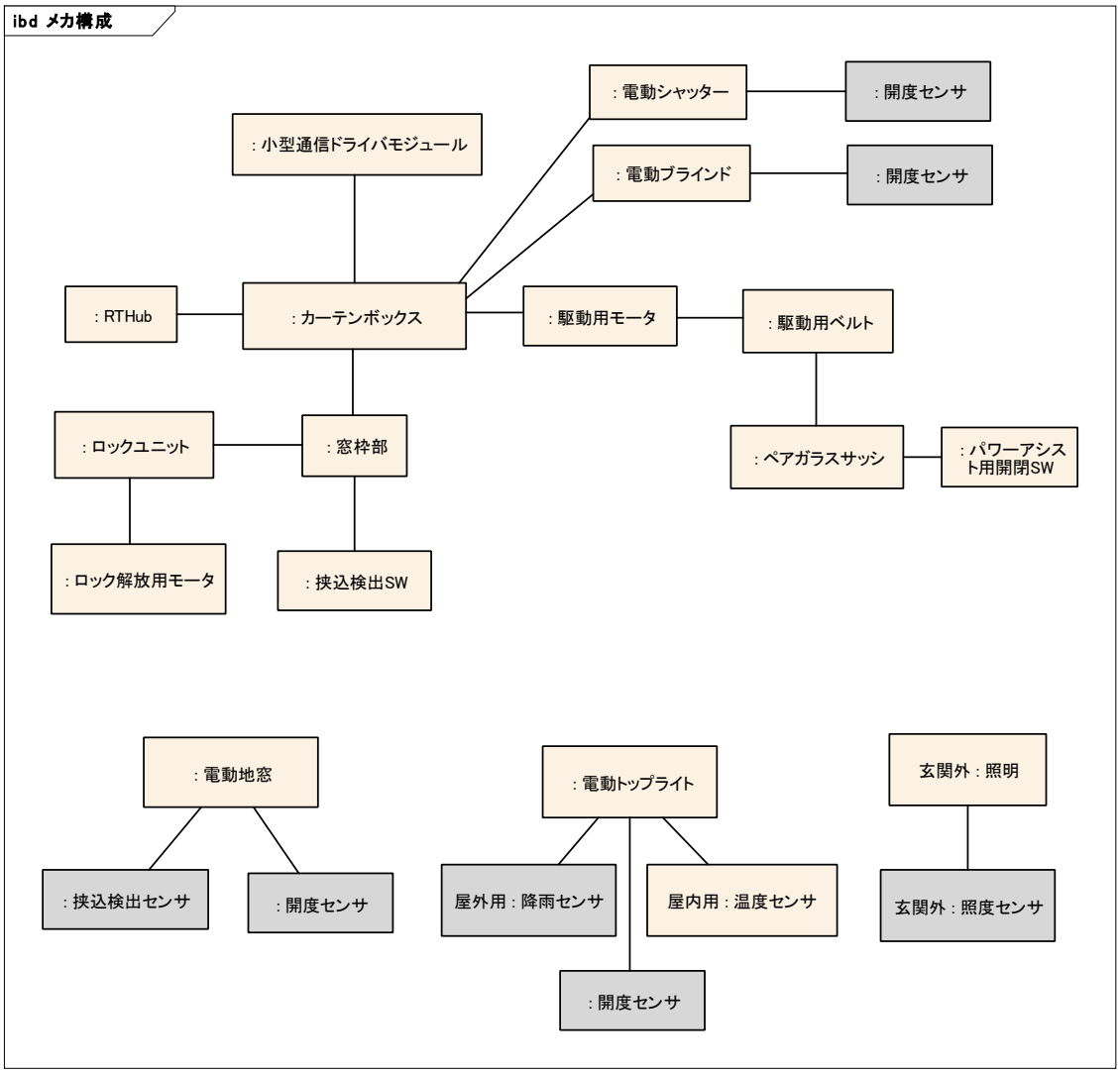


図 c-3-4　インテリジェント窓のハードウェア構造図（機器の関係図）

例えば、カーテンボックス内に駆動用モーターが設置され、駆動用ベルトを介してペアガラスサッシが動くようになっている。また、ペアガラスサッシにはパワーアシスト用開閉スイッチが設置されているといったハードウェアの関係性を容易に解釈することが可能である。

機器の設置場所に関する関係性と合わせて、各機器が機械・電気的に結合している関係性も必要となる。これに対して図 c-3-5 に示すように電気的な結合状態を示すハードウェア構造図を示す。

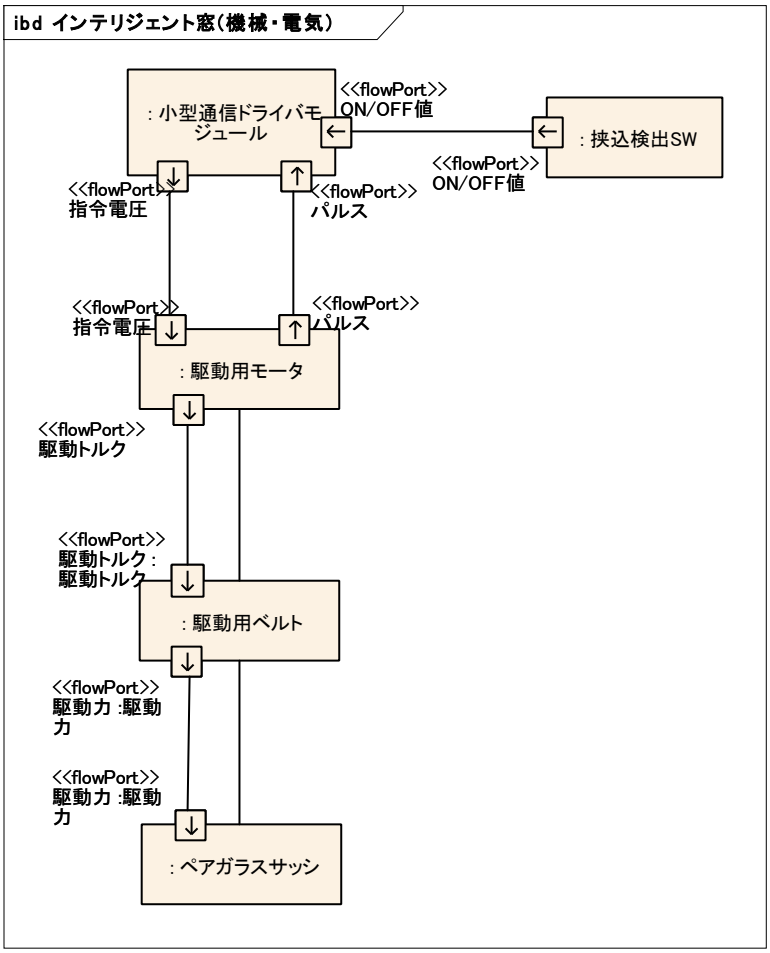


図 c-3-5　インテリジェント窓のハードウェア構造図(機械・電気系の関係図)

小型通信ドライバモジュールと駆動用モータは図 c-3-4 を見る限りカーテンボックス内に並列的に設置している機器として表現されている。しかし、電気的にはドライバモジュールから指令電圧が駆動用モーターに供給され、駆動用モータは、エンコーダからのパルス値をドライバモジュールに返す。また、駆動用モータに駆動用ベルトがつながっているが、そこには駆動トルクが働き、さらに駆動用ベルトにつながったペアガラスサッシに駆動力が加わり、結果的にペアガラスサッシが開閉することになる。以上の流れが図 c-3-5 によって表現されていることがわかる。

以上のハードウェア構造図の各設置部品をリスト化し、それぞれに対してリスクアセスメントを行う。表c-3-2 はインテリジェント窓に関するリスクアセスメント評価リストである。

まず、それぞれ対象となる設備機器の危険が起こる"Type"をカテゴライズする。Mechanical, Electrical, Thermal 等、いくつかの危険の"Type"でカテゴライズされており、部品によっては、複数の危険のタイプが存在するケースもあり得る。これらのカテゴライズは ISO14121-1(Safety of machinery -Risk assessment- Part1:Principles)の事例を基にしている。今回は ISO14121-1 で示されるカテゴライズの一部を使用しているが、これらの項目はその部品の特性次第で適宜選択していく必要もある。

次に危険性の原因となる"Origin"を検討する。これに対して結果として起こりえる事象を"Consequences"として列挙する。たとえば「ペアガラス」は動く設備である。そこで、"Type"として"Mechanical"な要素が考えられ、その"Origin"の一つを"Acceleration, deceleration"としたとき、これに対する危険な結果"Consequences"は"Crushing"となる。また、その危険の被害を受ける可能性のある対象者は"ユーザー"だけではなく、"施工業者", "メンテナンス業者"が考えられる。

各設備について、考えられるリスクの状況をリスト化し、次にそのリスクの"重篤度(Severity)"、曝露可能性(Exposure)"、" 発生頻度(Occurrence)"、" 回避可能性(Avoidance)"を見積り、最終的にリスクレベルに基づいた評価を行う。この評価の点数に応じて、ある評価点以上における設備機器を抽出し、今後はその設備に対する危険事象に対する原因の解析を進めるというプロセスをおこなう。

表 c-3-2　インテリジェント窓に関するリスクアセスメント評価リスト

No	対象部品	Type	Origin	Consequences	対象者	リスク見積もり				リスクレベル	評価	備考
						重篤度 Severity	曝露可能性 Exposure	発生頻度 Occurrence	回避可能性 Avoidance			
1	ペアガラスサッシ	Mechanical	Acceleration,deceleration	Crushing	ユーザ 施工業者 メンテナンス業者	Serious	Regular		Even chance			
2	ペアガラスサッシ	Mechanical	Moving elements	Crushing	ユーザ 施工業者 メンテナンス業者	Serious	Regular		Even chance			
3	ペアガラスサッシ	Noise	Moving parts	Discomfort	ユーザ	Slight	Regular		Possible			
4	インテリジェント窓ユニット	Mechanical	Gravity	Slipping,tripping and falling	ユーザ 施工業者 メンテナンス業者	Serious	Regular		Highly unlikely			
5	インテリジェント窓ユニット	Ergonomic	Posture	Discomfort	ユーザ 施工業者 メンテナンス業者		Rare		Possible			
6	駆動用モータ	Electrical	Overload	Burn	ユーザ	Serious	Regular		Highly unlikely			
7	駆動用モータ	Thermal	Objects or materials with a high or low temperature	Scald	ユーザ 施工業者 メンテナンス業者	Serious	Rare		Highly unlikely			
8	駆動用モータ	Noise	Unbalanced rotating parts	Discomfort	ユーザ	Slight	Regular		Possible			
9	駆動用ベルト	Mechanical	Falling object	Impact	ユーザ	Serious	Regular		Highly unlikely			
10	RTHub	Electrical	Short-circuit	Burn	ユーザ	Serious	Regular		Highly unlikely			
11	RTHub	Thermal	Objects or materials with a high or low temperature	Scald	ユーザ 施工業者 メンテナンス業者	Serious	Rare		Highly unlikely			
12	ロック駆動用モータ	Electrical	Overload	Burn	ユーザ	Serious	Regular		Highly unlikely			
13	ロック駆動用モータ	Thermal	Objects or materials with a high or low temperature	Scald	ユーザ 施工業者 メンテナンス業者	Serious	Rare		Highly unlikely			
14	ロック駆動用モータ	Noise	Unbalanced rotating parts	Discomfort	ユーザ	Slight	Regular		Possible			
15	電動ブラインド	Mechanical	Gravity	Slipping,tripping and falling	ユーザ	Slight	Regular		Possible			
16	電動ブラインド	Ergonomic	Posture	Discomfort	ユーザ		Rare		Possible			

c-3-4 ソフトウェア構造図を利用した Hazard and Operability Studies （HAZOP）

住宅システムのソフトウェアはすべて、RT コンポーネント化されており、各 RT コンポーネントの相互通信によって、連携動作されている。ここで、各 RT コンポーネントは各設備メーカーが安全評価を踏まえた上で開発しているという前提としている。よって、ソフトウェアにおけるリスクは、各社から提供される RT コンポーネントが相互に通信する際に、その通信状況によってリスクが発生すると考えられることから、統合した際の各 RT コンポーネントの通信状態によるリスクを見積もる。

まずハードウェア構造図と同様に、ソフトウェア面からもシステムの見える化を行うことが必要となる。そこで、ソフトウェア構造図を SysML 上で作成する。図 c-3-6 はインテリジェント窓おけるソフトウェア構造図である。RT ミドルウェアを利用していることで、各ブロックの粒度は RT コンポーネントと一対一で記載できることがわかる。図 c-3-7 にこのインテリジェント窓との連携を示したインテリジェント空調のソフトウェア構造図を示す。

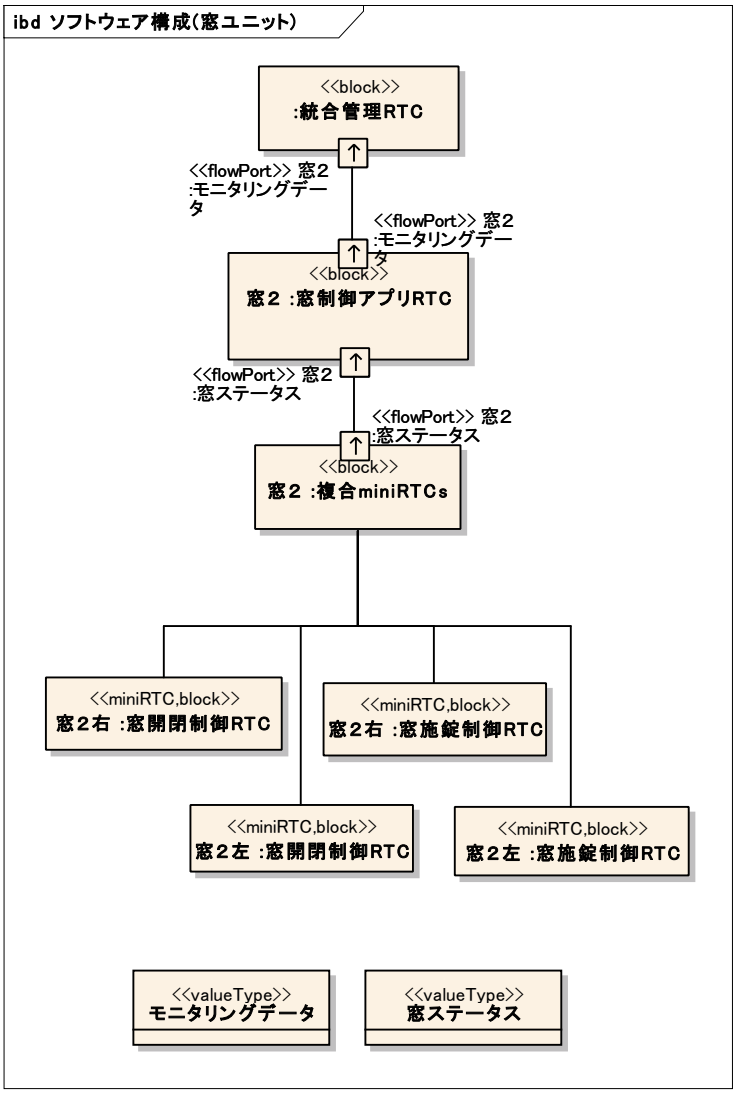


図 c-3-6 インテリジェント窓におけるソフトウェア構造図

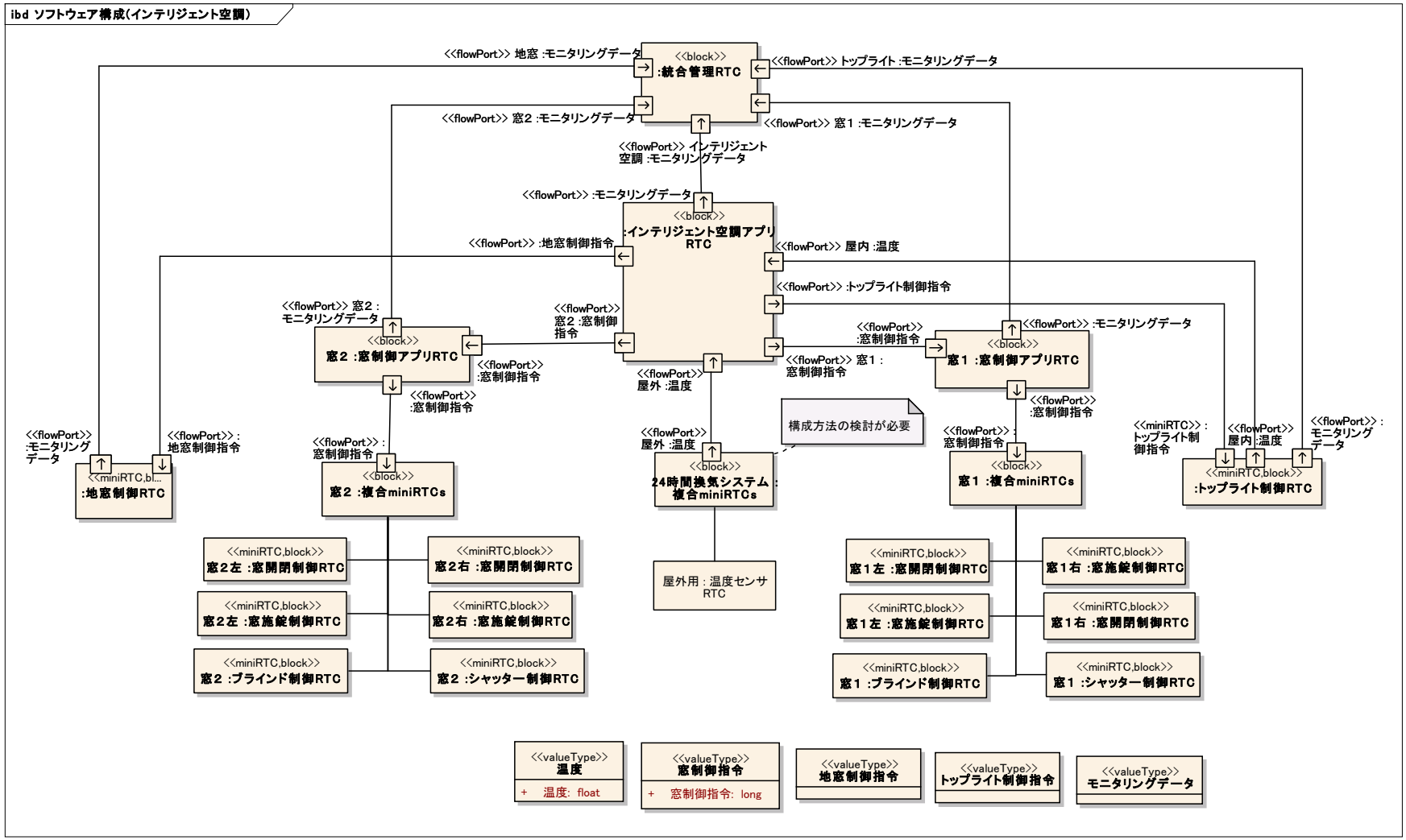


図 c-3-7 インテリジェント空調のソフトウェア構造図

前節のハードウェア構造図からのリスクアセスメント評価において、リスクの高い設備機器としてインテリジェント窓の開け閉めにかかる動作が挙げられるとしたとき、その自動制御にかかるソフトウェア部分は図 3-1-6 および図 3-7 で表現できることから、このソフトウェア構造図を基に、各コンポーネントの通信状況における Hazard and Operability Studies for Software（Software HAZOP）を行う。HAZOP とは、化学プラントの安全性を評価するために開発されたものであり、プロセスパラメータの目標値（目標状態）からのずれを想定し、そのずれの起こる原因と発生する危険事象を解析する。さらにその原因から危険事象に進展するのを防護する機能を評価し、対策を検討するというものである。この考え方をソフトウェアに応用したものが Software HAZOP であり、HAZOP における各化学プラントのプロセスがソフトウェアのモジュールに相当する。すなわち、化学プロセスにおける目標状態のずれをソフトウェアモジュール間の相互通信のタイミングのずれとして HAZOP を評価することになる。本システムは RT ミドルウェアで構成されているため、ソフトウェアモジュールは RT コンポーネント単位で構成されているため、各 RT コンポーネント間の相互通信に係る通信データが、「来ない」、「予期しないタイミングで来る」、「早く来る」、「遅く来る」、「間違っている」といった、5 パターンの状況が発生した場合のシステムの影響を見積り、そのリスクを評価する。表 3-3 にインテリジェント空調のソフトウェア構造図に記載されている RT コンポーネントに対する Software HAZOP を行った結果をまとめる。

表 c-3-3 RT コンポーネントに対する Software HAZOP リスト

取り込み元モデル　　住宅システム
取り込み元ダイアグラム　　住宅システム::ソフトウェア構成(インテリジェント空調)

