

## RTミドルウェアサマーキャンプ2012

様々なRTミドルウェア実装による  
ロボットシステムの構築  
～RTミドルウェア搭載スマホでロボット制御&  
安心・安全のための機能安全対応RTミドルウェア～

2012年8月2日  
開発本部 第四開発部



## アジェンダ

- 株式会社セックのご紹介
- RTミドルウェア導入のメリット
- ロボットシステムの事例紹介
  - 様々なRTミドルウェア
  - レーザレンジセンサシステム
  - インフォメーションロボットシステム
  - RT技術の住宅への適用
  - Android版RTミドルウェア「RTM on Android」
  - 機能安全対応RTミドルウェア「RTMSafety」





**中本啓之**  
HIROYUKI NAKAMOTO

**株式会社セック**  
開発本部第四開発部

**自己紹介:**

2003年頃よりロボット分野に従事するようになりました。RTミドルウェアやRSiなどのロボット技術の標準化活動や次世代ロボット知能化技術開発プロジェクトなどの国のロボットプロジェクトに参画してきました。

**RTミドルウェアが良いと思うところ:**

- ◆モジュール化技術により、ロボットシステムの開発効率や再利用性の向上に寄与できること。
- ◆オープンな技術であり、さまざまな規格やプラットフォームをつなぐ架け橋となれること。

**トラブル事例(体験):**

- ◆RTミドルウェアを人に説明するのが難しい。誰でも簡単に理解して使えるようにしていきたいです。

**何でも提案:**

- 1)ロボットの安全性・信頼性向上と生産性向上・コストダウンの相反する二つの課題解決に貢献していきたいです。



RTミドルウェアを使った  
インフォメーションロボット

機能安全認証取得  
RTミドルウェア  
「RTMSafety」



RTMSafety

**参加者にひとこと:**

アイデア次第で誰でもおもしろいロボットシステムが作れます。みんなでRTミドルウェアを盛り上げていきましょう！

**連絡先:**

nakamoto (at) sec.co.jp  
<http://www.sec.co.jp/robot/index.html>

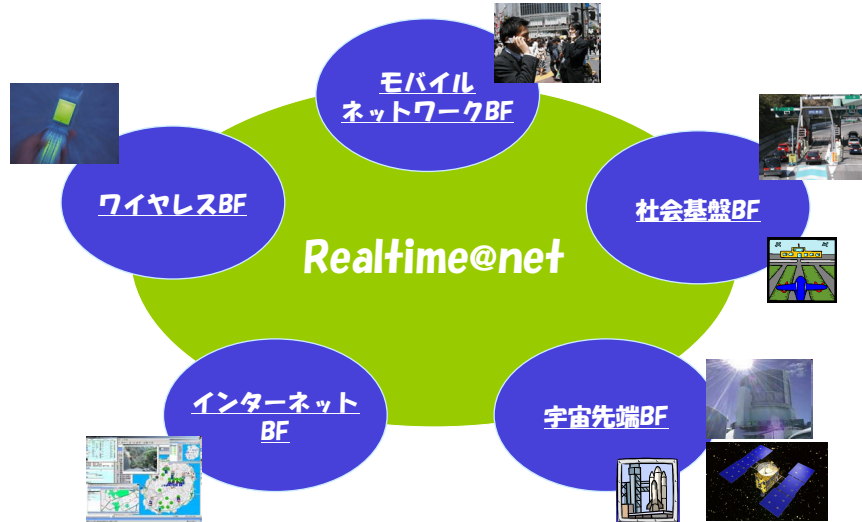
## 株式会社セックのご紹介

- 社名：株式会社セック（略称：SEC）  
Systems Engineering Consultants Co.,LTD.
- 設立：1970年5月
- 資本金：4億7730万円（2012年6月27日現在）
- 従業員数：261名（2012年4月1日現在）
- 本社：東京都世田谷区  
世田谷ビジネススクエア



Copyright© 2012 Systems Engineering Consultants Co., LTD. All rights reserved.

# 株式会社セックのご紹介

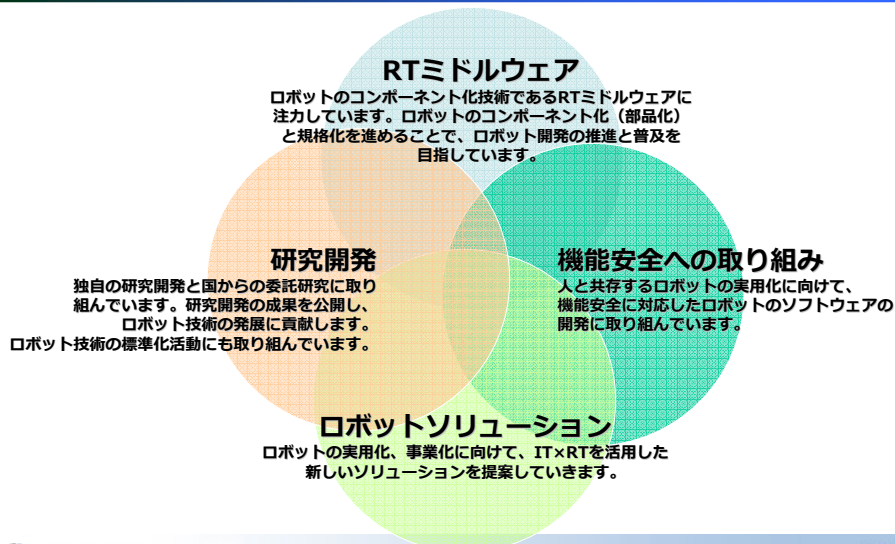


# 株式会社セックのご紹介

ビジネスフィールド	技術サービス			リアルタイムソリューション
	リアルタイムソフトウェア			
	エンベデッドソフトウェア	コアテクノロジー	技術アプリケーション	
モバイルネットワーク		オープンプラットフォーム技術 携帯電話端末搭載 ソフトウェア国際標準技術	基幹ネットワーク	RealtimePowerシリーズ※ RTMSafety
ワイヤレス	スマートフォン タブレット 携帯電話端末	地上デジタル放送技術		airCube airLook
インターネット	非接触型IC 情報家電 車載端末	ペナル播磨技術 XML	Webシステム	airSmartG
社会基盤システム		位置情報サービス技術	防衛 交通(ITS,航空) 環境エネルギー	Karearea
宇宙先端システム	人工衛星搭載機器 ロボット	ロボットソフトウェア 共通化技術	人工衛星 天文	

※「RealtimePower」はセックのソリューション製品の総称で、登録商標です。

## ロボット分野への取り組み



## アジェンダ

- 株式会社セックのご紹介
- RTミドルウェア導入のメリット