送信ブロック名	送信	送信ポート型	受信ブロック名	受信	受信ポ	ガイドワード	解釈	原因	Local影響	System影響	リスク指数	対策
複合miniRTCs::24時間換気シス	屋外	温度	インテリジェント空調アプリR	屋外	温度	Omission	温度情報が来ない	断線，セン	屋外温度を把握できない	効果的な空調を実行できない	低	
複合miniRTCs::24時間換気シス	屋外	温度	インテリジェント空調アプリR	屋外	温度	Commission	予期せぬタイミングで温度情報が	ノイズ，誤	屋外温度を把握できない	効果的な空調を実行できない	低	
複合miniRTCs::24時間換気シス	屋外	温度	インテリジェント空調アプリR	屋外	温度	Early	温度情報が早く到着する	ノイズ，誤	屋外温度を素早く把握でき	温度変化に素早く対応できる	低	
複合miniRTCs::24時間換気シス	屋外	温度	インテリジェント空調アプリR	屋外	温度	Late	温度情報が到着するのが遅れる	ノイズ，誤	屋外温度の把握にタイムラ	空調制御が実行されるまでにタ	低	
複合miniRTCs::24時間換気シス	屋外	温度	インテリジェント空調アプリR	屋外	温度	Value	温度情報が誤っている	ノイズ，誤	屋外温度を把握できない	効果的な空調を実行できない	低	
トップライト制御RTC	屋内	温度	インテリジェント空調アプリR	屋内	温度	Omission	温度情報が来ない	断線，セン	室内温度を把握できない	効果的な空調を実行できない	低	
トップライト制御RTC	屋内	温度	インテリジェント空調アプリR	屋内	温度	Commission	予期せぬタイミングで温度情報が	ノイズ，誤	室内温度を把握できない	効果的な空調を実行できない	低	
トップライト制御RTC	屋内	温度	インテリジェント空調アプリR	屋内	温度	Early	温度情報が早く到着する	ノイズ，誤	室内温度を素早く把握でき	温度変化に素早く対応できる	低	
トップライト制御RTC	屋内	温度	インテリジェント空調アプリR	屋内	温度	Late	温度情報が到着するのが遅れる	ノイズ，誤	室内温度の把握にタイムラ	空調制御が実行されるまでにタ	低	
トップライト制御RTC	屋内	温度	インテリジェント空調アプリR	屋内	温度	Value	温度情報が誤っている	ノイズ，誤	室内温度を把握できない	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC		地窓制御指令	地窓制御RTC		地窓制	Omission	制御指令が来ない	断線，セン	地窓が開閉されない	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC		地窓制御指令	地窓制御RTC		地窓制	Commission	意図しない制御指令がくる	ノイズ，誤	意図しない開閉が発生する	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC		地窓制御指令	地窓制御RTC		地窓制	Early	制御指令が早く到着する	ノイズ，誤	早めに開閉される	温度変化に素早く対応できる	低	
インテリジェント空調アプリRTC		地窓制御指令	地窓制御RTC		地窓制	Late	制御指令が到着するのが遅い	ノイズ，誤	開閉までに時間がかかる	空調制御が実行されるまでにタ	低	
インテリジェント空調アプリRTC		地窓制御指令	地窓制御RTC		地窓制	Value	制御指令が誤っている	ノイズ，誤	地窓が適切に開閉されない	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC		トップライト制	トップライト制御RTC		トップラ	Omission	制御指令が来ない	断線，セン	トップライトが開閉されない	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC		トップライト制	トップライト制御RTC		トップラ	Commission	意図しない制御指令がくる	ノイズ，誤	意図しない開閉が発生する	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC		トップライト制	トップライト制御RTC		トップラ	Early	制御指令が早く到着する	ノイズ，誤	早めに開閉される	温度変化に素早く対応できる	低	
インテリジェント空調アプリRTC		トップライト制	トップライト制御RTC		トップラ	Late	制御指令が到着するのが遅い	ノイズ，誤	開閉までに時間がかかる	空調制御が実行されるまでにタ	低	
インテリジェント空調アプリRTC		トップライト制	トップライト制御RTC		トップラ	Value	制御指令が誤っている	ノイズ，誤	トップライトが適切に開閉さ	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC	窓1	窓制御指令	窓制御アプリRTC::窓1		窓制御	Omission	制御指令が来ない	断線，セン	窓が開閉されない	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC	窓1	窓制御指令	窓制御アプリRTC::窓1		窓制御	Commission	意図しない制御指令がくる	ノイズ，誤	意図しない開閉が発生する	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC	窓1	窓制御指令	窓制御アプリRTC::窓1		窓制御	Early	制御指令が早く到着する	ノイズ，誤	早めに開閉される	温度変化に素早く対応できる	低	
インテリジェント空調アプリRTC	窓1	窓制御指令	窓制御アプリRTC::窓1		窓制御	Late	制御指令が到着するのが遅い	ノイズ，誤	開閉までに時間がかかる	空調制御が実行されるまでにタ	低	
インテリジェント空調アプリRTC	窓1	窓制御指令	窓制御アプリRTC::窓1		窓制御	Value	制御指令が誤っている	ノイズ，誤	窓が適切に開閉されない	効果的な空調を実行できない	低	
窓制御アプリRTC::窓1		窓制御指令	複合miniRTCs::窓1		窓制御	Omission	制御指令が来ない	断線，セン	窓が開閉されない	効果的な空調を実行できない	低	
窓制御アプリRTC::窓1		窓制御指令	複合miniRTCs::窓1		窓制御	Commission	意図しない制御指令がくる	ノイズ，誤	意図しない開閉が発生する	効果的な空調を実行できない	低	
窓制御アプリRTC::窓1		窓制御指令	複合miniRTCs::窓1		窓制御	Early	制御指令が早く到着する	ノイズ，誤	早めに開閉される	温度変化に素早く対応できる	低	
窓制御アプリRTC::窓1		窓制御指令	複合miniRTCs::窓1		窓制御	Late	制御指令が到着するのが遅い	ノイズ，誤	開閉までに時間がかかる	空調制御が実行されるまでにタ	低	
窓制御アプリRTC::窓1		窓制御指令	複合miniRTCs::窓1		窓制御	Value	制御指令が誤っている	ノイズ，誤	窓が適切に開閉されない	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC	窓2	窓制御指令	窓制御アプリRTC::窓2		窓制御	Omission	制御指令が来ない	断線，セン	窓が開閉されない	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC	窓2	窓制御指令	窓制御アプリRTC::窓2		窓制御	Commission	意図しない制御指令がくる	ノイズ，誤	意図しない開閉が発生する	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC	窓2	窓制御指令	窓制御アプリRTC::窓2		窓制御	Early	制御指令が早く到着する	ノイズ，誤	早めに開閉される	温度変化に素早く対応できる	低	
インテリジェント空調アプリRTC	窓2	窓制御指令	窓制御アプリRTC::窓2		窓制御	Late	制御指令が到着するのが遅い	ノイズ，誤	開閉までに時間がかかる	空調制御が実行されるまでにタ	低	
インテリジェント空調アプリRTC	窓2	窓制御指令	窓制御アプリRTC::窓2		窓制御	Value	制御指令が誤っている	ノイズ，誤	窓が適切に開閉されない	効果的な空調を実行できない	低	
窓制御アプリRTC::窓2		窓制御指令	複合miniRTCs::窓2		窓制御	Omission	制御指令が来ない	断線，セン	窓が開閉されない	効果的な空調を実行できない	低	
窓制御アプリRTC::窓2		窓制御指令	複合miniRTCs::窓2		窓制御	Commission	意図しない制御指令がくる	ノイズ，誤	意図しない開閉が発生する	効果的な空調を実行できない	低	
窓制御アプリRTC::窓2		窓制御指令	複合miniRTCs::窓2		窓制御	Early	制御指令が早く到着する	ノイズ，誤	早めに開閉される	温度変化に素早く対応できる	低	
窓制御アプリRTC::窓2		窓制御指令	複合miniRTCs::窓2		窓制御	Late	制御指令が到着するのが遅い	ノイズ，誤	開閉までに時間がかかる	空調制御が実行されるまでにタ	低	
窓制御アプリRTC::窓2		窓制御指令	複合miniRTCs::窓2		窓制御	Value	制御指令が誤っている	ノイズ，誤	窓が適切に開閉されない	効果的な空調を実行できない	低	
インテリジェント空調アプリRTC		モニタリングラ	統合管理RTC		インテ	モニタリ	Omission	状態情報が来ない	断線など	制御状態を把握できなくな	制御状態を提示できなくなる	低
インテリジェント空調アプリRTC		モニタリングラ	統合管理RTC		インテ	モニタリ	Commission	予期せぬタイミングで状態情報が	ノイズ，誤	制御状態を把握できなくな	制御状態を提示できなくなる	低
インテリジェント空調アプリRTC		モニタリングラ	統合管理RTC		インテ	モニタリ	Early	状態情報が早く到着する	ノイズ，誤	早めに制御状態が通知され	早めに制御状態が通知される	低
インテリジェント空調アプリRTC		モニタリングラ	統合管理RTC		インテ	モニタリ	Late	状態情報が到着するのが遅い	ノイズ，誤	制御状態を把握するまでに	制御状態の通知に時間がかか	低
インテリジェント空調アプリRTC		モニタリングラ	統合管理RTC		インテ	モニタリ	Value	状態情報が誤っている	ノイズ，誤	制御状態を把握できなくな	制御状態を提示できなくなる	低
地窓制御RTC		モニタリングラ	統合管理RTC		地窓	モニタリ	Omission	状態情報が来ない	断線など	地窓の状態を把握できなく	効果的な空調を実行できない	低
地窓制御RTC		モニタリングラ	統合管理RTC		地窓	モニタリ	Commission	予期せぬタイミングで状態情報が	ノイズ，誤	地窓の状態を把握できなく	効果的な空調を実行できない	低
地窓制御RTC		モニタリングラ	統合管理RTC		地窓	モニタリ	Early	状態情報が早く到着する	ノイズ，誤	早めに地窓の状態が通知さ	早めに状態が通知される	低
地窓制御RTC		モニタリングラ	統合管理RTC		地窓	モニタリ	Late	状態情報が到着するのが遅い	ノイズ，誤	地窓の状態を把握するまで	状態の通知に時間がかかる	低
地窓制御RTC		モニタリングラ	統合管理RTC		地窓	モニタリ	Value	状態情報が誤っている	ノイズ，誤	地窓の状態を把握できなく	効果的な空調を実行できない	低

トップライト制御RTC		モニタリングラ	統合管理RTC	トップ	モニタリ	Omission	状態情報が来ない	断線など	トップライトの状態を把握で	効果的な空調を実行できない	低	
トップライト制御RTC		モニタリングラ	統合管理RTC	トップ	モニタリ	Commission	予期せぬタイミングで状態情報が	ノイズ，誤	トップライトの状態を把握で	効果的な空調を実行できない	低	
トップライト制御RTC		モニタリングラ	統合管理RTC	トップ	モニタリ	Early	状態情報が早く到着する	ノイズ，誤	早めにトップライトの状態が	早めに状態が通知される	低	
トップライト制御RTC		モニタリングラ	統合管理RTC	トップ	モニタリ	Late	状態情報が到着するのが遅い	ノイズ，誤	トップライトの状態を把握す	状態の通知に時間がかかる	低	
トップライト制御RTC		モニタリングラ	統合管理RTC	トップ	モニタリ	Value	状態情報が誤っている	ノイズ，誤	トップライトの状態を把握で	効果的な空調を実行できない	低	
複合miniRTCs::窓１	窓１	窓ステータス	窓制御アプリRTC::窓１	窓１	窓ステ	Omission	状態情報が来ない	断線など	窓の状態を把握できなくな	効果的な空調を実行できない	低	
複合miniRTCs::窓１	窓１	窓ステータス	窓制御アプリRTC::窓１	窓１	窓ステ	Commission	予期せぬタイミングで状態情報が	ノイズ，誤	窓の状態を把握できなくな	効果的な空調を実行できない	低	
複合miniRTCs::窓１	窓１	窓ステータス	窓制御アプリRTC::窓１	窓１	窓ステ	Early	状態情報が早く到着する	ノイズ，誤	早めに窓の状態が通知され	早めに状態が通知される	低	
複合miniRTCs::窓１	窓１	窓ステータス	窓制御アプリRTC::窓１	窓１	窓ステ	Late	状態情報が到着するのが遅い	ノイズ，誤	窓の状態を把握するまでに	状態の通知に時間がかかる	低	
複合miniRTCs::窓１	窓１	窓ステータス	窓制御アプリRTC::窓１	窓１	窓ステ	Value	状態情報が誤っている	ノイズ，誤	窓の状態を把握できなくな	効果的な空調を実行できない	低	
窓制御アプリRTC::窓１		モニタリングラ	統合管理RTC	窓１	モニタリ	Omission	状態情報が来ない	断線など	窓の状態を把握できなくな	効果的な空調を実行できない	低	
窓制御アプリRTC::窓１		モニタリングラ	統合管理RTC	窓１	モニタリ	Commission	予期せぬタイミングで状態情報が	ノイズ，誤	窓の状態を把握できなくな	効果的な空調を実行できない	低	
窓制御アプリRTC::窓１		モニタリングラ	統合管理RTC	窓１	モニタリ	Early	状態情報が早く到着する	ノイズ，誤	早めに窓の状態が通知され	早めに状態が通知される	低	
窓制御アプリRTC::窓１		モニタリングラ	統合管理RTC	窓１	モニタリ	Late	状態情報が到着するのが遅い	ノイズ，誤	窓の状態を把握するまでに	状態の通知に時間がかかる	低	
窓制御アプリRTC::窓１		モニタリングラ	統合管理RTC	窓１	モニタリ	Value	状態情報が誤っている	ノイズ，誤	窓の状態を把握できなくな	効果的な空調を実行できない	低	
複合miniRTCs::窓２	窓２	窓ステータス	窓制御アプリRTC::窓２	窓２	窓ステ	Omission	状態情報が来ない	断線など	窓の状態を把握できなくな	効果的な空調を実行できない	低	
複合miniRTCs::窓２	窓２	窓ステータス	窓制御アプリRTC::窓２	窓２	窓ステ	Commission	予期せぬタイミングで状態情報が	ノイズ，誤	窓の状態を把握できなくな	効果的な空調を実行できない	低	
複合miniRTCs::窓２	窓２	窓ステータス	窓制御アプリRTC::窓２	窓２	窓ステ	Early	状態情報が早く到着する	ノイズ，誤	早めに窓の状態が通知され	早めに状態が通知される	低	
複合miniRTCs::窓２	窓２	窓ステータス	窓制御アプリRTC::窓２	窓２	窓ステ	Late	状態情報が到着するのが遅い	ノイズ，誤	窓の状態を把握するまでに	状態の通知に時間がかかる	低	
複合miniRTCs::窓２	窓２	窓ステータス	窓制御アプリRTC::窓２	窓２	窓ステ	Value	状態情報が誤っている	ノイズ，誤	窓の状態を把握できなくな	効果的な空調を実行できない	低	
窓制御アプリRTC::窓２	窓２	モニタリングラ	統合管理RTC	窓２	モニタリ	Omission	状態情報が来ない	断線など	窓の状態を把握できなくな	効果的な空調を実行できない	低	
窓制御アプリRTC::窓２	窓２	モニタリングラ	統合管理RTC	窓２	モニタリ	Commission	予期せぬタイミングで状態情報が	ノイズ，誤	窓の状態を把握できなくな	効果的な空調を実行できない	低	
窓制御アプリRTC::窓２	窓２	モニタリングラ	統合管理RTC	窓２	モニタリ	Early	状態情報が早く到着する	ノイズ，誤	早めに窓の状態が通知され	早めに状態が通知される	低	
窓制御アプリRTC::窓２	窓２	モニタリングラ	統合管理RTC	窓２	モニタリ	Late	状態情報が到着するのが遅い	ノイズ，誤	窓の状態を把握するまでに	状態の通知に時間がかかる	低	
窓制御アプリRTC::窓２	窓２	モニタリングラ	統合管理RTC	窓２	モニタリ	Value	状態情報が誤っている	ノイズ，誤	窓の状態を把握できなくな	効果的な空調を実行できない	低	

c-3-5　リスクアセスメント評価リスト Software HAZOP リストおよびネットワーク構造図を利用した FTA 評価

ハードウェア構造図から導出されたリスクアセスメント評価リストおよびソフトウェア構造図から導出された Software HAZOP リストからリスクの高い設備機器が抽出される。加えて、本住宅システムは、他種類の通信プロトコルを利用して、各 RT コンポーネントが通信しており、特に有線か無線かによって、その通信状態の信頼性に影響を受けることから、ネットワーク構造図を作成することが重要となる。図 c-3-8 および図 c-3-9 に住宅システムの全体のネットワーク構造図を示す。この中では、PLC、CAN、DIO、Zigbee 等の通信プロトコルが利用されており、特に Zigbee は無線通信ということで、信頼性の面でデータ通信におけるリスクが高いことになる。すなわち、リスク評価点の高い設備機器まわりで Zigbee が利用される場合は、より曝露可能性を高く見積もる必要がある。

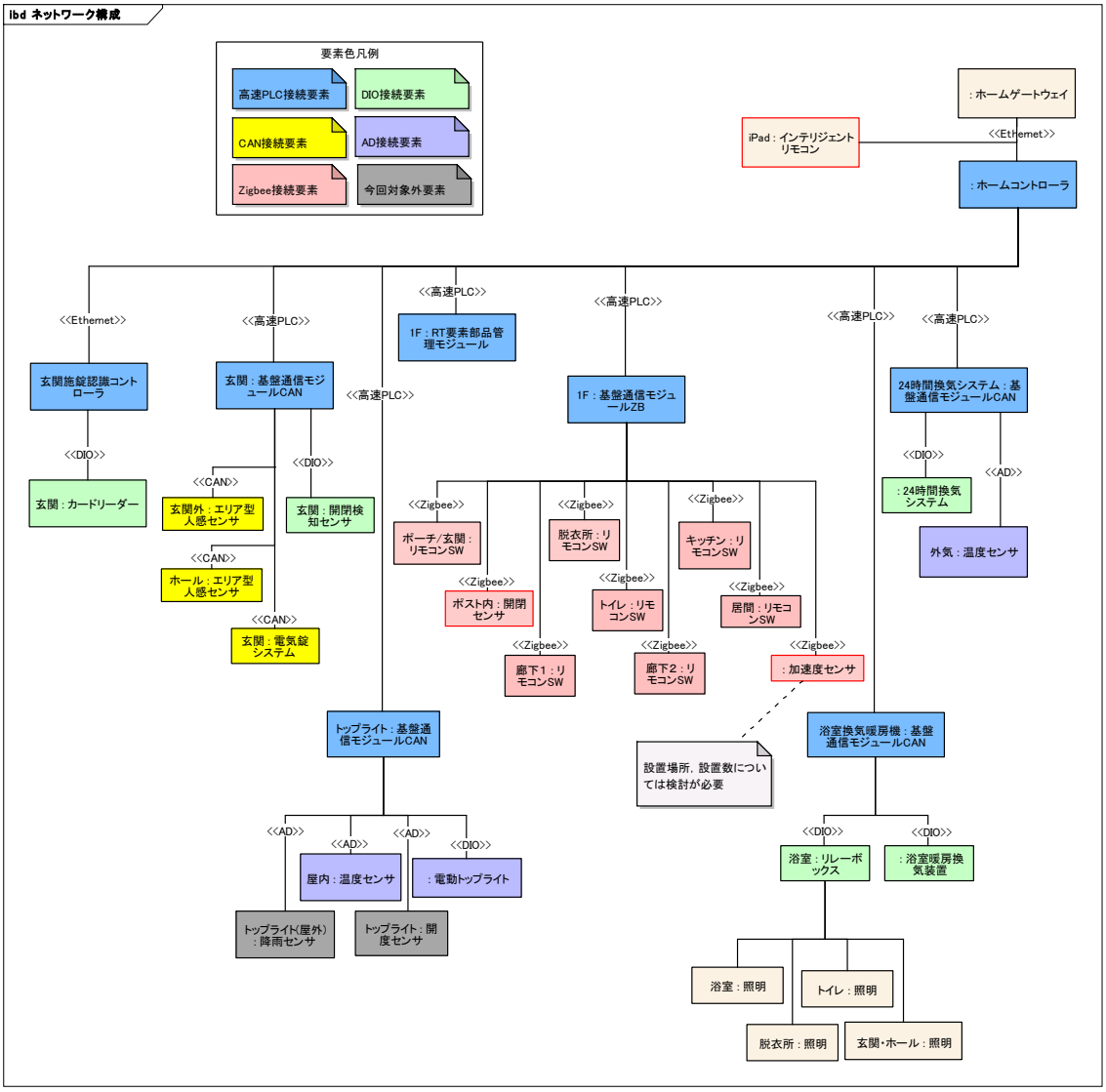


図 c-3-8　住宅システムにおけるネットワーク構造図（1）

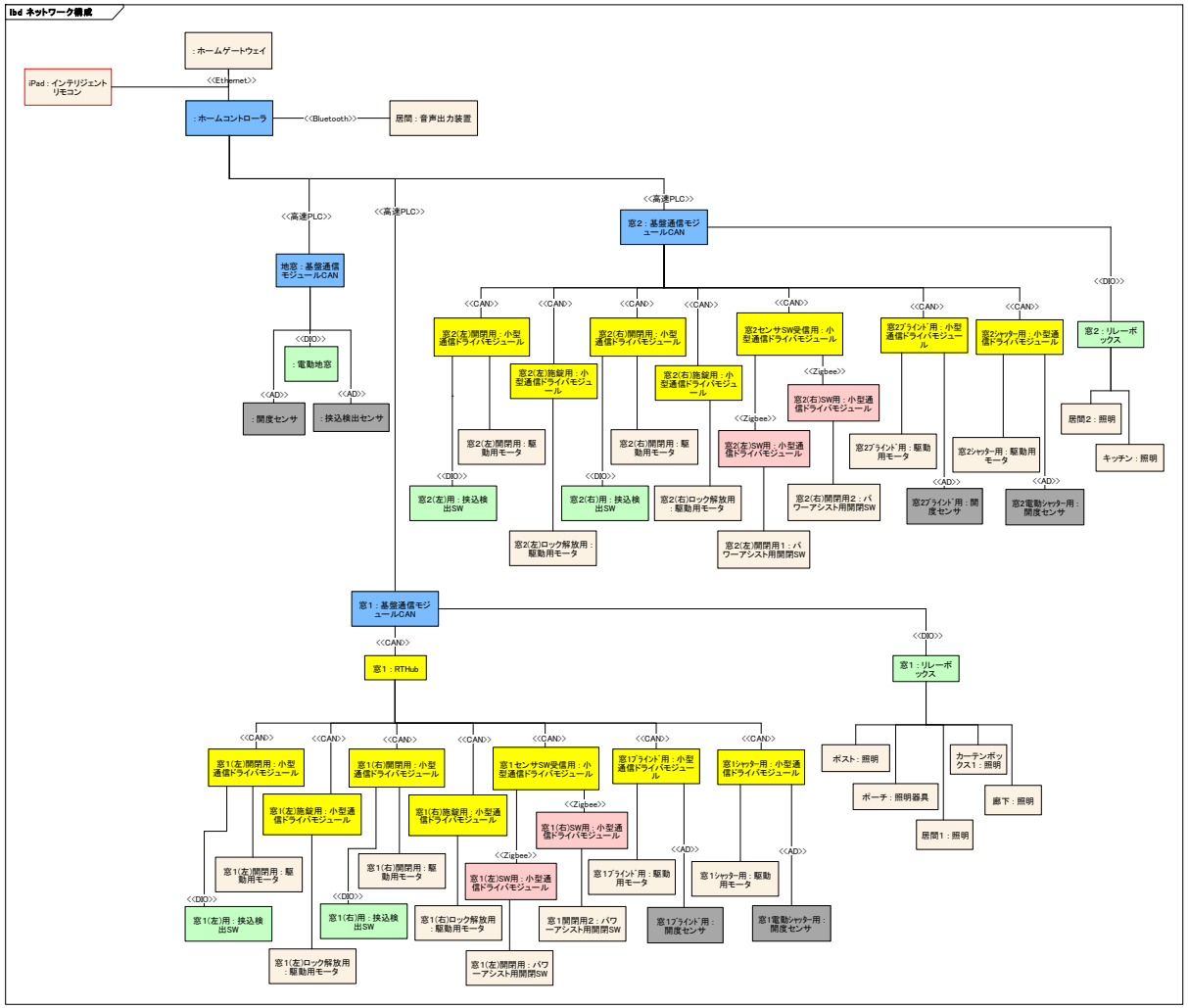


図 c-3-9　住宅システムにおけるネットワーク構造図（2）

リスクアセスメント評価リスト、Software HAZOP リストおよびネットワーク構造図から、リスクの高い設備機器を抽出し、その設備機器における Fault Tree Analysis(FTA)を評価行う。図 c-3-10 がインテリジェント窓に体が挟まるという高いリスクに対する FTA である。

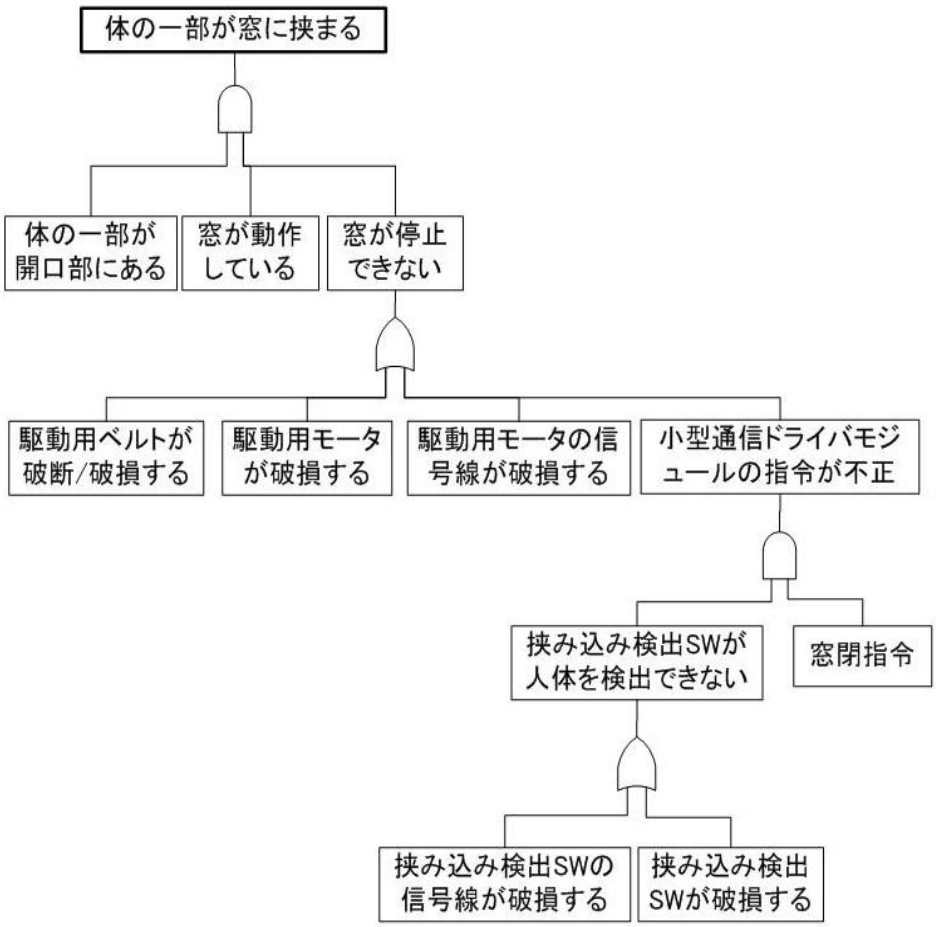


図 c-3-10 インテリジェント窓の挟み込みにおける FTA

窓に対する挟み込みに対して、その事象が起こる状態は「体の一部が開口部にある」、「窓が動作している」、「窓が停止できない」の AND で起こることになる。ここで、故障要因としては、「窓が停止できない」という事象がどの故障原因でおこるかをさらに解析する。図中ではいくつか考えられるが、特に物理的な破損等については、その故障確率を評価し、それに応じたメンテナンス期間の設定ならびに部材の交換時期を決定し、破損という事象の確率を許容できるレベルにまで下げるための方策を施す。以上の危険事象に対する FTA を行い、安全に関するセンサを付加するなど、システムの変更が生じた場合は、再度、ハードウェア構造図、ソフトウェア構造図、ならびにネットワーク構造図を修正し、リスクアセスメント評価リスト、Software HAZOP リストを作成し、すべてのリスク評価が許容できるレベルに見積もられた段階で、システムの安全性評価を終了とする。

以上が、本事業で構築した RT 要素を組み込んだ住宅システムの安全性評価手順となる。

年度毎の特許、論文、外部発表等の件数

特許、論文、外部発表等の件数（内訳）

区分 / 年度	特許出願			論文		その他外部発表（プレス発表等）
	国内	外国	PCT※出願	査読付き	その他	
H20FY	0 件	0 件	0 件	0 件	0 件	0 件
H21FY	0 件	0 件	0 件	2 件	4 件	2 件
H22FY	2 件	0 件	0 件	3 件	11 件	4 件

Ⅳ. 実用化、事業化の見通しについて

1.実用化、事業化の見通し

1-1 要素部品管理モジュールおよび基盤通信モジュール（株式会社アルゴシステム）

研究開発の成果を踏まえ、以後2年以内の実用化を目指し完成度を更に高めていく。ドキュメントを充実させ、カタログなどの販売資料を作成して販売活動を行うよう計画している。合わせて、ホームページなどに掲載して、PR する。また、マーケティング活動を強化して市場動向や分野別のセンサー等のニーズを調査して有望な分野で使用されるセンサー要素部品の開発を行い、製造・販売する計画である。更に、今回受託された企業、大学、独立行政法人等の連携で、パッケージ商品化して販売することも計画している。

1-2 RT ミドルウェア（株式会社セック）

本研究開発をおこない、住宅の RT 化のための技術開発をおこない、実用化への見通しを立てることができた。しかしながら、RT 住宅の実用化、事業化に向けては、住宅の設備機器メーカの取り込み、システムの安全性、品質の確保などの課題がある。本格的な実用化、事業化は、2014 年度を目標とし、以下の施策を進める。

○株式会社ミサワホーム総合研究所、独立行政法人産業技術総合研究所などと協力し、設備機器メーカへの RT 技術の提案、普及を図る。必要に応じて、実用化、事業化のための FS（事業化可能性調査）も視野に入れて活動する。

○滝田技研株式会社、株式会社オカテック、株式会社リバスト、旭光電機株式会社と協力し、RT ミドルウェア対応のモータドライバやセンサの事業化、普及を進める。さらに、他のセンサメーカなどにも RT ミドルウェア技術の売込みをおこない、RT 要素部品のラインナップ拡充を図る。

本プロジェクトで開発した組込み向けの軽量 RT ミドルウェアや PLC（電力線通信）技術は、一般住宅に留まらず、環境エネルギー分野などへの適用が可能であると考えており、早期の事業化、製品化の可能性を検討する。

1-3 RT ミドルウェア開発支援ツール（株式会社テクノロジックアート）

本プロジェクトにて開発した RT ミドルウェア開発支援ツール群は，大きく以下の2種類に分類することができる．

○汎用 RT ミドルウェアでも利用可能なツール
- ・プラグアンドプレイ設定ツール
- ・RT システムローダー
- ・RT 要素部品コンフィギュレーション支援ツール
- ・RT システム状態遷移定義ツール

○組み込み用 RT ミドルウェアのみで利用可能な専用ツール
- ・高速制御用(CAN 用)RTC-Lite(miniRTC)向け RT 要素部品化支援ツール
- ・低速制御用(Zigbee 用)RTC-Lite(microRTC)向け RT 要素部品化支援ツール
- ・PIC/dsPIC 用 RTC-Lite 向け RT 要素部品化支援ツール
- ・RTCHub/miniRTC/microRTC 用 RTSystemEditor

これらのツールのうち，「汎用 RT ミドルウェアでも利用可能なツール」については，弊社の既存製品である「PatternWeaver for RT-Middleware」のオプションとして，既に公開済みである．一方，「組み込み用 RT ミドルウェアのみで利用可能な専用ツール」については，ツールを利用するためには，対象となる RT ミドルウェアが必要となり，ツール単独で公開を行ってもあまり意味がない．そこで，これらのツールについては，各種ミドルウェアの開発機関と連携して，公開時期，公開方法の検討を行っていく必要がある．

公開済みのツールについては，これまでにもプロジェクト内のユーザおよび一般ユーザの意見，要望の調査を行ってきたが，更に広範，詳細な調査を行い，更なる機能強化，開発効率の向上，ユーザビリティの向上を図っていく．また，各種ツール自体の有効性をより広く一般にアピールし，ユーザ数をより一層増やしていく必要がある．そして，寄せられてきた要望を基に更にツールの機能強化を行っていく体制や，ツール自体のサポートを行う体制を早急に整備していく．

1-4　小型通信ドライバモジュールと小型リニアアクチュエータ（THK 株式会社）

（1）成果の実用化可能性

本プロジェクトで開発した小型通信ドライバモジュールは、汎用的な小型モータコントローラドライバとして使用可能である。よって、RTミドルウェアのネットワークにおけるRTコンポーネントとしての用途だけではなく、小型化・省電力化の要求が高い生産設備機器向け、または一般消費財向けの小型モータコントローラドライバとしての需要も多く見込まれる。

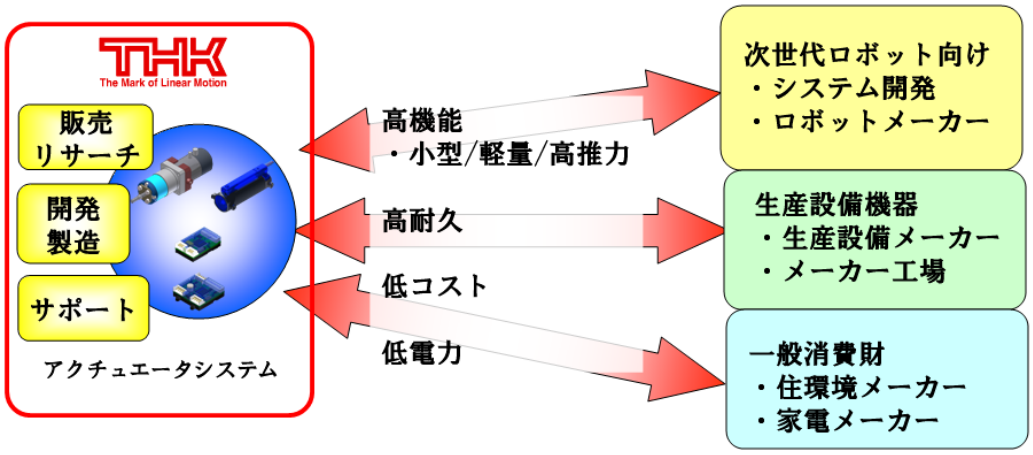


図1-4-1　需要が見込まれる用途

現在のRT市場は市場規模が小さく、RT要素部品を実用化する際の課題となる。一方で、THKの既存の販売市場である生産設備機器市場および一般消費財市場に展開すれば、スケールメリットが生まれ、安価かつ安定して供給できる可能性がある。生産設備機器市場では高耐久、低コストであることが必要不可欠であることから、RT要素部品を、高信頼かつリーズナブルな製品にブラッシュアップすることができる。

そこで、まずは生産設備機器向けの販売を目標とする。現在、特定ユーザーへのサンプル出荷を始めており、そこから評価を重ね、2011年中に標準製品としての発売を目指している。生産設備機器用途として実績と信頼を積み重ねたのち、2012年中に次世代ロボットRT用途向けの標準製品を発売する予定である。

(2)事業化までのシナリオ

前述のとおり、本プロジェクト終了後も引き続き、開発品のブラッシュアップと社内標準製品化に取り組み、まず生産設備機器向けの標準製品の発売を目指す。

既存のモータコントローラドライバが適用された生産設備システムに対し、本プロジェクトで開発したRT要素部品および小型通信ドライバモジュールで構成された設備システムは、非常にコンパクトとなる。よって今後、多くの生産設備システムでの使用されることを見込んでいる。

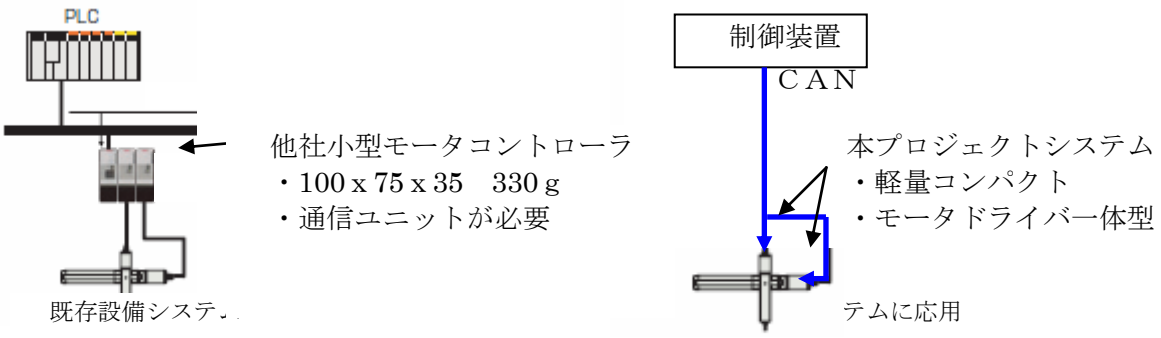


図 1-4-2　既存システムとの比較

またTHKでは、単軸ロボットの市場ですでに10%程度の市場シェアを確保しており、それらの製品に本システムを組み合わせていくことで、300億円といわれる単軸ロボットも含めた電動アクチュエータ市場において、更なるシェアの拡大を狙っている。

一方で、次世代ロボット(RT)向けアクチュエータシステムといったロボット要素部品事業を、今後のTHKにおける新規の主幹事業とすべく、本プロジェクトで開発したシステムを軸に、新規製品の開発とラインナップの拡張を行う。

次世代ロボット市場の10年後の予測として、下記の経済産業省と独立行政法人新エネルギー・産業技術総合開発機構(NEDO)によって推計されたものをベースとすると、2020年には、3兆円の市場規模がある。もし仮に、その20%がロボット向けアクチュエータシステムの市場で、うち10%を担えるとしたなら、600億円という事業規模が見込まれる。

その足がかりとして、まずは2011年中にRT開発者へサンプル出荷を行い、2012年度中に販売・サポート体制を整え、標準製品として販売を目指す。

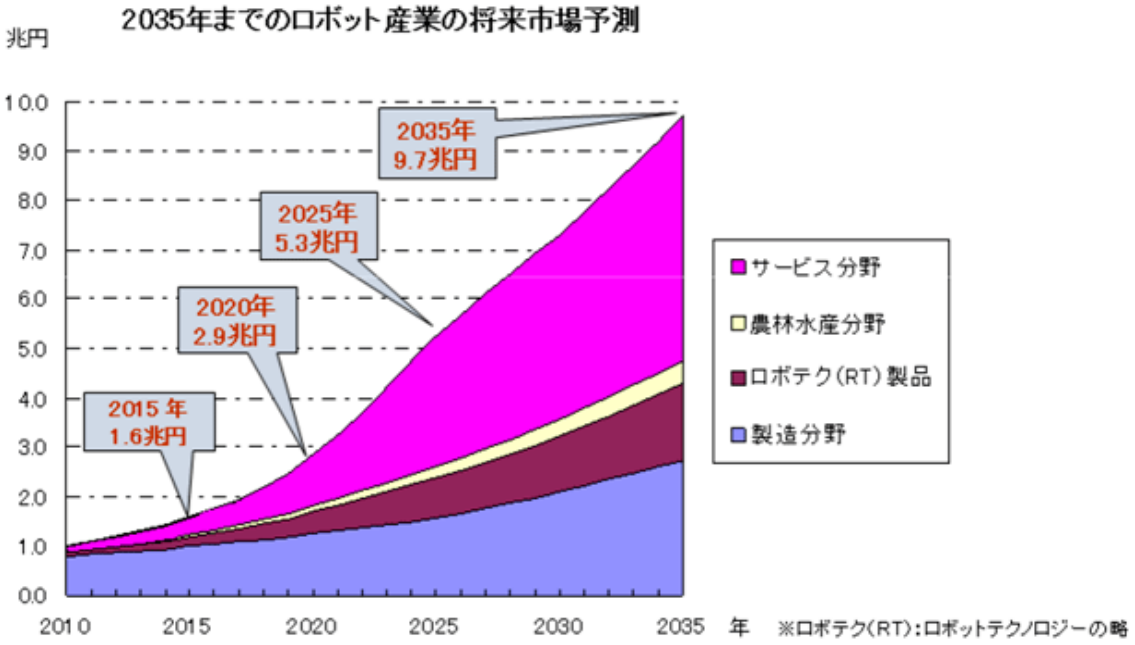


図 1-4-3 ロボット産業の将来市場予測

(3)波及効果

我が国において自動車産業の次に発展するであろう次世代ロボット市場は、労働力不足対策や犯罪、災害、医療等における将来への不安を軽減し、安心で安全な社会を実現する手段として、様々な分野で活用されることが期待されている。そのためには、RTシステムを効率的に開発し、実用化していくことが必要となる。現在でもRTシステムを簡易に構築する為のRTミドルウェアが開発されているが、現実に、RTミドルウェアに対応したアクチュエータやそれに対応するセンサ等の開発が進んでいない。

本プロジェクトにより開発したシステムでは、小型通信ドライバモジュールによってRTミドルウェアのネットワークに簡単に接続可能なRT要素部品がラインナップされており、それらを生産設備機器向けにも販売することでスケールメリットを生み、価格を下げることが可能になる。

開発したシステムを利用すれば、RTシステム開発者は、開発費の削減、開発期間の削減が可能になり、より高度なRTシステムの開発が行われるようになる。そこで、新たなRTのニーズが生まれ、新しいRT要素部品が生まれることで、RT向け製品の市場も成熟し採算が取れるようになり、またさらにコストが下がる。こうした良い循環のなかで高度に発達したRTシステムは、自動車産業のように世界をリードしていくこととなり、わが国の経済再生への足がかりとなるはずである。その循環を生むきっかけになればと期待する。

1-5 RTミドルウェアの標準化ならびに安全性の検討（独立行政法人産業技術総合研究所）

本事業において、開発された製品については、担当企業が独自のビジネスモデルに沿って実用化を進めていくことになるが、今後の活動として重要な点は、住宅システム向けのオープンな規格としてRTミドルウェアを位置づけていくことである。現在、スマートグリッドの動きがあるように、PLC や Zigbee といった通信規格の統一化を図り、事業所及び各家庭の家電を初めとした住宅設備や家電機器の消費電力情報や制御までも電力会社から操作できる仕組みが導入しつつある。すなわち、オープンな通信規格を導入することで、様々な機器がネットワークにつながり、連係動作可能な仕組みも合わせて構築できることになる。このような背景の下、今回開発された組込MPU 向け RT ミドルウェアをうまく業界に浸透させていくかの活動が、実用化に向けた大きな課題となると考えられる。今までも、情報家電機器に対してECHONETといった標準化の動きが進められているが、その規格をつくることによってビジネスの広がりを生み出すまでには至っていない。住宅だけでなく、都市といった社会システムまで広げた枠組みでの標準化をねらうためには、さらに業種を広げた活動を進めていく必要があり、特にユーザー企業からの意見を積極的に取り組み、それを各メーカーが規格化していくコンソーシアム活動を活発にしていくことが重要である。

一方、オープンな規格を進めていく上で、事業化には2つの課題が挙げられる。一つが、従来から企業がビジネスモデルとしてきた囲い込み型のビジネスモデルからの脱却である。企業のビジネスとしては、ネットワーク系について独自規格で進めており、ユーザーがある製品を購入した場合、それに関係する別の製品に関して、他社の製品を排除できるというメリットが生まれる。従来はそのビジネスモデルは有効であったが、携帯電話に見られるように、全ての機器がネットワーク化されることが前提になると、独自規格を進めることは、逆にグローバルな規格から外れ、ガラパゴス化することで、ビジネスマーケット自体も縮小してしまう状況にある。すなわち、今後、住宅システムにおいても、オープンプラットフォームのビジネスとしての有効性を各業界に宣伝していくことが重要となる。

もう一つが、責任分解点の曖昧さの問題である。独自規格で進めることは、その規格自体がメーカーの責任において決められているため、それを採用している機器もそのメーカーが責任を負えるだけの製品保証を担保できる。しかし、オープンな規格となると、他社の製品がシステム内に混在するために、住宅システムとして何らかのトラブルが発生した場合、どの製品が原因なのか、また、それに対するサポートは誰が行うのかと言った製品保証に関する責任分解点が曖昧となってしまう。何が接続されるかわからないオープンなシステムにおいて、メーカーが製品を提供するには、現状ではリスクが高いといえる。これを解決するためには、システム内のトラブルを監視する仕組みや、原因となる製品を早急に抽出する仕組み、ならびに、製品毎の認証を第三者機関等で評価できる仕組みを並行して進めることが重要であり、これは民間企業だけの活動では困難な課題である。

このように、オープンな規格を普及させていくためには、民間企業独自の活動だけでは困難であり、各企業を中立の立場からまとめる第三者的な機関が必要となる。このような活動に対して、従来から RT ミドルウェアの標準化に向けた活動を進めている産業技術総合研究所として、ロボットだけでなく、幅広い分野において RT ミドルウェアを初めとしたオープン規格の普及活動を進めていく必要があると考えている。

現在、産業技術総合研究所コンソーシアムとして、住宅・ビルといった建築関係のユーザー側である企業をまとめたコンソーシアムをプロジェクト当初から構築しており、本事業で開発された成果を関連業界に普及する活動を並行して進めていくと考えている。また、住宅やビルといった建物だけでなく、スマートグリッドのような都市といった社会システムへの応用は同じ技術で可能であることから、より関連企業の枠を広げ、オープン規格の有効性をアピールしていくことが重要である。一方、技術的に RT ミドルウェアが導入されたシステムにおいて、そのシステムの安全基準や安全評価手法、ならびに、安全を担保するツールの整備を進めていくことも重要である。

添付資料1

平成２１・０３・２７産局第２号
平 成 ２ １ 年 ４ 月 １ 日

ロボット・新機械イノベーションプログラム基本計画

１．目的

我が国の製造業を支えてきたロボット技術・機械技術を基盤とし、ＩＴ技術・知能化技術など先端的要素技術との融合を促進することにより、家庭、医療・福祉、災害対応など幅広い分野で活躍する次世代ロボットや新機械技術の開発・実用化を促進し、生産性の向上と人間生活の質の向上を実現するとともに、我が国経済社会の基盤である製造業の競争力の維持・強化を目指す。

２．政策的位置付け

○経済財政改革の基本方針２００８（２００８年６月閣議決定）

経済成長戦略の３本の柱、革新的技術創造戦略のうち、革新的技術戦略のひとつとしてロボット技術が位置付けられている。

○科学技術基本計画（２００６年３月閣議決定）

ロボット・新機械技術は、特に重点的に研究開発を推進すべき分野（重点推進４分野）の一つである情報通信分野や、推進分野であるものづくり技術分野、社会基盤分野に位置付けられている。

○「経済成長戦略大綱」（２００６年７月財政・経済一体改革会議。２００８年６月改定版を経済財政諮問会議に報告）

国際競争力の強化の取組みのうち、世界の潜在需要を喚起する新産業群の創出において、我が国が特に優位性を有し、早期にその成果を社会に還元するため、積極的な実証や環境整備を行うべき技術として、ロボットやＭＥＭＳ技術が位置付けられている。

○「新産業創造戦略」（２００５年６月経済産業省取りまとめ）

先端的新産業分野として、「ロボット」を戦略７分野の一つとして掲げ、２０１０（平成２２年）までの市場規模、その成長に向けたアクションプログラムを盛り込んでいる。当該アクションプログラムには、ユーザ（施設、地域）を巻き込んだ実証試験を中心としたモデル開発事業による先行用途開発、モデル事業と連携した重要な要素技術や共通インフラ技術の開発支援、及び人間とロボットの共存に必要な安全性の確保と、保険制度等の制度基盤の整備が提示されている。

新機械技術の重要分野であるＭＥＭＳ技術について、当該新産業群の創出を支える重点四分野（「科学技術基本計画」による）の分野間の融合による推進が指摘されている。

○「イノベーション２５」（２００７年６月閣議決定）

ロボット・新機械技術は、生涯健康な社会や多様な人生を送れる社会の実現に向けて、中長期的に取り組むべき課題として、新たな走行車等の普及促進のための環境整備、高度みまもり技術導入のためのルール作りなどの安全・安心な社会形成、また、ユビキタスネットワークや民生用ロボットの本格普及に向けた環境整備、低侵襲診断・治療技術の実現、安全・安心な社会のための将来デバイスの実現、さらに世界的課題解決に貢献する社会のための新しいものづくり技術など、今後の研究開発の進展等によって、その成果を社会に適用していく上で取組が必要であるとともに、随時見直しをし、その取組を加速・拡充していくことが必要とされている。

○「ロボット産業政策研究会」報告書（２００９年３月とりまとめ）

近い将来に、次世代ロボットが実際に役立つものになるよう、特に技術開発・事業開発、安全確保、社会ルールの整理・策定のための取組等についてまとめた。

３．達成目標

（１）我が国製造業の高度化に必要不可欠な基盤技術である機械分野においては、バイオ技術やＩＴ技術等の異分野技術を活用した従来の機械の概念を超えた新しい機械の創造及びその計測技術の確立を図ることを目標とする。例えば、２０１５年頃に革新的ＭＥＭＳの本格普及を目指すことにより、安全・安心な社会の構築に貢献する。

（２）安全・安心な社会、便利でゆとりある生活の実現のために必要不可欠なロボットは、信頼性技術、高機能化・知能化技術、システム化技術が特に重要であり、これら技術を開発することで、２０１５年頃には、自律的に多様な作業を行うロボットの実用化を目指す。

４．研究開発内容

［プロジェクト］

Ⅰ．ロボット技術開発

（１） 生活支援ロボット実用化プロジェクト（運営費交付金）

①概要

介護者支援や移動支援等の生活支援にあたっては、人との接触度が高くなるため、より一層の安全性の確保が必要。このため生活支援ロボットの対人安全性技術の開発・実証と、安全基準設定等に向けた安全性・有効性データの集約・分析を実施する。

②技術目標及び達成時期

２０１３年度までに、生活支援ロボットのリスクアセスメント手法や、安全性基準適合性評価手法・情報の蓄積提供手法を確立する。また、対人安全性に関する指標、機械・電気安全、機能安全の試験・評価方法や手順について、国際標準提案を行えるようにするとともに、実証試験を実施する。

③研究開発機関

２００９年度～２０１３年度

（２）基盤ロボット技術活用型オープンイノベーション促進プロジェクト（運営費交付金）

①概要

これまでの研究開発プロジェクトの成果を活用し、生活環境やロボットで使用される各種要素部品をＲＴ(Ｒｏｂｏｔ　Ｔｅｃｈｎｏｌｏｇｙ)システムで利用しやすい共通の接続方式、制御方式の下で利用可能な形で提供（ＲＴコンポーネント化）するための基盤を開発する。これにより既存の生活環境を簡単にＲＴシステム化し、それらを活用することにより様々な生活支援機能の提供、基盤ロボット技術の普及と標準化を推進する。

②技術目標及び達成時期

２０１０年度までに、共通の通信インタフェースとＲＴミドルウェアで動作させる基盤通信モジュール、既存の要素部品をＲＴコンポーネント化したＲＴ要素部品、それらを用いたＲＴシステムを開発する。

③研究開発期間

２００８年度～２０１０年度

（３）次世代ロボット知能化技術開発プロジェクト（運営費交付金）

①概要

生活空間や多品種少量生産の製造現場など状況が変わりやすい環境下では、ロボットの使用条件や用途は大きく限定されている。これを克服するため、ロボットが確実性（ロバスト性）をもって稼動し、ロボットの環境・状況認識能力等の向上とともに、ロボットの知能要素をモジュール化し、その蓄積管理及び組合せ等を可能とする技術を開発する。

②技術目標及び達成時期

２０１１年度までに、次世代ロボットが高度な作業（タスク）を行う上で必要な効率的で実用的な知能化技術を開発する。具体的には、魅力的でニーズが高いタスクを設定し、知能化技術モジュールを開発し、高機能なロボットシステムの構築を実証する。

③研究開発期間

２００７年度～２０１１年度

（４）戦略的先端ロボット要素技術開発プロジェクト（運営費交付金）

①概要

市場ニーズ及び技術戦略マップに基づき、約１０年後にロボット技術の活用により達成するミッションを設定した上で、これを達成するために必要なロボットシステム及び要素技術開発を、関係府省の連携の下で実施する。

②技術目標及び達成時期

市場ニーズ及び技術戦略マップに基づき、約１０年後にロボットを活用して達成するミッションを設定した上で、これを達成するために必要なロボットシステム及び要素技術の開発を実施する。具体的かつ先端的なＲＴ開発を支援することで、我が国のＲＴ競争力の維持・発展を図るとともに、研究開発成果の他分野（自動車、情報家電等）への波及を図る。

③研究開発期間

２００６年度～２０１０年度

Ⅱ．ＭＥＭＳの技術開発・新機械産業の領域開拓

（１）異分野融合型次世代デバイス製造技術開発プロジェクト（運営費交付金）

①概要

高信頼性が必要な医療分野や特殊環境等で活用され、医療や安全・安心等の社会的課題を解決する、小型・高性能・省エネルギーな次世代デバイスの基盤プロセス技術を、ＭＥＭＳ製造技術とナノ・バイオ等の異分野技術の融合により開発する。

②技術目標及び達成時期

２０１２年度までに、次世代デバイス製造に必要不可欠な基盤プロセス技術群である、バイオ・有機材料融合プロセス技術、３次元ナノ構造形成プロセス技術、マイクロ・ナノ構造大面積・連続製造プロセス技術を開発すると共に、得られた知見を系統的に蓄積しデーターベス化し、従来の技術情報と統合的に取り扱える知識データベースシステム整備を行う。

③研究開発期間

２００８年度～２０１２年度

５．政策目標の実現に向けた環境整備（成果の実用化、導入普及に向けた取組）

〔導入普及促進〕

ロボットやその関連部品等の見本市の開催等を支援することによって、システム開発者、要素部品の開発者、ロボットユーザ等の間のマッチングを図り、中小・ベンチャーや異業種企業のロボット産業への参入を促進する。

また、市場創出に貢献するロボットを表彰し、ロボットユーザ、メーカから一般の方まで広くＰＲする表彰制度「今年のロボット」大賞を共催機関と協力して実施している。

開発したソフトウェア等の成果については、広く一般に提供するなど積極的な普及を図ることにより、より多くの開発主体がロボット技術開発に参加できる環境を創出し、ロボット技術開発の裾野の拡大を図る。

将来のロボットは人に接する場面が多くなるであろう。したがって、ロボットの導入・普及を促進するためには、安全に対する考え方を整理し、周知することが重要である。平成１９年７月には人間と共存する次世代ロボットの安全性を確保するための基本的な考え方をまとめた「次世代ロボット安全性確保ガイドライン」をとりまとめた。今後は、普及や具体化に向けた取組みが求められており、技術開発と並行して安全に係るルールなどの整備を推進することで普及をより現実化させることが必要である。

ＭＥＭＳの一層の実用化促進を図るため、異分野や製造設備を有していない企業でも容易にＭＥＭＳビジネスに参入できるように、ＭＥＭＳ用設計・解析支援システムを開発した。その成果を活用しつつ、実習を中心とした人材育成及び試作環境の充実、製造拠点（ファンドリー）強化などＭＥＭＳ産業全体の競争力の維持・強化を図る。

〔標準化〕

各プロジェクトで得られた成果のうち、標準化すべきものについては適切な標準

化活動（国際規格（ＩＳＯ／ＩＥＣ）、日本工業規格（ＪＩＳ）、その他国際的に認知された標準団体（ＯＭＧ等）への提案等）を実施する。

特に、ロボットの安全基準や性能の評価基準については、過去に実施した研究開発プロジェクト等による実証データや「次世代ロボット安全性確保ガイドライン」の活用を図りつつ我が国発の国際標準としての提案について検討し、拡大するロボット市場における国際競争力の確保を目指す。

なお、これまでの研究施策の成果である、ロボット部分品の接続の共通化を目指したＲＴＭ（ロボット・テクノロジー・ミドルウェア）が、ＯＭＧ（ソフトウェア技術の国際標準化団体）において、平成１９年１２月に標準仕様として採択されている。

ＭＥＭＳ技術・製品を世界市場に広く普及するために技術戦略マップに基づくＭＥＭＳ標準化戦略の策定、国際規格案の開発、提案、推進等の標準化活動に継続的に取り組む。

６．研究開発の実施に当たっての留意事項

事業の全部又は一部について独立行政法人の運営費交付金により実施されるもの（事業名に（運営費交付金）と記載したもの）は、中期目標、中期計画等に基づき、運営費交付金の総額の範囲内で、当該独立行政法人の裁量によって実施されるものである。

７．改訂履歴

（１）平成１４年２月２８日付け、２１世紀ロボットチャレンジプログラム基本計画制定。
（２）平成１５年３月１０日付け制定。２１世紀ロボットチャレンジプログラム基本計画（平成１４・０２・２５産局第３号）は、廃止。
（３）平成１６年２月３日付け制定。２１世紀ロボットチャレンジプログラム基本計画（平成１５・０３・０７産局第１１号）は、廃止。
（４）平成１７年３月３１日付け制定。２１世紀ロボットチャレンジプログラム基本計画（平成１６・０２・０３産局第１６号）は、廃止。
（５）平成１８年３月３１日付け制定。２１世紀ロボットチャレンジプログラム基本計画（平成１７・０３・２５産局第１８号）は、廃止。
（６）平成１９年４月２日付け制定。２１世紀ロボットチャレンジプログラム基本計画（平成１８・０３・３１産局第７号）は、廃止。
（７）平成１４年２月２８日付け、新製造技術プログラム基本計画制定。
（８）平成１５年３月１０日付け制定。新製造技術プログラム基本計画（平成１４・０２・２５産局第６号）は、廃止。
（９）平成１６年２月３日付け制定。新製造技術プログラム基本計画（平成１５・０３・０７産局第９号）は、廃止。
（１０）平成１７年３月３１日付け制定。新製造技術プログラム基本計画（平成１６・０２・０３産局第１１号）は廃止。
（１１）平成１８年３月３１日付け制定。新製造技術プログラム基本計画（平成１７・０３・２５産局第５号）は、廃止。
（１２）平成１９年４月２日付け制定。新製造技術プログラム基本計画（平成１８・０３・３１産局第６号）は、廃止。

（１３）平成２０年４月１日付け、ロボット・新機械イノベーションプログラム基本計画制定。２１世紀ロボットチャレンジプログラム基本計画（平成１９・０３・１５産局第２号）及び新製造技術プログラム基本計画（平成１９・０３・１９産局第３号）は、本イノベーションプログラム基本計画に統合することとし、廃止。

（１４）平成２１年４月１日付け制定。ロボット・新機械イノベーションプログラム基本計画（平成２０・０３・２７産局第３号）は、廃止。

Ｐ０８０１４

（ロボット・新機械イノベーションプログラム）

「基盤ロボット技術活用型オープンイノベーション促進プロジェクト」基本計画

機械システム技術開発部

１．研究開発の目的・目標・内容

（１）研究開発の目的

我が国のロボット産業は、産業用ロボットの普及により製造業を中心に拡大発展し、今日、国際的にもトップレベルのロボット技術（以下、「ＲＴ」という。）を蓄積している。我が国の少子高齢化や安心・安全の問題が急速に進行しつつある中、ＲＴを製造業以外も含む様々な分野で活用することが期待されている。具体的には、労働力不足や要介護者の増加などの課題を解決するとともに、犯罪、災害や医療等における将来への不安の軽減による安心で安全な社会を実現する手段として、ＲＴを駆使して機能を実現するシステム（以下、「ＲＴシステム」という。）を効率的に開発、実用化して、様々な分野で活用することが期待されている。例えば、家庭や職場の環境内でセンサやモータなど既存の部品をネットワーク接続してＲＴシステムを構築し、状況に応じた判断によって人の活動を支援できれば生活環境をより快適かつ安全にしたり、職場の生産性を向上させることができる。

しかしながら、ＲＴシステムを組み上げるには各種部品を集めて実装し、個々のシステムに合わせた制御ソフトを開発するという難しさ・煩雑さがあり、これがＲＴ分野への新規参入の障壁となっていた。そこで独立行政法人新エネルギー・産業技術総合開発機構（以下、「ＮＥＤＯ技術開発機構」という。）ではこの障壁を解消することを目的として、ＲＴミドルウェア並びに画像認識、音声認識及び運動制御の機能を有する共通基盤モジュール（別添）などのＲＴ開発基盤の整備を進めてきた。

本プロジェクトでは、これまでの開発成果を補完するものとして、生活環境やロボットで使用される各種要素部品を、ＲＴシステムで利用しやすい共通の接続方式、制御方式のもとで利用可能な形で提供（ＲＴコンポーネント化）するための基盤を開発する。またＲＴコンポーネント化された各種要素部品を用いることで既存の生活環境を簡単にＲＴシステム化し、さまざまな生活支援機能を提供することが可能であることを示す。

本開発によってＲＴシステムの開発基盤を充実させることにより、製造分野をはじめとする一部の分野に限られているＲＴ適応分野を拡大することを本プロジェクトの第一の目的とする。さらに、ロボット分野への中小・ベンチャーや異業種を含む多様な企業や研究機関等の新規参入を促進することにより、ロボット産業の裾野拡大を図ることを第二の目的として、本プロジェクトを実施する。

なお、本プロジェクトは、経済産業省が推進する「ロボット・新機械イノベーションプログラム」並びに内閣府が推進する「社会還元加速プロジェクト」の一環として実施する。

（２）研究開発の目標（最終目標　平成２２年度）

本プロジェクトでは、生活環境やロボットに使われる既存の要素部品を、共通の通信インタフェースとＲＴミドルウェアで動作させる「基盤通信モジュール」を開発する。次に、「基盤通信モジュール」を用いることにより既存の要素部品が容易にＲＴコンポーネント化でき、ＲＴシステム内で共通して利用できることを示すとともに、それを「ＲＴ要素部品」として広く提供する。さらに「ＲＴ要素部品」を用いた「ＲＴシステム」を開発し、実証試験を行い、同システムの有効性を検証することを目標とする。

研究開発の目標の詳細については、研究開発計画（別紙）に記載のとおり。

（３）研究開発の内容

上記目標を達成するために、次の３つの研究開発項目について、別紙の研究開発計画に基づき研究開発を実施する。

［委託事業］

①基盤通信モジュール及び開発ツールの開発

②基盤通信モジュールを用いたＲＴ要素部品の開発

③ＲＴ要素部品群によるＲＴシステムの開発・実証

２．研究開発の実施方式

（１）研究開発の実施体制

本研究開発は、ＮＥＤＯ技術開発機構が、原則、本邦の企業、大学、研究組合、公益法人等の研究機関（原則、国内に研究開発拠点を有していること。）の単独ないしは複数の機関によって構成される研究体から公募によって研究開発実施者を選定後、コンソーシアム（研究共同体であって法人格である必要はない。）を構築し、委託して実施する。応募にあたり、複数の研究機関によって研究体を構成する場合は、ＲＴシステムを提案するＲＴシステム開発機関が代表機関となって、基盤通信モジュール開発機関、ＲＴ要素部品開発機関などを含む構成とする。また、本研究開発は、ＮＥＤＯ技術開発機構が指名するプロジェクトリーダの下にコンソーシアム及びその責任者を置き、各研究開発項目の達成目標を実現すべく柔軟に研究開発を実施する方式を採用する。コンソーシアム責任者は、プロジェクトの技術目標等の達成及び研究開発の進捗把握を主に担当し、適宜プロジェクトリーダに報告し、指導を受けるものとする。プロジェクトリーダは、研究開発の進捗状況を適宜 NEDO に報告するとともに、NEDO と協議しつつプロジェクトの実施体制見直し、予算配分変更及び研究者の人選と成果の評価を実施し、成果の最大化に努めるものとする。

（２）研究開発の運営管理

研究開発全体の管理・執行に責任を有するＮＥＤＯ技術開発機構は、経済産業省及びプロジェクトリーダと密接な関係を維持しつつ、プログラムの目的及び目標並びに本研究開

発の目的及び目標に照らして適切な運営管理を実施する。具体的には、必要に応じて、ＮＥＤＯ技術開発機構に設置する推進委員会等、外部有識者の意見を運営管理に反映させるほか、四半期に一回程度プロジェクトリーダ等を通じてプロジェクトの進捗について報告を受けること等を行う。

ＮＥＤＯ技術開発機構は、過去の関連プロジェクト成果を本プロジェクトで可能な限り活用することとする。また、「次世代ロボット知能化技術開発プロジェクト」など推進中の関連プロジェクトとの相互協力が円滑に行われるようプロジェクト間連携を推進する。

（３）その他

①科学技術連携施策群の次世代ロボット共通プラットフォーム技術を積極的に活用する。次世代ロボット共通プラットフォーム技術の内容については、下記を参照する。

http://www.renkei.jst.go.jp/platform/robot/index.html

②開発終了後、開発した基盤通信モジュール及びＲＴ要素部品の２年間の有償サンプル提供が可能な体制を整える。

③ＲＴコンポーネントの仕様、及びＲＴミドルウェアの実装例については、下記を参照する。

i) ＲＴコンポーネントの仕様

http://www.omg.org/technology/documents/domain_spec_catalog.htm

ii) ＲＴミドルウェアの実装例

http://www.is.aist.go.jp/rt/OpenRTM-aist/html/

３．研究開発の実施期間

本研究開発の期間は、平成２０年度から平成２２年度までの３年間とする。

４．評価に関する事項

ＮＥＤＯ技術開発機構は、技術的及び政策的観点から、研究開発の意義、目標達成度、成果の技術的意義並びに将来の産業への波及効果等について、外部有識者による研究開発の事後評価を平成２３年度に実施する。なお、評価の時期については、当該研究開発に係る技術動向、政策動向や当該研究開発の進捗状況等に応じて、前倒しする等、適宜見直すものとする。さらに、平成２２年度までの各年度中に推進委員会等で各研究開発内容を精査し、必要に応じてプロジェクトの加速・縮小・中止等見直しを迅速に行う。

５．その他の重要項目

（１）研究開発成果の取扱い

①ＮＥＤＯ技術開発機構内での成果の利用

本プロジェクト期間内であっても、ＮＥＤＯ技術開発機構が推進する他の関連プロジ

ェクトに積極的にサンプル提供する（有償提供を含む)。

②成果の普及

実施者は、得られた研究成果についてプロジェクト終了後も継続的な研究、継続的なデモシステムの展示、研究発表等を行い、システムインテグレータ、ロボット製造メーカ、機械メーカ及び部品メーカや大学・研究機関等に対して成果の普及に努めることとする。

③成果の産業化

実施者は、プロジェクト終了後２年を目途に、各実施者がプロジェクトの成果を活用して、基盤通信モジュール、ＲＴ要素部品、ＲＴシステムなどの製品化を目指すこと。

なお、プロジェクト終了後少なくとも２年間、あるいは製品化までは、開発した「基盤通信モジュール」及び「ＲＴ要素部品」を一般ユーザにサンプル提供すること（有償提供を含む)。

また、開発品の仕様は、量産時に市場での競争力があるコストを実現できるものであること。

④知的財産権の帰属

委託研究開発の成果に関わる知的財産権については、「独立行政法人新エネルギー・産業技術総合開発機構新エネルギー・産業技術業務方法書」第２７条の規定等に基づき、原則として、すべて委託先に帰属させることとする 。

（２）基本計画の変更

ＮＥＤＯ技術開発機構は、研究開発内容の妥当性を確保するため、社会・経済的状況、国内外の研究開発動向、政策動向、プログラム基本計画の変更、第三者の視点からの評価結果、研究開発費の確保状況、当該研究開発の進捗状況等を総合的に勘案し、達成目標、実施期間、研究開発体制等、基本計画の見直しを弾力的に行うものとする。

（３）根拠法

本プロジェクトは、独立行政法人新エネルギー・産業技術総合開発機構法第１５条第１項第２号に基づき実施する。

６．基本計画の改訂履歴

（１）平成２０年４月、制定。

（２）平成２０年７月、イノベーションプログラム基本計画制定により改訂。

（別紙）研究開発計画

研究開発項目①「基盤通信モジュール及び開発ツールの開発」

１．研究開発の必要性

家庭や職場の環境内でＲＴシステムを構築するには、多数のセンサやモータなどの要素部品と制御装置を接続する必要があるが、現状では要素部品が共通化・標準化されていないため煩雑でコストが高くなり、実用化が困難な状況である。したがって、ＲＴシステムを普及させるには、要素部品の共通化・標準化を図り、低コストかつ短期間でＲＴシステムを開発できるようにすることが重要である。とりわけ既存の環境に「ＲＴ要素部品」を導入することにより容易に生活環境の中でＲＴを利用できるようにすることはＲＴシステムの普及に大きく貢献する。これを実現するには、既存の要素部品に標準的なネットワークとの接続機能及びＲＴミドルウェアで動作する機能を与えてＲＴコンポーネント化する従来にないモジュールの開発が必要である。そこで本研究開発項目では、この機能を有する「基盤通信モジュール」を開発する。

また、開発する「基盤通信モジュール」を各種要素部品と組み合わせて「ＲＴ要素部品」とする際には、ＲＴ要素部品開発機関がネットワーク接続側及び要素部品接続側の各種パラメータ設定、信号処理プロセスのプログラミングなどを行う必要がある。これを容易にする「開発ツール」も併せて開発することにより、様々なタイプの「ＲＴ要素部品」の低コスト化及び開発期間の短縮が期待できる。

２．研究開発の具体的内容

既存のセンサ、モータなどの要素部品をネットワーク接続可能としＲＴコンポーネント化するための「基盤通信モジュール」及び「開発ツール」の開発を行う。

３．達成目標

（１）基本性能

以下の条件を満たす「基盤通信モジュール」及び「開発ツール」を開発する。

① ＲＴミドルウェアを実装し、研究開発項目②で開発するＲＴ要素部品がＲＴシステムから OpenRTM 仕様に基づきＲＴコンポーネントとして利用可能とする。

② 独自のネットワークを用いるのではなく、既存の標準化されたネットワークと接続可能とする。また、「基盤通信モジュール」間の通信は特別な理由がない限り、既存の標準化された方式を用いる。

③ 家庭や職場の環境内に構築するＲＴシステムで必要となる要素部品と接続可能なインタフェースを有する。このインタフェース仕様は、要素部品の使われ方を考慮して設定する。

（２）その他の性能

① 提案に基づきプロジェクト着手の際には要素部品の消費電力目標を設定し、低消費電力化を目指す。

② 種々の要素部品との組み合わせを考慮して、できる限り小型化軽量とし、サイズ（基板面積）を名刺の 1/2 以下とする。ただし、開発状況によりサイズ目標は適宜見直すものとする。なお、コネクタと電源をこのサイズに納めることは必須条件ではなく、プロジェクト終了後、無理なく実用化可能とする。

（３）開発品の提供

① 開発した「基盤通信モジュール」をＲＴ要素部品開発機関が利用できる形で提供する。

② (i)各種要素部品との接続、(ii)要素部品の制御・信号処理プログラム、(iii)ネットワークとの通信設定とその保持などを容易とする「開発ツール」を、ＲＴ要素部品開発機関が利用できる形で提供する。

③ 開発した「基盤通信モジュール」及びＲＴミドルウェアにて動作させる際に必要となる項目を記載した仕様書及び取扱説明書をＲＴ要素部品開発機関が利用できる形で提供する。

（４）有効性の検証

① 開発した「基盤通信モジュール」及び「開発ツール」が開発仕様を満たし、有効に機能することをコンソーシアムメンバ間で協力して検証する。

② ＲＴシステムの実証に際しては、ＲＴシステム開発機関と協力して、開発品が有効に機能することを検証する。

（５）特記事項

ＲＴコンポーネントは下記の OpenRTM 仕様書に準拠するものとする。

Robotic Technology Component (RTC)、1.0 adopted、OMG。

http://www.omg.org/technology/documents/domain_spec_catalog.htm

研究開発項目②「基盤通信モジュールを用いたＲＴ要素部品の開発」

１．研究開発の必要性

ＲＴを利用して生活環境をより快適かつ安全にしたり、職場の生産性を向上させるには、家庭や職場の環境の状態をモニタして、必要な処理を自動で実行できることが重要である。

具体的には、温度、湿度や照度などの物理量の値や推移をセンシングし、それに基づいて情報の発信、空調や照明等の機器のコントロール、アクチュエータによる扉やブラインドの開閉等の複合的な機能を実現することが求められる。これらを実現するには、センサやアクチュエータなどの要素部品をネットワークで接続し、ＲＴシステムとして動作させることが必要である。また、これらの機能を有するＲＴシステムを容易に開発する仕組みも必要である。しかしながら、高度な機能を有するＲＴシステムを容易に構築できる要素部品は一般に入手できる形では実用化されていない。そこで本研究開発項目では、高度な機能を容易に実現できるＲＴミドルウェアを用いて、ＲＴシステム内で共通して利用できる「ＲＴ要素部品」を開発する。これにより、用途に合わせたＲＴシステムを容易に構築できるようになる。

２．研究開発の具体的内容

研究開発項目①で開発された「基盤通信モジュール」と、既存のセンサ、モータなどの要素部品とを「開発ツール」を用いて接続し、ネットワーク接続及びシステム化を可能とする「ＲＴ要素部品」の開発を行う。「基盤通信モジュール」に代えて、「次世代ロボット共通基盤開発プロジェクト」にて開発された、「画像認識、音声認識、運動制御の機能を有する共通基盤モジュール」（別添２）から使用できるものを選定して利用しても良い。

３．達成目標

以下の条件を満たす「ＲＴ要素部品」を開発する。

（１）基本性能

① 「基盤通信モジュール」、又は「共通基盤モジュール」と組み合わされている。これらは要素部品と一体化されていることが望ましいが、処理部として分離されても良い。

② ＲＴシステムから OpenRTM 仕様に基づきＲＴコンポーネントとして利用できる。

（２）開発品の提供

開発したＲＴ要素部品及びＲＴミドルウェアにて動作させる際に必要となる項目を記載した仕様書及び取扱説明書を、ＲＴシステム開発機関が利用できる形で提供する。

（３）有効性の検証

① 開発した「ＲＴ要素部品」が開発仕様を満たし、有効に機能することをコンソーシ

アムメンバ間で協力して検証する。

② ＲＴシステムの実証に際しては、ＲＴシステム開発機関と協力して、開発品が有効に機能することを検証する。

（５）特記事項

ＲＴコンポーネントは下記の OpenRTM 仕様書に準拠するものとする。

Robotic Technology Component (RTC)、1.0 adopted、OMG。

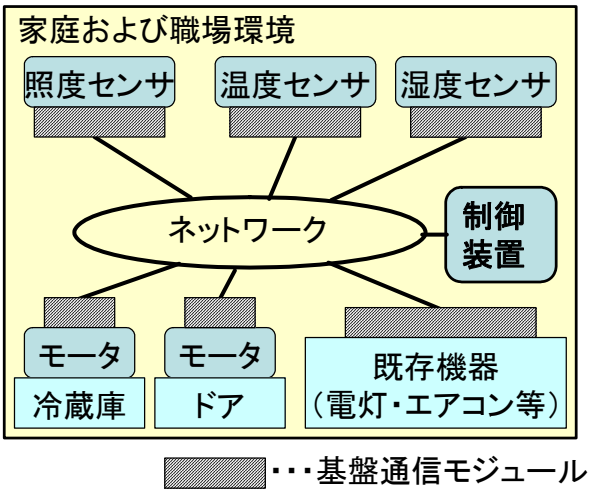
http://www.omg.org/technology/documents/domain_spec_catalog.htm

研究開発項目③「ＲＴ要素部品群によるＲＴシステムの開発・実証」

１．研究開発の必要性

生活環境をより快適かつ安全にしたり、職場の生産性を向上させるには、実際に家庭や職場の環境の状態をモニタして、必要な処理を自動で実行できるＲＴシステムを構築して、その機能を検証することが重要である（イメージ図参照）。また、これらの機能を有するＲＴシステムを容易に開発する仕組みも必要である。しかしながら、高度な機能を有するＲＴシステムは一般に入手できる形では実用化されていない。

そのため、ＲＴシステムの開発と、それを用いた実証システムの構築と、その有効性を検証することが必要である。



実証システム例のイメージ図

２．研究開発の具体的内容

研究開発項目①及び②で開発されたＲＴ要素部品群を用いてＲＴシステムを開発し、その実証システムを家庭や職場を模擬した環境内に構築して有効性の検証を行う。

３．達成目標

以下の条件を満たすＲＴシステムを開発して有効性を検証するための実証を行い、実用レベルを達成する。実証に際しては、プロジェクト期間内に、必要に応じて第三者に対してデモンストレーションする。

（１）基本性能

① ＲＴ要素部品を組み合わせたＲＴシステムとする。ＲＴシステム化の際に必要となる制御装置も開発する。

② OpenRTM 仕様に基づくＲＴシステムとして開発する。

③ システムインテグレータ（本プロジェクトではＲＴシステム開発機関がこれに相当）

や、ある程度のスキルを持ったエンドユーザが容易にシステムの構築、変更ができるシステム構築・運用ツールを開発する。

④　実証システムは実証目的に合わせて、要素部品又はＲＴシステムの入れ替えが可能である。

⑤　実証システムは、要素部品としてセンサとアクチュエータとを含める構成とする。

（２）その他の性能

家庭や職場の環境内で構築されるＲＴシステムは、バッテリや省電力での駆動が想定されるので以下の条件を満たす。

①　システム全体の省電力を意識した構成とする。

②　家庭や職場の環境で想定される温度、湿度、照度などの諸条件のもとで正常に動作する。

（３）有効性の検証

①　実証システムや小規模な個別デモシステムなどを用いて、推進委員会等によるＲＴシステム及び実証システムの有用性評価を適時実施する。

②　実証で得られた結果や知見を基盤通信モジュール開発機関及びＲＴ要素部品開発機関にフィードバックし、必要に応じて改良を促す。

（別添）共通基盤モジュール（次世代ロボット共通基盤開発プロジェクト　平成１７年度～平成１９年度）

Ⅰ．画像認識用デバイス及びモジュールの仕様

１．基本性能

生活空間等の実環境で稼働するロボットのステレオカメラの画像を処理し、ロボットの自己位置同定、環境の３次元マップ取得をリアルタイムで実行するために以下の性能を備える。

・２系統以上のカメラ画像をフレームレート30fps以上、16ビット以上のカラー解像度で同時入力・処理可能。
・カメラ画像の入力と画像処理を毎フレーム実行可能。
・移動しながら自己位置同定と環境の 3 次元マップの取得を行うための処理能力としてシーン内の 1000 箇所以上の特徴的な領域（8×8 画素以上）について、ステレオ計測と動き計測を 100ms 以下で実行可能。
・２m先の対象物を10cm以下の精度で検出可能。
・各計測データについての信頼性評価値の出力が可能。

２．ＲＴコンポーネントとしての動作

開発したモジュールを以下の RTM（ＲＴミドルウェア）の仕様に基づくＲＴコンポーネントとして提供。

Robotic Technology Component (RTC)、1.0 adopted、OMG.
http://www.omg.org/technology/documents/domain_spec_catalog.htm

３．消費電力

ピーク動作に必要な消費電力が 20W 以下。

４．基板サイズ・質量

基板面積150cm^{2}以下、質量250g以下。

５．耐ノイズ性

強電系と共存して安定に動作する。

６．付加的機能

（１）照明条件への適応やノイズ除去のための画像前処理機能として、階調補正及びフィルタリング処理の適用が可能。
（２）人物の検出及び顔の登録・照合を行うことが可能。
（３）人のジェスチャを認識する機能を有する。
（４）部屋内を移動することにより、部屋の3次元マップを構築可能。
（５）部屋のマップと現在のセンサ入力情報から、自己位置を同定可能。

Ⅱ．音声認識用デバイス及びモジュールの仕様

１．基本性能

ロボットが稼働する生活空間等の実環境で音声情報を処理し、人間とのコミュニケーションを行うために以下の性能を備える。

・様々な処理の搭載・入れ替え、性能の改善、個別ロボット向けのカスタマイズが可能。
・不特定話者の単語認識が可能。
・日常生活空間の雑音環境下で耐雑音処理により 70%以上の単語認識率を実現可能。
・音源方向の検出が可能。
・8ch 以上の多チャンネル入力が可能。

２．ＲＴコンポーネントとしての動作

開発したモジュールを以下の RTM（ＲＴミドルウェア）の仕様に基づくＲＴコンポーネントとして提供。

Robotic Technology Component (RTC)、1.0 adopted、OMG.
http://www.omg.org/technology/documents/domain_spec_catalog.htm

３．消費電力

ピーク動作に必要な消費電力が 20W 以下。

４．基板サイズ・質量

基板面積150cm^{2}以下、質量250g以下。

５．耐ノイズ性

強電系と共存して安定に動作する。

６．付加的機能

（１）ロボット発話やメカノイズをキャンセルできる（雑音発生時の認識率 70%以上）。
（２）自由発話の大語彙音声認識が可能。
（３）認識すべき音声以外の音に対する誤認識は30%以下。
（４）発話者の口とマイクの距離が 50cm以上でも性能が達成可能。

Ⅲ．運動制御用デバイス及びモジュールの仕様

１．基本性能

実運用環境下で動作する多自由度ロボットの分散処理を可能とする高度な処理機能を実現するために以下の性能を備える。

・１軸以上のアクチュエータを制御可能。

・多自由度協調動作を行うための制御情報、状態量等を出力できる。

・1ms 以下の周期処理が実現可能。

・実時間通信インタフェースを備える。

・Linux にて稼働する。

２．ＲＴコンポーネントとしての動作

開発したモジュールを以下の RTM（ＲＴミドルウェア）の仕様に基づくＲＴコンポーネントとして提供。

Robotic Technology Component (RTC)、1.0 adopted、OMG.

http://www.omg.org/technology/documents/domain_spec_catalog.htm

３．消費電力・耐熱性

・制御部が必要とする消費電力が最大で 15W 以下。

・アクチュエータ等の発熱源近傍で安定に動作する。

４．耐ノイズ性

強電系と共存して安定に動作する。

５．基板サイズ・質量

パワー部を除いた要素モジュールは面積 50cm^{2}以下、質量 150g 以下。

６．付加的機能

加速度センサ、ジャイロ、力センサやレーザレーダ等のセンサからの信号を入力し、その信号を処理可能。

ロボット分野

我が国の社会が直面する課題の解決に向け、ロボットに対する期待は大きい。しかしながら、現状では市場に投入されている大半のロボットは産業用ロボットであり、いわゆるサービスロボットの市場は確立したとは言えず、実用化の事例も少ないのが現状である。我が国の社会ニーズへの対応及び我が国ロボット産業の競争力強化という観点から、今後、次世代ロボット市場を創出、拡大していくことは重要であると言える。

他方でロボットは、機械技術、エレクトロニクス技術、材料技術、情報通信技術等、幅広い技術の統合システムであり、また、技術も市場も十分に成熟していない現時点では、個々の製品ごとに技術の擦り合わせを要する典型的な垂直連携型産業である。したがって、ロボット産業が発展するためには、我が国の「高度部材産業集積」は大きな強みとなる。また、その発展は、中堅・中小企業などの裾野産業に対して大きな波及効果をもたらすことが期待される。

今後は、実際の生活空間でロボットが活躍することを前提に、開発されたロボット技術の他分野への応用、ロボットの高機能化など、一層のレベルアップを図る必要がある。また、それと並行して事業化への支援、安全性確保等の制度整備、標準化等の環境整備のための議論が必要である。

ロボット分野の導入シナリオでは、上記議論をはじめ、経済産業省の研究開発プロジェクト等を明示し、次世代ロボット普及への道筋を示している。また、技術マップでは、ロボット分野の技術の特徴を捉え、様々なニーズのロボットに必要となる機能や環境を抽出しその関係を整理するほか、技術ロードマップにおいて、その実現時期を示している。

ロボット分野の技術戦略マップ

Ⅰ．導入シナリオ

（1）ロボット分野の目標と将来実現する社会像

少子高齢化への対応、労働力人口の減少、安全・安心な社会の実現、便利でゆとりある生活の実現のために、ロボットが生活、公共の場でより身近な存在として役立つことが期待されている。我が国では、1980 年代以降、自動車や電機・電子産業等のユーザ産業の成長や人手不足を背景に、産業用ロボットの本格的な導入が進んだ。ただし、1990 年代以降、産業用ロボットの市場規模は緩やかな成長にとどまり、用途も特定の産業分野に限られていた【参考資料 図 1：世界の産業用ロボット稼働台数、図 2：日本の産業用ロボットの出荷額及び輸出割合の推移】。また、先に挙げたような非産業用の次世代ロボット、いわゆるサービスロボットの市場は確立したとは言えず、実用化の事例も少ないのが現状である。

しかし、ロボットを巡る状況は着実に変わりつつある。製造業においては、ロボット･セルのように、さらに高度化した産業用ロボットが生産現場に投入されつつある。また、サービス業の分野においても、2005 年に開催された愛知万博では、サービスロボットの実用化に向けた実証実験が行われたのを始め、実際のビジネスにおいても、清掃ロボットや食事支援ロボット、災害復旧作業を行う遠隔操作型ロボット等の導入が進んでいる。このように、我が国のロボット産業・技術は、次の成長段階に踏みだしているところである。

他方、アジア諸国等の台頭を背景とした国際競争が激化しており、我が国ロボット産業の競争力強化や、あるいは地震や水害等の大規模災害に対する不安といった社会的課題の解決・社会ニーズへの対応に、我が国に蓄積された基盤的なロボット技術：ロボットテクノロジー（以下、ロボテク（RT））を活用・高度化することで取組むとともに、次世代ロボット市場を創出、拡大していくことは重要である。現状、状況が変わりやすい環境下ではロボットの認識能力や自律的な判断能力が低いため、使用条件や用途が限定的となっている。それに対し、ロボットの環境・状況認識能力や自律的な判断能力及び作業の遂行能力を向上させ、生活空間等の状況が変わりやすい環境下においてもロボットが確実性をもって稼動する技術（＝知能化技術）も上記解決に求められる最重要な技術課題の一つと言える。

ロボットやロボテクの概念及び市場の捉え方は複雑である【参考資料 図 3：ロボット及びロボテク（RT）の概念整理、図 4：ロボット及びロボテク（RT）製品の市場の捉え方】が､ロボテクとしての他分野への活用拡大は重要な視点である。また、今後のロボットの導入に際しては、研究開発のみならず、安全課題への取り組み、ビジネス振興のための制度整備、標準化の推進、事業化支援なども重要である。我が国製造業を支えてきたロボット技術を基盤とし、知能化技術など先端的要素技術との融合を促進

することにより、家庭、医療、福祉、災害対応など幅広い分野で活躍する次世代ロボットの開発・実用化を促進することが望まれている。

さらに、ロボット単体での開発、実用化だけではなく、社会システムの中でどのように個々のロボットやロボットの要素技術、さらには統合システムの機能が位置づけられるかを、アプリケーションあるいはニーズの中で具体的に踏まえて開発するかという視点も市場を創出していく上で重要である。今後、我が国が直面するとみられる少子高齢化、労働力不足等の課題に、ロボテクが活躍し、より良い社会に貢献していくことが求められる【参考資料 図5:(福祉介護分野)脳卒中患者の機能回復リハビリロボット(人の支援のためのロボテク(RT))、図 6:(農業分野)次世代型施設生産システム(生産技術のためのロボテク(RT))】。

(2)研究開発の取組

研究開発の推進については、各種ロボットの普及を始めとしたロボット産業の裾野拡大に向け、産学官・農工商連携などを通じて総合的に行うことが重要である。これまで、10 年後の市場ニーズ及び社会ニーズから導かれたミッションを設定し、これを達成するために必要なロボットシステム等の開発を行うもの(戦略的先端ロボット要素技術開発プロジェクト)、生活空間など状況が変わりやすい環境下においても、ロボットが確実性(ロバスト性)をもって稼働できるようにするため、自律的に活躍するロボット開発の効率化を向上する再利用可能な知能化モジュール(ソフトウェア)の研究開発(次世代ロボット知能化技術開発プロジェクト)等の推進や、過去に行った国の研究開発プロジェクトの成果等を活用したRT要素デバイス開発や活用事例の創出(基盤ロボット技術活用型オープンイノベーション促進プロジェクト)を実施している。また、介護者支援や移動支援等の人との接触度が高い生活支援ロボットの実用化のため、対人安全性基準及び基準適合性評価手法を確立し、さらに安全性基準の国際標準化を目指した研究開発も開始された(生活支援ロボット実用化プロジェクト)。

(3)関連施策の取組

〔起業・事業支援〕

ロボットやその関連部品等の見本市の開催等を支援することによって、システム開発者、要素部品の開発者、ロボットユーザ等の間のマッチングを図り、中小・ベンチャーや異業種企業のロボット産業への参入を促進する。また、市場創出に貢献するロボットを表彰し、ロボットユーザ、メーカから一般の方まで広くPRする表彰制度「今年のロボット」大賞を共催機関と協力して実施している。

〔実用化促進〕

開発したソフトウェア等の成果については、広く一般に提供するなど積極的な普及を図ることにより、より多くの開発主体がロボット技術開発に参加できる環境を創出し、ロボット技術開発の裾野の拡大を図る。

また、ロボット産業政策研究会報告書(2009 年 3 月とりまとめ)において、近い将

来に次世代ロボットが実際に役立つものになるよう、特に技術開発・事業開発、安全確保、社会ルールの整理・策定のための取組等についてまとめた。

〔ガイドライン整備〕

将来のロボットは人に接する場面が多くなると想定される。したがって、ロボットの導入と普及を促進するためには、安全に対する考え方を整理し、周知することが重要である。2007 年 7 月には人間と共存する次世代ロボットの安全性を確保するための基本的な考え方をまとめた「次世代ロボット安全性確保ガイドライン」をとりまとめた。今後は、普及や具体化に向けた取組が求められており、技術開発と並行して安全に係るルールなどの整備を推進することで普及をより現実化させることが必要である。具体的には、安全設計ガイドライン及び運用ガイドラインの策定を関係機関と協調して進めることとする。

〔規制・制度改革〕

これまでロボットの導入が想定されていなかった分野においては、現行の各種規制や制度との不整合によって導入が制約される可能性がある。そこではじめに、サービスロボットと連動したエレベータの検査項目について、導入実績がある“エレベータに人と同乗しないロボット”というケースについての業界自主検査基準作りの支援を開始した。この結果、2007 年 11 月に業界自主基準として、ロボットビジネス推進協議会において「サービスロボットの運用が可能なエレベータの検査運用指針」が策定された（検査内容等について規定）。今後、サービスロボットを導入する際に、以下に述べる法規による規制が問題となることが予想される。まず第一に、シニアもしくは障害者向け電動車いす等の移動ロボットが道路交通法の制約を受けることが予想される。第二に、身体リハビリを支援する運動補助スーツや、介護福祉士等による介護作業を支援するパワースーツや、手術支援ロボット等が医師法や医療法の制約を受けることが予想される。第三に、全般として電波法の制約が考えられる。さらに、家庭用ロボットのメーカに対しては、製造物責任問題がロボット普及を阻害しない様に PL 法に対するリスクヘッジ（例えば、免責の特例措置）が重要になる。

今後も、研究開発プロジェクトなどを通じた規制・制度改革課題の抽出を図っていく。

〔基準・標準化〕

各プロジェクトで得られた成果のうち、標準化すべきものについては適切な標準化活動（国際規格（ISO/IEC）、日本工業規格（JIS）、その他国際的に認知された標準団体（OMG 等）への提案等）を実施する。

特に、ロボットの安全基準や性能の評価基準については、サービスロボットの標準形態を考慮した上で上記ガイドラインを国際標準となるべく提案する。各種サービスロボットに共通の安全性技術要件や安全性能の評価手法及び評価基準については、過去に実施した研究開発プロジェクト等による実証データの活用を図りつつ、我が国発

の国際標準として早急に提案することを検討し、拡大するロボット市場における国際競争力の確保を目指す。

なお、これまでの研究施策の成果である、ロボット部分品の接続の共通化を目指した RTM（ロボット・テクノロジー・ミドルウェア）が、OMG（ソフトウェア技術の国際標準化団体）において、2007 年 12 月に標準仕様として採択されている。

〔広報･啓発〕

市場創出に貢献するロボットを表彰し、ロボットユーザ、メーカから一般の方まで広く PR する表彰制度「今年のロボット」大賞を共催機関と協力して実施している。（再掲）

（４）海外での取組

欧州では、特定の技術テーマに関する研究開発を促進させるための組織である欧州テクノロジー・プラットフォーム（ETP：European Technology Platform）のもと、欧州ロボット工学プラットフォーム（EUROP：European Robotics Platform）による産官学共同の研究開発体制のベース作り、標準化への取組を加速させている。

また、米国では、大手ソフトウェア会社がロボットソフトウェアの開発プラットフォームの開発・販売を進めている。Willow Garage 社では研究、開発を進める中で、各国共通となるロボット向けのオープンソースソフトウェアの発展とパーソナルロボット向けのオープンソースコミュニティを支援している。このように、システムインテグレーション技術や人材育成、知的基盤整備としてのロボット用オープンソースソフトウェア、ミドルウェアの集積も重要視されている。

欧州の例、米国の例のいずれも、今後、ロボットが市場に投入されることを見越して、標準化、共通化のイニシアティブをとろうとする動きがあることを示唆している。

（５）民間での取組

サービスロボットはこれまでエンターテインメント中心であったが、実用的なサービスロボットの本格参入を打ち出す企業も出るなど、人の役に立つサービスロボットの実用化が視野に入ってきた。

産業・研究分野の壁を越えて、事業者・研究者・技術者・政策決定者の連携と相互理解を強め、実社会で活躍するロボテク（RT）の開発と、これを活用したソリューションビジネスの開拓を促進することにより、ロボテク開発の成果を社会に還元し、もって、豊かな生活とよりよい社会の実現を目指すことを目的として、2006 年 12 月、「ロボットビジネス推進協議会」が発足した。同協議会では、ビジネスマッチング活動などの RT ソリューションビジネスの事業化支援や、共通規格検討及び RT ミドルウェアの普及促進等の活動を行っている。ロボットビジネス推進協議会は、2009 年に組織改正がなされ、部会を 3 つに再編し、その下に安全普及、移動型ロボット、医療福祉、エレベータ、通信、RT ミドルウェア、保険構築等々の 11 の WG・委員会を設置し、具体的なミッション指向の活動を行うことを主眼に事業を実施している。

（６）改訂のポイント

- 今回は改訂を行っていない。

Ⅱ．技術マップ

（１）技術マップ

導入シナリオを踏まえ、市場ニーズ、社会ニーズ（それに対応した製品イメージ）を実現する上で必要となる技術課題を抽出・俯瞰した技術マップを作成した。

① 第一に、ロボットの種類として、少子・高齢化、労働力人口の減少、安心・安全な社会の実現、便利でゆとりある生活の実現のために、ロボットの活躍が期待される 3 分野「次世代産業用ロボット」、「サービスロボット」、「フィールドロボット」を設定し、その主な用途毎に分類した。例えば「次世代産業用ロボット」では、組立てロボットと搬送ロボットに大きく分けて、“セル生産対応”など、今後求められる機能を示している。

② 第二に、ロボット技術の特徴を踏まえ、各々のロボットに求められるニーズやタスクが異なる場合でも、分野を超えて共通に求められる機能を目的・必要機能として整理した。これらロボットのすべての機能を 7 項目（環境構造化・標準化、コミュニケーション、マニピュレーション、移動、エネルギー源・パワーマネージメント、安全技術、運用技術）に整理している。

③ 第三に、これらの目的・必要機能を実現するための技術を大きく 8 項目（システム化技術、環境構造化、認識処理、センシング、制御、機構、アクチュエータ、標準化）に分類するとともに、今後各技術分類の中で求められる要素技術を抽出した。

④ さらに、現在の技術動向から必要な要素技術の不足がないかを確認した。

また、技術マップに示されている技術体系を読み解く一助として、シーズである要素技術を組み合わせると、どのようにニーズである人への貢献を果たすことができるのか、例示を試みた【添付資料 図 1：生活支援ロボット、図 2：農業用ロボット】。2 つの応用事例で共通するのは、ロボットは様々な要素技術の集合体であるということである。また、同時に、これらの要素技術を高度化させて組み合わせることにより、将来様々なシーンで人の役に立つロボットが登場しうることを示している。

なお、添付資料 図 1､2 に示されている絵は、あくまでも機能や技術を集約して表現した一例であり、ロボットや機能の形状を意図的に示すものではない。

（２）重要技術の考え方

抽出された重要技術は､それぞれにロボット分野の将来発展に大きな役割を果たすものと予想されている。その場合、重要性の軸としていくつかのものが考えられている。それをここでは、以下の 7 つの観点で評価を行った。

①「日本の技術競争力優位」

現在、研究レベル及び産業レベルでの技術力が優位な分野で、産業上、将来国際

的な優位性を実現するために重要と考えられる技術。

②「共通基盤性」

ロボット開発の時間短縮、低コスト化、ロボットの利便性向上、適用範囲拡大等の観点で、ロボット用のカスタマイズ化や標準化が共通基盤的に整備されることが必要となると考えられる技術・方式。

③「ブレークスルー技術」

次世代のロボット開発にあたり、必要度が高く、技術的難易度が高いと考えられる技術。

④「市場のインパクト」

次世代のロボット分野や自動車等の他分野への波及効果が大きく、市場インパクトが大きいと考えられる技術。

⑤「基礎技術の開発が必要」

技術としては重要であるが、直ちに商品・市場化には結びつかない学術的研究段階にある技術。

⑥「安全・安心の確保のために必要」

ロボットを活用する上でユーザや受益者の安全・安心を確保するために必要な技術。

⑦「標準化の検討が望まれる技術」

今後、ロボット技術が普及する上で標準化の検討が望まれる技術、又は我が国がイニシアティブをとる上で標準化の検討が望まれる技術。

（３）改訂のポイント

- 委員会での検討に基づき、F＜安全技術＞の項目に、「ソフトウェアによる機能安全技術」と「環境支援による安全システム技術」を追記した。

Ⅲ．技術ロードマップ

（１）技術ロードマップ

技術マップにおいて分類した 3 分野のロボット「次世代産業用ロボット」、「サービスロボット」、「フィールドロボット」について、10 年後以降の各ロボットの将来像（ミッション）を想定し、要求される仕様や必要な技術開発等を技術マップに示された重要技術から抽出し、時間軸に展開した。なお、必要に応じて分野毎に詳細な設定を追加した。また、3 分野のロボットに共通する技術や普及パターン等を共通コンセプトとしてまとめた。

（２）改訂のポイント

- 委員会及び WG での検討に基づき、産業用ロボット分野におけるキーワードとサービスロボット分野における農業、医療、福祉介護分野のロボットを追加した。

Ⅳ．その他の改訂のポイント

○ アカデミック・ロードマップとの連携とアカデミアからの提言について

- 2006 年度に、日本ロボット学会、人工知能学会、日本人間工学会の 3 学会が協力して、長期のロボットに関する学術的な研究領域・方向性を探索することを目的とするアカデミック・ロードマップを策定した。今回、技術戦略マップの改訂にあたり、アカデミアから提言を伺ったところ、その一部は次のとおり。

・ ロボテク（RT）の要素技術は広範にある。各種産業へのロボテク（RT）の貢献も目に見えない形で進んでいるとみられるが、現在は測定する手段がない。寄与を測定することにより社会への貢献度が把握できる。

・ 産業用ロボットは、成熟産業としてみられることもあるが、研究題材も多く残されている。今後は、産業界から大学等研究機関へ課題としていること等の要望を伝え、若手研究者に取り組んでもらうような方策も望まれる。

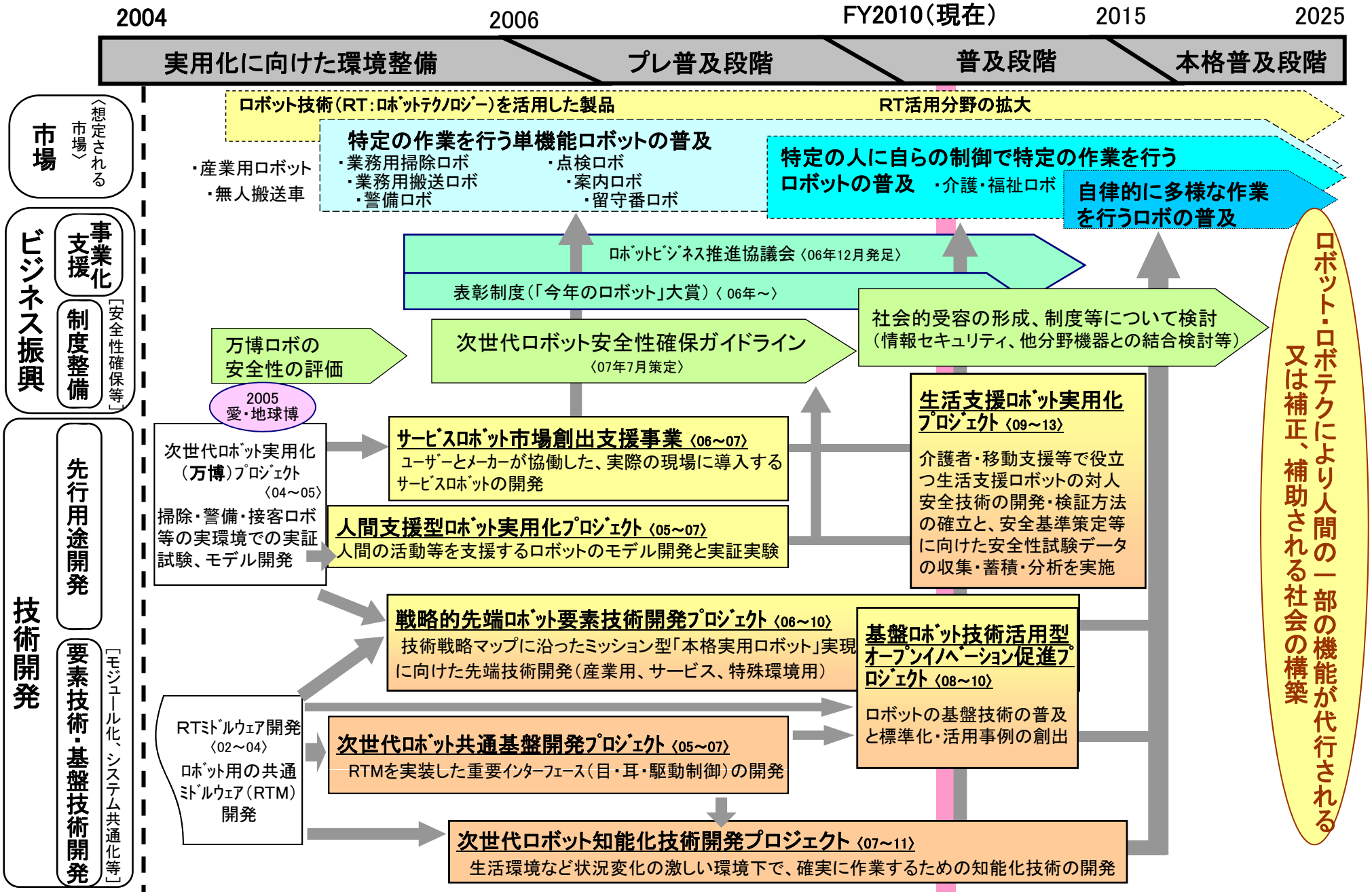
・ アカデミアとしては、力制御の本格的な実用化に代表されるような、理論体系の構築からはじまり、技術を成熟させていくことが必要である。

- 引き続きアカデミック・ロードマップとの連携を図りつつ、これら提言を受け、研究開発プロジェクトの企画・立案、実施等を進めていく。

○ロボット産業の将来市場の予測について

- 導入シナリオと並行して検討したロボット産業の将来市場の予測を示した【図 7：ロボット産業の市場推移の予測】。社会構造の変化、ロボットの潜在的な導入ニーズを踏まえ、ロボット市場創出に向けて産学官が総合的かつ戦略的な取組を行うことで、ロボット産業の将来市場は、2035 年には最大 9.7 兆円へ拡大することが期待される。

ロボット分野の導入シナリオ



ロボット分野の技術マップと重要技術

（注）重点化の評価：
1.日本の技術競争力優位
2.共通基盤性
3.ブレークスルー技術
4.市場のインパクト
5.基礎技術の開発が必要
6.安全・安心の確保のために必要
7.標準化の検討が望まれる技術

ロボットの種類

【次世代産業用ロボット分野】

組立てロボット *①
A B C F
セル生産対応
人との共同作業
工具操作
簡単教示
柔軟物ハンドリング

搬送ロボット *①
A C D E F
セル間を移動
セルへの部品供給
セルからの製品回収

【サービスロボット分野】

搬送，案内ロボット
警備（安全・安心），お供（見守り）ロボット
清掃ロボット *②
A B D E F G
サービスプロバイダ経由の個人サービス

メディアサービスロボット *②
A B D E F G
情報支援
エンタテインメント
教育支援

日常生活支援ロボット
A B C D E F G
対個人サービス

介護・福祉従事者支援ロボット
A B C D E F G
移動・移乗支援
検査支援
リハビリ支援

【フィールドロボット分野】

建設ロボット *③
A B C D E F
土木建築施工
構造物解体，廃棄物処理
構造物組立
無人化施工

水中作業ロボット *③
A C D E F
環境計測
漁業資源保全

防災ロボット *③
A B C D E F
情報収集
救出支援
被害軽減

プラント保全ロボット *③
A B C D E F
点検作業
補修作業

農業用ロボット *③
A C D E F G
農作業支援

目的・必要機能

A＜環境構造化・標準化＞
・ロボット用コンテンツサービス
・他のRT機器と通信できる
・情報家電と通信できる
・他のロボット要素と互換性がとれる
・迅速な開発ができる
・再利用性を高める
・他標準規格と連携する（ex.医療情報交換規約）
・施工情報連携（設計、施工対象、施工結果など） *③
・施工工程間の施工情報交換 *③

B＜コミュニケーション＞
・話者の方向を向く
・対話できる
・ジェスチャを理解できる
・データベース情報を提供できる
・人の状況が理解できる
・人の意図が理解できる
・人について学習し、適応できる
・人にとって好ましいインターフェイス
・メディアとして働く *③
・オペレータ操作の補助、補完 *③
・オペレータ操作への情報提示 *③
・作業対象物の状況提示（視覚、力覚・・・） *③
・複雑な作業装置（アーム等）の簡便な操作系 *③
・タスク的な作業指示 *③

C＜マニピュレーション＞
・複数のアーム等でいろいろな形状のものを掴める *③
・安全な軽量化 *③
・組み立て分解作業ができる。 *③
・道具を使って作業ができる。 *③
・多様な形状のものを迅速かつ高精度・高信頼度でハンドリングできる *③
・人間の動作をスケールアップした作業装置（大きさ、力・・・） *③
・重い（大きい）対象物を安全に思い通りにハンドリング *③
・土などのように性質の変化にも安定した掘削に思い通りにハンドリング *③

D＜移動＞
・障害物の識別
・オープンエリアでの測位
・人の動きの検出
・動的歩行・走行・跳躍動作
・環境認識と把握
・衝突の回避
・行動の学習と計画
・自分の位置が解る
・必要に応じて高精度で停止できる *③
・ラフロード・ラフロード，瓦礫上での安定な姿勢での作業、瓦礫上での効率よい移動 *③

E＜エネルギー源・パワーマネジメント＞
・長寿命・省電力 *③
・電源コードが不要 *③
・重量物可搬なアクチュエータ
・重量物可搬な動力系

F＜安全技術＞
・衝突や挟圧時の本質安全技術
・健全性診断、及び修復技術
・人検知機能に基づく自律回避と制動
・リスク評価を反映した安全方策
・安全防護階層構造の適用
・事故事例DB
・確実なインタロック制御
・ソフトウェアによる機能安全技術
・環境支援による安全システム技術

G＜運用技術＞
・リスクアセスメントに基づく運用
・マンマシンIF（安全情報提示）
・保守点検計画の策定

技術分類

＜システム化技術＞
A,B,C,D,E,F,G *②
・総合デザイン技術
・インテグレーション技術
（耐環境性、小型軽量） *③
・サービス科学 *②
A
・RTプロセッサ
・RTミドルウェア

＜環境構造化＞
A,G *②
・ユビキタスセンサ
・個人対応サービス
・ロボット同士の連携
・機器シンプル化
・外部情報連携（施工情報） *③
・移動体高速通信インフラ *③
・アドホック通信とUWB通信インフラ *③

＜認識処理＞
B
・音声処理、対話処理
・ジェスチャ、姿勢認識
・状況・意図推定/理解
・学習/適応技術
・作業対象物状態認識 *③
・作業指示理解 *③
・最適情報提示 *③
・対話、動作の学習・生成
・脳活動状態センシング

＜センシング＞
B
・話者方向センサ
・ビジョンセンサ
C
・触覚センサ
・ビジョンセンサ
大型構造物姿勢位置センシング *③
作業対象性質（土質）センシング *③
センサの小型化 *③
D
・ビジョンセンサ
・測位センサ
・環境認識センサ
・挙動検出センサ

＜制御＞
C
・マニピュレータ制御
・大型重量マニプレータ制御 *③
・作業計画 *③
D
・経路計画
・自律移動制御
・全天候自律移動
多数ロボットの協調制御
・人とロボットのハイブリッド制御
E,G
・自己エネルギー管理
F
・安全予測制御
・接触安全制御
・機能安全技術
・環境安全技術

＜機構＞
C
・アーム
・ハンド
D
・2足～多足
・脚車輪
・不整地，段差，狭隘地，狭窄空間でのモビリティ *③ *③

＜アクチュエータ＞
A,C,D,E,F
・ロボット適合ｱｸﾁｭｴｰﾀ
・過負荷適合制御
・重量物可搬アクチュエータ、動力系 *③ *③

＜標準化＞
A
・要素互換性
・標準規格互換
E
・エネルギー供給
F,G
・事故原因解析

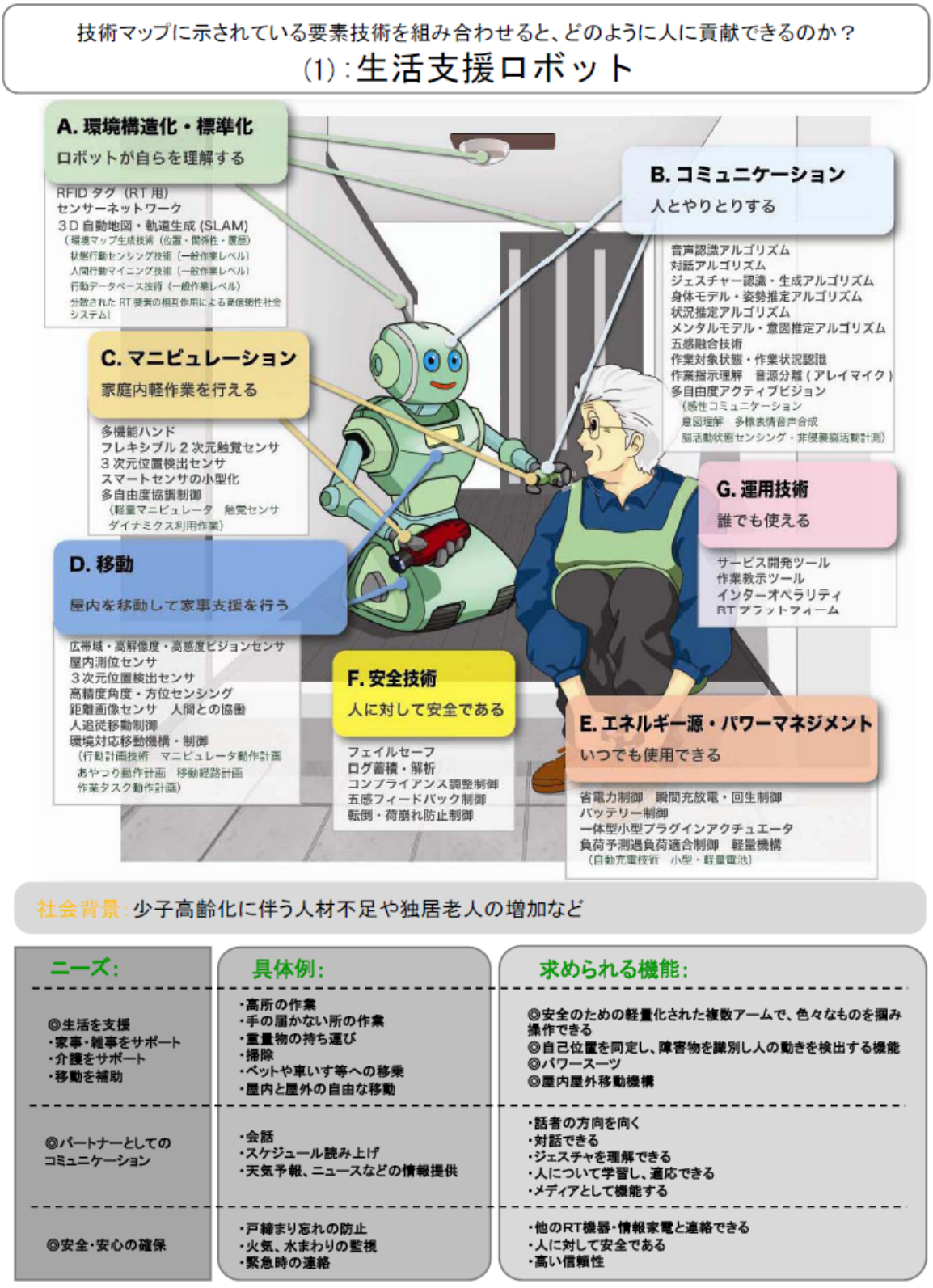
要素技術と重点化の評価

要素技術	1	2	3	4	5	6	7	
サービス開発ツール		○	○	○				
作業教示ツール		○	○	○				*①
モデリング・分析		○						
インターオペラビリティ		○		○				
開発環境・ツール		○	○	○			○	
RTミドルウェア組込デバイス	○	○					○	
ハンドリングデータベース利用技術		○			○		○	*①
RFIDタグ（RT用）	○	○		○			○	*②
センサーネットワーク（短期設置・高速通信）		○		○			○	*③
ユビキタスセンサ融合	○				○		○	
自動地図・軌道生成					○		○	
3D自動地図・軌道生成（SLAM）					○		○	*②,*③
CAD情報利用目標生成		○					○	*③
音声認識アルゴリズム		○	○	○				
対話アルゴリズム		○			○			
ジェスチャ認識・生成アルゴリズム				○	○			
身体モデル・姿勢推定アルゴリズム				○	○			
状況推定アルゴリズム				○	○			
メンタルモデル・意図推定アルゴリズム				○	○			
五感融合技術								*②
熟練者スキルの学習アルゴリズム		○			○			*③
作業対象状態・作業状況認識			○	○				*③
作業指示理解			○	○				*③
視覚・力覚・触覚・聴覚融合型情報提示			○	○				*③
屋内GPS、アドホック設置GPS		○	○	○				*③
表現動作生成		○						
筋電位計測技術		○	○		○	○		
音源分離（アレイマイク）			○	○				
多自由度アクティブビジョン					○			
フレキシブル2次元触覚センサ	○				○	○		
3次元位置検出センサ					○			
高精度角度・方位センシング		○	○			○		*③
土質センシング（リアルタイム）				○	○			*③
スマートセンサの小型化						○		
広帯域・高解像度・高感度ビジョンセンサ	○		○	○				*②
感度自律調整ビジョンセンサ				○	○			
高速画像処理アルゴリズム・チップ	○		○	○				*②
距離画像センサ	○		○	○		○		
屋内測位センサ（スードライト、ビーコンなど）		○		○				
足場センシング・認識				○	○			*③
材質センシング・認識					○			*③
多自由度協調制御	○	○		○				*②
力制御				○		○		*③
大型重量物ハンドリング制御		○	○					*③
作業計画（簡単指示で掘削ー積込等）			○	○				*③
テレオペレーション		○		○				*③
人間との協働		○		○		○		*③
環境対応移動機構・制御		○		○		○		*②
人追従移動制御	○				○			
省電力制御		○	○	○		○		*②
瞬間充放電・回生制御	○	○				○		*②
バッテリー制御		○				○		*②
コンプライアンス調整制御			○			○		
五感フィードバック制御					○	○		*②
階層・重畳制御					○	○		
転倒・荷崩れ防止制御			○	○		○		*③
軽量マニピュレータ	○	○	○	○		○		
テザーマニピュレーション		○	○	○				*③
多機能ハンド	○		○					
教示用ハンド		○						*①
小型独立関節機構					○			
不整地・段差・狭隘地・狭窄空間等でのモビリティ		○	○	○				*③
冗長可変剛性機構（人の関節）					○			
一体型小型プラグインアクチュエータ	○	○		○				
負荷予測過負荷適合制御					○			
大容量アクチュエータ（電動・油圧動力系）		○	○	○				*③
コンポーネント		○		○			○	
プロファイリング		○		○	○		○	
技術要素流出監視システム		○		○	○		○	*①
RTプラットフォーム		○	○	○			○	
無線技術（超音波通信、アドホック通信を含む）		○					○	*③
単純動作利用方法蓄積技術			○		○		○	*①
ネットプラグイン（ex.UPnP拡張）		○		○			○	
スーパーキャパシタ	○		○	○				
小型・軽量電池			○	○	○			*③
燃料電池	○			○	○			
軽量機構		○			○	○		
フェイルセーフ		○				○		
ログ蓄積・解析		○	○	○	○	○	○	
性能評価法・評価基準・オントロジー		○				○	○	*③

*①：特に「次世代産業用ロボット分野」に必要な項目
*②：特に「サービスロボット分野」に必要な項目
*③：特に「フィールドロボット分野」に必要な項目
本技術マップは平成17年度検討項目を中心に作成している。

添付資料

図１：生活支援ロボット



社会背景：少子高齢化に伴う人材不足や独居老人の増加など

ニーズ：	具体例：	求められる機能：
◎生活を支援 ・家事・雑事をサポート ・介護をサポート ・移動を補助	・高所の作業 ・手の届かない所の作業 ・重量物の持ち運び ・掃除 ・ベットや車いす等への移乗 ・屋内と屋外の自由な移動	◎安全のための軽量化された複数アームで、色々なものを掴み操作できる ◎自己位置を同定し、障害物を識別し人の動きを検出する機能 ◎パワースーツ ◎屋内屋外移動機構
◎パートナーとしてのコミュニケーション	・会話 ・スケジュール読み上げ ・天気予報、ニュースなどの情報提供	・話者の方向を向く ・対話できる ・ジェスチャを理解できる ・人について学習し、適応できる ・メディアとして機能する
◎安全・安心の確保	・戸締まり忘れの防止 ・火気、水まわりの監視 ・緊急時の連絡	・他のRT機器・情報家電と連絡できる ・人に対して安全である ・高い信頼性

（つづき）

ニーズ：	具体例：	求められる機能：
◎安全・安心の確保 （続き）	◎生活支援ロボットによる事故の防止 ・生活支援ロボット安全認証の仕組み ・生活支援ロボットの利用者相談窓口	◎生活支援ロボットのリスクアセスメント（ISO14121） ・危険源を特定 ・危険源のリスク見積もり ・リスク評価（受入可否を判断） ◎リスク対策（ISO12100－1，－2） ・本質的安全設計、安全防護と付加保護方策、使用上の情報 ◎生活支援ロボットの認証機関 ・生活支援ロボットの安全性認証手順 ◎生活支援ロボットの試験機関 ・生活支援ロボットの安全性試験方法 ◎法規制検討委員会 ・政府各府省が管轄する法規制への対応を検討するしくみ ・ノウハウのデータベース化

イメージ

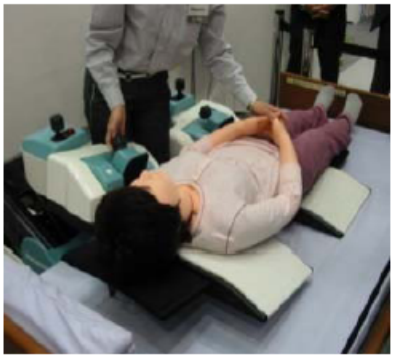
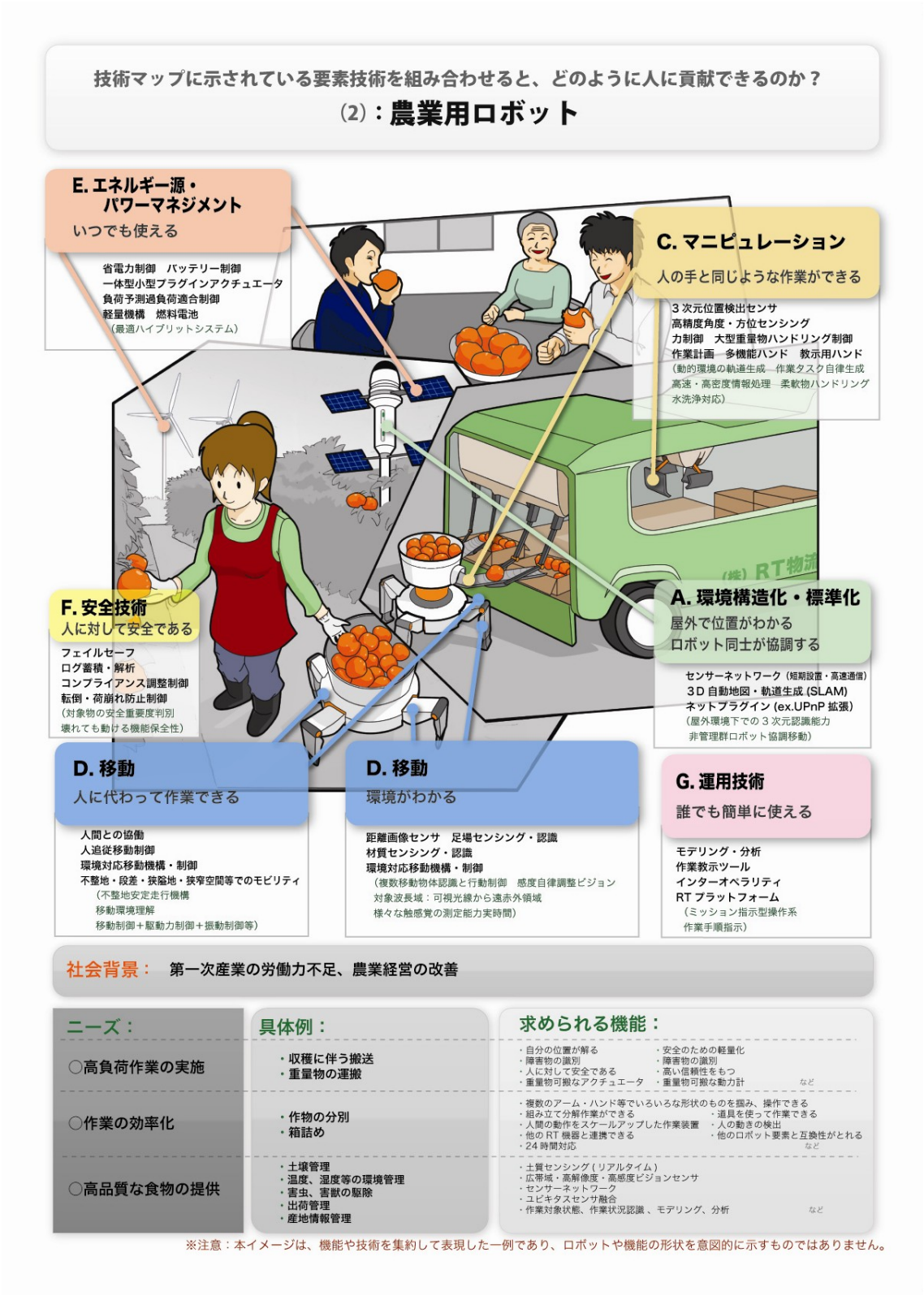
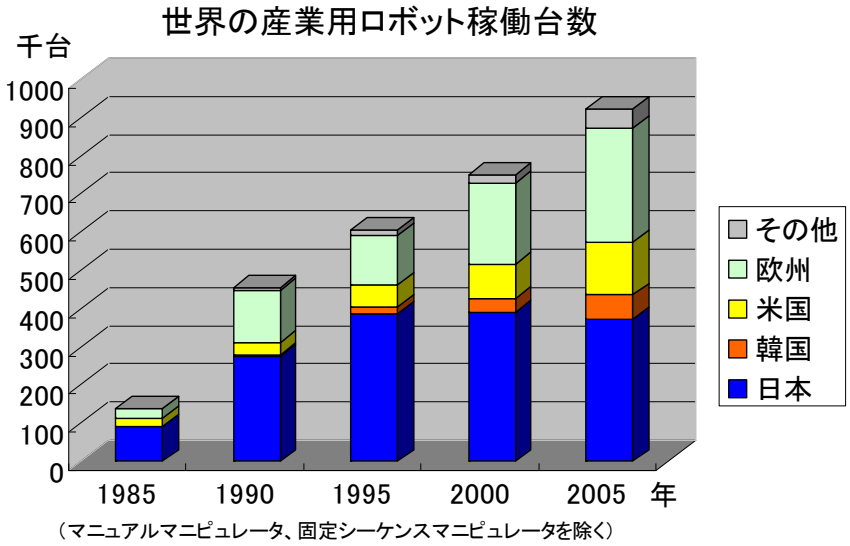
作業支援	重作業補助
	
移動支援	ベット移乗支援
	

図２：農業用ロボット


技術マップに示されている要素技術を組み合わせると、どのように人に貢献できるのか？
(2)：農業用ロボット
E. エネルギー源・
パワーマネジメント
いつでも使える
省電力制御　バッテリー制御
一体型小型プラグインアクチュエータ
負荷予測過負荷適合制御
軽量機構　燃料電池
（最適ハイブリットシステム）
C. マニピュレーション
人の手と同じような作業ができる
３次元位置検出センサ
高精度角度・方位センシング
力制御　大型重量物ハンドリング制御
作業計画　多機能ハンド　教示用ハンド
（動的環境の軌道生成　作業タスク自律生成
高速・高密度情報処理　柔軟物ハンドリング
水洗浄対応）
F. 安全技術
人に対して安全である
フェイルセーフ
ログ蓄積・解析
コンプライアンス調整制御
転倒・荷崩れ防止制御
（対象物の安全重要度判別
壊れても動ける機能保全性）
A. 環境構造化・標準化
屋外で位置がわかる
ロボット同士が協調する
センサーネットワーク（短期設置・高速通信）
３D 自動地図・軌道生成 (SLAM)
ネットプラグイン (ex.UPnP 拡張)
（屋外環境下での３次元認識能力
非管理群ロボット協調移動）
D. 移動
人に代わって作業できる
人間との協働
人追従移動制御
環境対応移動機構・制御
不整地・段差・狭隘地・狭窄空間等でのモビリティ
（不整地安定走行機構
移動環境理解
移動制御＋駆動力制御＋振動制御等）
D. 移動
環境がわかる
距離画像センサ　足場センシング・認識
材質センシング・認識
環境対応移動機構・制御
（複数移動物体認識と行動制御　感度自律調整ビジョン
対象波長域：可視光線から遠赤外領域
様々な触感覚の測定能力実時間）
G. 運用技術
誰でも簡単に使える
モデリング・分析
作業教示ツール
インターオペラリティ
RT プラットフォーム
（ミッション指示型操作系
作業手順指示）
社会背景：　第一次産業の労働力不足、農業経営の改善
ニーズ：
具体例：
求められる機能：
○高負荷作業の実施
・収穫に伴う搬送
・重量物の運搬
・自分の位置が解る
・障害物の識別
・人に対して安全である
・重量物可搬なアクチュエータ
・安全のための軽量化
・障害物の識別
・高い信頼性をもつ
・重量物可搬な動力計
など
○作業の効率化
・作物の分別
・箱詰め
・複数のアーム・ハンド等でいろいろな形状のものを掴み、操作できる
・組み立て分解作業ができる
・人間の動作をスケールアップした作業装置
・他の RT 機器と連携できる
・24 時間対応
・道具を使って作業できる
・人の動きの検出
・他のロボット要素と互換性がとれる
など
○高品質な食物の提供
・土壌管理
・温度、湿度等の環境管理
・害虫、害獣の駆除
・出荷管理
・産地情報管理
・土質センシング（リアルタイム）
・広帯域・高解像度・高感度ビジョンセンサ
・センサーネットワーク
・ユビキタスセンサ融合
・作業対象状態、作業状況認識 、モデリング、分析
など
※注意：本イメージは、機能や技術を集約して表現した一例であり、ロボットや機能の形状を意図的に示すものではありません。

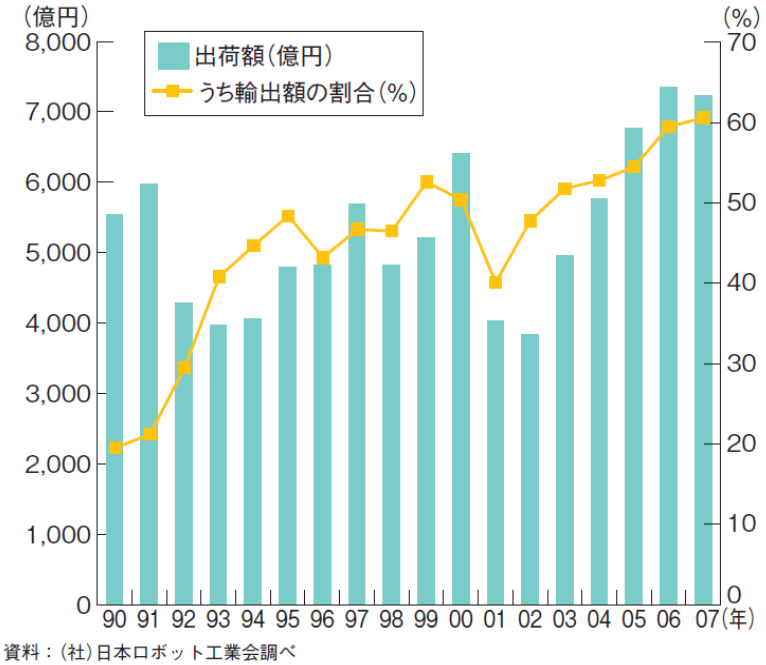
参考資料



（マニュアルマニピュレータ、固定シーケンスマニピュレータを除く）

社団法人日本ロボット工業会HPより作成

図１　世界の産業用ロボット稼動台数



資料：(社)日本ロボット工業会調べ

ものづくり白書２００９より引用

図２　日本の産業用ロボットの出荷額及び輸出割合の推移

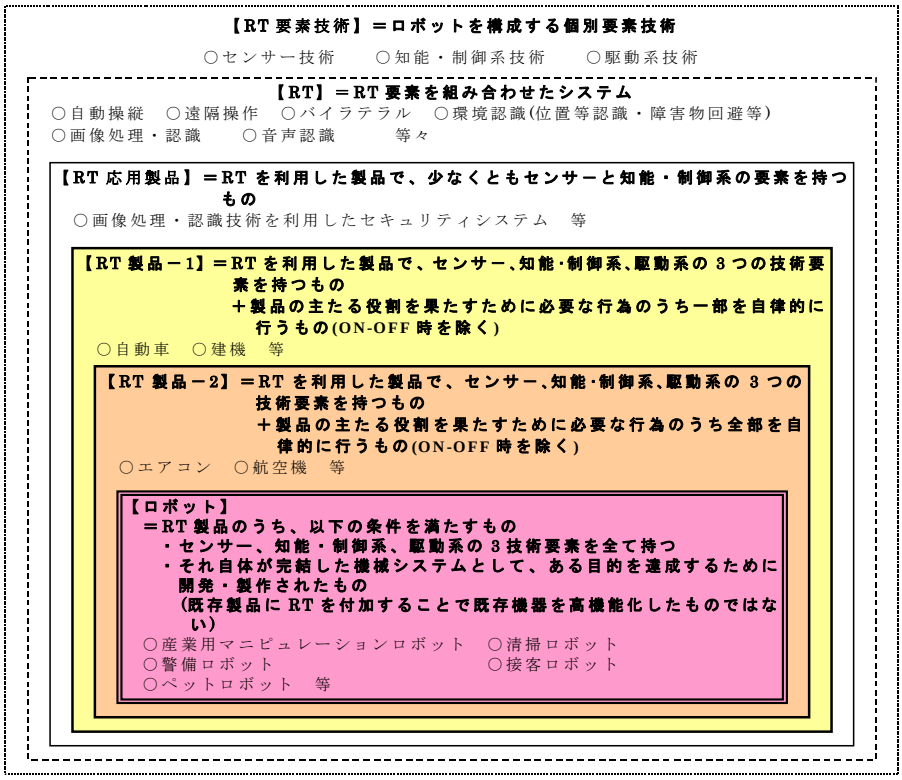


図３　ロボット及びロボテク（RT）の概念整理

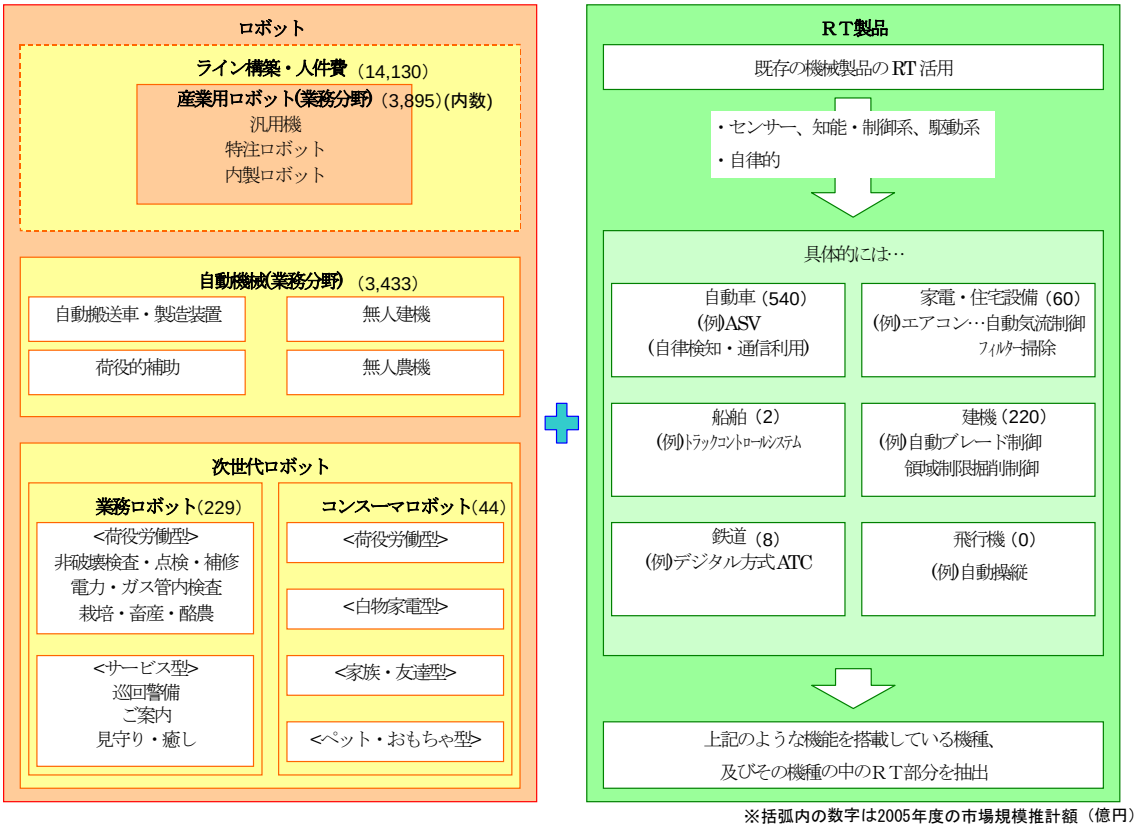


図４　ロボット及びロボテク（RT）製品の市場の捉え方

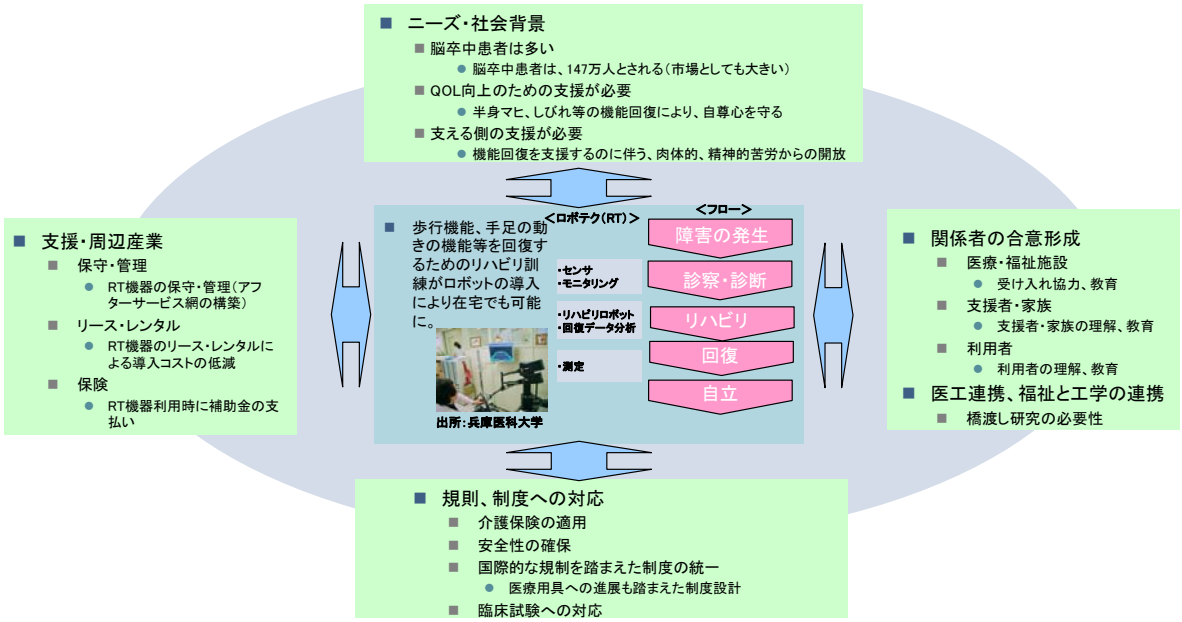


図５　（福祉介護分野）脳卒中患者の機能回復リハビリロボット
（人の支援のためのロボテク（RT））

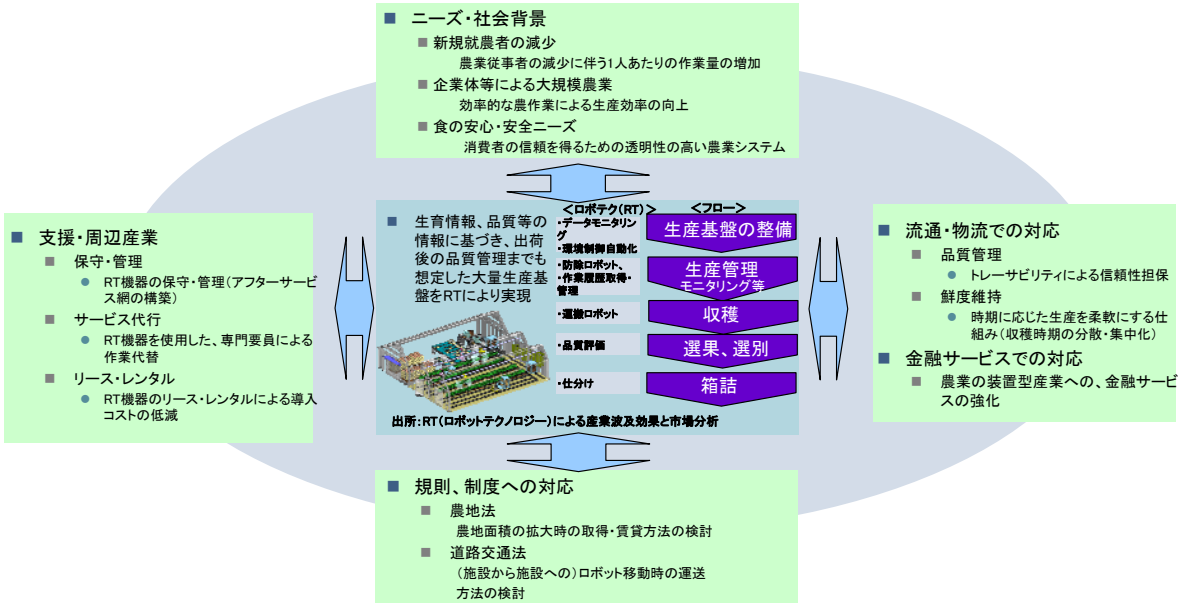


図６　（農業分野）次世代型施設生産システム（生産技術のためのロボテク（RT））

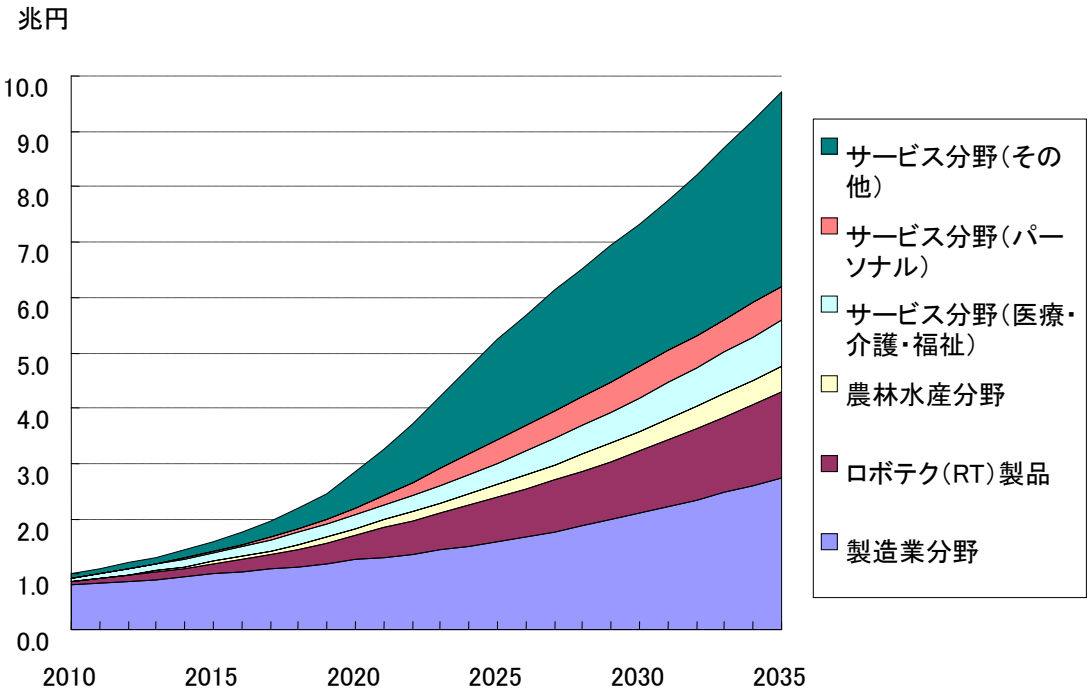


図７　ロボット産業の市場推移の予測

添付資料４

●特許論文リスト

【特許】

番号	出願者	出願番号	国内 外国 PCT	出願日	状態	名　　称	発明者
1	ミサワホーム株式会社	特願2011-045664	国内	2011/03/02	出願	ウインドウ装置	飯島雅人他
2	THK 株式会社		国内		出願書類作成中		

（※Patent Cooperation Treaty :特許協力条約）

【論文】

番号	発表者	所属	タイトル	発表誌名、ページ番号	査読	発表年
1	大原賢一	大阪大学	ホームネットワーク用 RT デバイスのモジュールデザイン	ロボット学会学術講演会，2F2-04	無	2009
2	大原賢一	大阪大学	Smart RT Device with Easy Replaceability in Ubiquitous Robot Environment	The 6th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2009), pp.194-198	有	2009
3	谷川民生	産業技術総合研究所	Smart home for security and low power consumption based on Ubiquitous Robotics	The 6th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2009)，pp.185-188	有	2009
4	大原賢一	大阪大学	RTC-Lite を利用したユビキタス・ロボットのためのモジュールデザイン	第 10 回計測自動制御学会システムインテグレーション部門講演会，3D3-3	無	2009
5	池添明宏	株式会社セック	RT ミドルウェアを活用した住宅環境管理・支援システムの開発	第 10 回計測自動制御学会システムインテグレーション部門講演会、p.1467-1469、3D2-5	無	2009
6	豊田光弘	株式会社セック	miniRTC:省資源マイコンで動作し複数の通信プロトコルに対応した RT ミドルウェア	第 10 回計測自動制御学会システムインテグレーション部門講演会、p.1494-1496、3D4-4	無	2009
7	大原賢一	大阪大学	分散配置されたロボット要素のための時刻同期モジュールの開発	日本機械学会ロボティクスメカトロニクス講演会 2010 講演論文集，1A2-G01	無	2010
8	小島 一浩	産業技術総合研究所	オープンソースハードウェアによる集合知センサーネットワークの構築とデバイス開発	日本機械学会ロボティクス・メカトロニクス講演会 2010 講演論文集，2P1-B05	無	2010
9	中本啓之	株式会社セック	RT ミドルウェアを活用した住宅環境管理・支援システムの開発（第 2 報）	日本機械学会ロボティクス・メカトロニクス講演会 2010、2P1-B02	無	2010

10	豊田光弘	株式会社 セック	組み込み向け軽量 RT ミドルウェアによるプラグアンドプレイ機能を有した RT システムの構築	日本機械学会ロボティクス・メカトロニクス講演会 2010、2A2-C06	無	2010
11	大原賢一	大阪大学	RT MODULE DESIGN FOR HOME AND BUILDING AUTOMATION	The 27th International Symposium on Automation and Robotics in Construction (ISARC2010), pp.165-174	有	2010
12	大原賢一	大阪大学	RT ミドルウェアを用いた無線ホームネットワークシステム	第 12 回 建設ロボットシンポジウム論文集 pp.217	無	2010
13	谷川民生	産業技術総合研究所	高セキュリティおよび低消費を実現する住宅の RT 化	第２８回日本ロボット学会学術講演会講演論文集 RSJ2010AC2P2-1	無	2010
14	谷川民生	産業技術総合研究所	Smart home for security and low power consumption with common network modules based on RT middleware	International Conference on Advanced Mechatronics 2010 (ICAM2010)	有	2010
15	大原賢一	大阪大学	System Integration for Wireless Home Network with RT Middleware	The 7th International Conference on Ubiquitous Robots and Ambient Intelligence, pp.494-497	有	2010
16	飯島雅人	ミサワホーム総合研究所	住宅への RT 活用の可能性	第 11 回計測自動制御学会 システムインテグレーション部門講演会(SI2010)、1E4-1	無	2010
17	豊田光弘	株式会社セック	RT ミドルウェアを活用した住宅環境管理・支援システムの開発（第 3 報）	第 11 回計測自動制御学会 システムインテグレーション部門講演会(SI2010)、1E4-2	無	2010
18	坂本武志	ﾃｸﾉﾛｼﾞｯｸｱｰﾄ	RT ミドルウェア開発支援ツールの開発	第 11 回 計測自動制御学会 システムインテグレーション部門講演会(SI2010) 、1E4-3	無	2010
19	大原賢一	大阪大学	RTC-Lite を利用した無線通信に基づくホームネットワークシステム構築	第 11 回計測自動制御学会 システムインテグレーション部門講演会(SI2010)、1E4-4	無	2010
20	遠藤嘉将	THK 株式会社	SEED 次世代ロボット向けエンドエフェクタ構成要素	第 11 回計測自動制御学会 システムインテグレーション部門講演会(SI2010)、1E4-5	無	2010

【講演】

番号	発表者	所属	タイトル	講演会名	発表年
1	大原賢一	大阪大学	ユビキタス・ロボティクスの関連技術紹介	若手研究者による RT シーズ発表会	2010
2	谷川民生	産業技術総合研究所	ユビキタスロボティクスからコンパクトシティ構想へ	若手研究者による RT シーズ発表会	2010
3	大原賢一	大阪大学	オープンプラットフォームに基づくホームネットワークシステム	ロボティクスフォーラム2011 in 大阪	2011

【展示会】

番号	発表者	所属	タイトル	展示会名	発表年
1	永塚正樹他	THK 株式会社	次世代ロボット向けエンドエフェクタ構成要素 SEED	国際ロボット展	2009
2	永塚正樹他	THK 株式会社	次世代ロボット向けエンドエフェクタ構成要素 SEED	第 10 回計測自動制御学会システムインテグレーション部門講演会(SI2009)	2009
3	永塚正樹他	THK 株式会社	次世代ロボット向けエンドエフェクタ構成要素 SEED	日本機械学会ロボティクス・メカトロニクス講演会 2010 講演会	2010

添付資料5

事前評価書

作成日	平成２０年3月２４日

１．事業名称	基盤ロボット技術活用型オープンイノベーション促進プロジェクト （２１世紀ロボットチャレンジプログラム）
２．推進部署名	機械システム技術開発部
３．事業概要	⑴ 概　要 我が国のロボット産業は、産業用ロボットの普及により製造業を中心に拡大発展し、今日、国際的にもトップレベルのロボット技術（以下、「ＲＴ」という。）を蓄積している。我が国の少子高齢化や安心・安全の問題が急速に進行しつつある中、ＲＴを製造業以外も含む様々な分野で活用することが期待されている。 本プロジェクトでは、これまでのロボット・プロジェクト等の開発成果を補完するものとして、生活環境やロボットで使用される各種要素部品を、ＲＴを駆使して機能を実現するシステム（以下、「ＲＴシステム」という。）で利用しやすい共通の接続方式、制御方式のもとで利用可能な形で提供（ＲＴコンポーネント化）するための基盤を開発する。またＲＴコンポーネント化された各種要素部品を用いることで既存の生活環境を簡単にＲＴシステム化し、さまざまな生活支援機能を提供することが可能であることを示す。 ⑵ 事業規模：総事業費１億円（平成 20 年度事業費１億円） ⑶ 事業期間：平成 20 年度～22 年度（3 年間）

４．評価の検討状況

（１）事業の位置付け・必要性

我が国のロボット産業は、産業用ロボットの普及により製造業を中心に拡大発展し、今日、国際的にもトップレベルのロボット技術（以下、「ＲＴ」という。）を蓄積している。我が国の少子高齢化や安心・安全の問題が急速に進行しつつある中、ＲＴを製造業以外も含む様々な分野で活用することが期待されている。具体的には、労働力不足や要介護者の増加などの課題を解決するとともに、犯罪、災害や医療等における将来への不安の軽減による安心で安全な社会を実現する手段として、ＲＴを駆使して機能を実現するシステム（以下、「ＲＴシステム」という。）を効率的に開発、実用化して、様々な分野で活用することが期待されている。例えば、家庭や職場の環境内でセンサやモータなど既存の部品をネットワーク接続してＲＴシステムを構築し、状況に応じた判断によって人の活動を支援できれば生活環境をより快適かつ安全にしたり、職場の生産性を向上させることができる。

しかしながら、ＲＴシステムを組み上げるには各種部品を集めて実装し、個々のシステムに合わせた制御ソフトを開発するという難しさ・煩雑さがあり、これがＲＴ分野への新規参入の障壁となっていた。そこで独立行政法人新エネルギー・産業技術総合開発機構（以下、ＮＥＤＯ技術開発機構という。）ではこの障壁を解消することを目的として、ＲＴミドルウェア、及び画像認識、音声認識、運動制御の機能を有する共通基盤モジュールなどのＲＴ開発基盤の整備を進めてきた。

本プロジェクトでは、これまでの開発成果を補完するものとして、生活環境やロボットで使用される各種要素部品を、ＲＴシステムで利用しやすい共通の接続方式、制御方式のもとで利用可能な形で提供（ＲＴコンポーネント化）するための基盤を開発する。

また、ＲＴコンポーネント化された各種要素部品を用いることで既存の生活環境を簡単にＲＴシステム化し、さまざまな生活支援機能を提供することが可能であることを示す。

本開発によってＲＴシステムの開発基盤を充実させることにより、製造分野をはじめとする一部の分野に限られているＲＴ適応分野を拡大することを本プロジェクトの第一の目的とする。更に、ロボット分野への中小・ベンチャーや異業種を含む多様な企業や研究機関等の新規参入を促進することにより、ロボット産業の裾野拡大を図ることを第二の目的として、本プロジェクトを実施する

（2）研究開発目標の妥当性

＜目標＞

本プロジェクトでは、生活環境やロボットに使われる既存の要素部品を、共通の通信インタフェースとＲＴミドルウェアで動作させる「基盤通信モジュール」を開発する。次に、「基盤通信モジュール」を用いることにより既存の要素部品が容易にRT コンポーネント化でき、ＲＴシステム内で共通して利用できることを示すとともに、それを「ＲＴ要素部品」として広く提供する。さらに「ＲＴ要素部品」を用いた「ＲＴシステム」を開発し、同システムの有効性を検証するために実証試験を行うことを目標とする。具体的には、次の３つの研究開発項目について、別紙の研究開発計画に基づき研究開発を実施する。

①基盤通信モジュールおよび開発ツールの開発

②基盤通信モジュールを用いたＲＴ要素部品の開発

③ＲＴ要素部品群によるＲＴシステムの開発・実証

＜妥当性＞

ＲＴシステムを組み上げるには各種部品を集めて実装し、個々のシステムに合わせた制御ソフトを開発するという難しさ・煩雑さがあり、これがＲＴの多様な発展、ロボット分野への中小・ベンチャーや異業種を含む多様な企業や研究機関等の新規参入の障壁となっている。

本プロジェクトは、これらＲＴシステム開発に係る課題を解決するため上記目標を掲げており、いずれもＲＴシステム開発の低コスト化・効率化に必要な技術開発であり、ＲＴ適応分野の拡大、異業種を含む多様な企業・研究機関等の新規参入を促進し、ロボット産業の裾野拡大を図ることが期待できる。

（3）研究開発マネジメント

研究開発全体の管理・執行に責任を有するＮＥＤＯ技術開発機構は、経済産業省及びプロジェクトリーダと密接な関係を維持しつつ、プログラムの目的及び目標並びに本研究開発の目的及び目標に照らして適切な運営管理を実施する。具体的には、必要に応じて、ＮＥＤＯ技術開発機構に設置する推進委員会等、外部有識者の意見を運営管理に反映させるほか、四半期に一回程度プロジェクトリーダ等を通じてプロジェクトの進捗について報告を受けること等を行う。

また、ＮＥＤＯ技術開発機構は、過去のロボット・プロジェクト成果を本プロジェクトで可能な限り活用するとともに、「次世代ロボット知能化技術開発プロジェクト」など推進中のロボット・プロジェクトとの相互協力が円滑に行われるようプロジェクト間連携を推進する計画であり、マネジメント体制として妥当と考える。

（4）研究開発成果

実施者は本プロジェクトの成果を、プロジェクト期間内においてはＮＥＤＯ技術開発機構が推進する他のプロジェクトに積極的にサンプル提供（有償提供を含む）すること、また、プロジェクト終了後においても得られた研究成果について継続的な研究、継続的なデモシステムの展示及び研究発表を行うなどして、広く成果の普及に努めるとともに、プロジェクト終了後２年を目途に、プロジェクトの成果を活用して、基盤通信モジュール、ＲＴ要素部品、およびＲＴシステムなどの製品化を目指すことを基本計画上に明記しており、早期に本プロジェクト成果を事業化するために必要な取り組みとして有効かつ妥当と考える。

（5）実用化・事業化の見通し

２０１２年以降に本研究開発成果を活用したＲＴ製品の実用化・事業化が見込まれる。

（6）その他特記事項

特になし。

5．総合評価

本プロジェクトは、ＲＴによる少子高齢化等の解決を期待する市場ニーズ及び既存ＲＴ技術の更なる活用を促進するための技術ニーズを適切に踏まえたうえで、ＲＴ産業の課題である開発及び事業化における低コスト、短工期を実現するために不可欠な技術開発を目標としていることから、ＮＥＤＯ技術開発機構にて実施することが適切であると判断する。

（別紙）

研究開発項目①「基盤通信モジュールおよび開発ツールの開発」

【研究開発の具体的内容】

既存のセンサ、モータなどの要素部品をネットワーク接続可能としＲＴコンポーネント化するための「基盤通信モジュール」及び「開発ツール」の開発を行う。

【達成目標】（平成２２年度）

（1）基本性能

以下の条件を満たす「基盤通信モジュール」及び「開発ツール」を開発すること。

①ＲＴミドルウエアを実装し、研究開発項目②で開発する「基盤通信モジュール」を利用したＲＴ要素部品がＲＴシステムから OpenRTM 仕様に基づきＲＴコンポーネントとして利用できること。

②独自のネットワークを用いるのではなく、既存の標準化されたネットワークと接続可能とすること。また、「基盤通信モジュール」間の通信は特別な理由がない限り、既存の標準化された方式を用いること。

③家庭や職場の環境内に構築するＲＴシステムで必要となる要素部品と接続可能なインタフェースを有すること。このインタフェース仕様は、要素部品の使われ方を考慮して設定すること。

（2）その他の性能

①低消費電力化を目指すこと。

②種々の要素部品との組み合わせを考慮して、できる限り小型化軽量とし、望ましくはサイズ（基板面積）を名刺の 1/2 以下とすること。ただし、コネクタと電源はこのサイズに納めなくても良い。プロジェクト終了後、無理なく実用化できること。

（3）開発品の提供

①開発した「基盤通信モジュール」をＲＴ要素部品開発機関が利用できる形で提供すること。

②(ｉ)各種要素部品との接続、(ii)要素部品の制御・信号処理プログラム及び(iii)ネットワークとの通信設定とその保持などを容易とする「開発ツール」を、ＲＴ要素部品開発機関が利用できる形で提供すること。

③開発した「基盤通信モジュール」及びＲＴミドルウェアにて動作させる際に必要となる項目を記載した仕様書及び取扱説明書をＲＴ要素部品開発機関が利用できる形で提供すること。

（4）有効性の検証

①開発した「基盤通信モジュール」及び「開発ツール」が開発仕様を満たし、有効に機能することをコンソーシアムメンバ間で協力して検証すること。

②ＲＴシステムの実証に際しては、ＲＴシステム開発機関と協力して、開発品が有効に機能することを検証すること。

研究開発項目②「基盤通信モジュールを用いたＲＴ要素部品の開発」

【研究開発の具体的内容】

研究開発項目①で開発された「基盤通信モジュール」と、既存のセンサ、モータなどの要素部品とを「開発ツール」を用いて接続し、ネットワーク接続及びシステム化を可能とする「ＲＴ要素部品」の開発を行う。「基盤通信モジュール」に代えて、「次世代ロボット共通基盤開発プロジェクト」にて開発された、「画像認識、音声認識、運動制御の機能を有する共通基盤モジュール」から使用できるものを選定して利用しても良い。

【達成目標】（平成２２年度）

以下の条件を満たす「ＲＴ要素部品」を開発すること。

（1）基本性能

①「基盤通信モジュール」、あるいは「共通基盤モジュール」と組み合わされていること。これらは要素部品と一体化されていることが望ましいが、処理部として分離されても良い。

②ＲＴシステムから OpenRTM 仕様に基づきＲＴコンポーネントとして利用できること。

（2）開発品の提供

開発したＲＴ要素部品及びＲＴミドルウェアにて動作させる際に必要となる項目を記載した仕様書及び取扱説明書を、ＲＴシステム開発機関が利用できる形で提供すること。

（3）有効性の検証

①開発した「ＲＴ要素部品」が開発仕様を満たし、有効に機能することをコンソーシアムメンバ間で協力して検証すること。

②ＲＴシステムの実証に際しては、ＲＴシステム開発機関と協力して、開発品が有効に機能することを検証すること。

研究開発項目③「ＲＴ要素部品群によるＲＴシステムの開発・実証」

【研究開発の具体的内容】

研究開発項目①及び②で開発されたＲＴ要素部品群を用いてＲＴシステムを開発し、その実証システムを家庭や職場を模擬した環境内に構築して有効性の検証を行う。

【達成目標】（平成２２年度）

以下の条件を満たすＲＴシステムを開発して有効性を検証するための実証を行うこと。実証に際しては、プロジェクト期間内に、必要に応じて第三者に対してデモンストレーションすること。

（1）基本性能

①ＲＴ要素部品を組み合わせたＲＴシステムとすること。ＲＴシステム化の際に必要となる制御装置も開発すること。

②OpenRTM 仕様に基づくＲＴシステムとして開発すること。

③システムインテグレータ（本プロジェクトではＲＴシステム開発機関がこれに相当）や、ある程度のスキルを持ったエンドユーザが容易にシステムの構築、変更ができるシステム構築・運用ツールを開発すること。

④実証システムは実証目的に合わせて、要素部品またはＲＴシステムの入れ替えが可能であること。

⑤実証システムは、要素部品としてセンサとアクチュエータとを含める構成とすること。

（2）その他の性能

①家庭や職場の環境内で構築されるＲＴシステムは、バッテリや省電力での駆動が想定されるので以下の条件を満たすこと。

②システム全体の省電力を意識した構成とすること。

③家庭や職場の環境で想定される温度、湿度、照度などの諸条件のもとで正常に動作すること。

（3）有効性の検証

①実証システムや小規模な個別デモシステムなどを用いて、推進委員会等によるＲＴシステム及び実証システムの有用性評価を適時受けること。

②実証で得られた結果や知見を基盤通信モジュール開発機関及びＲＴ要素部品開発機関にフィードバックし、必要に応じて改良を促すこと。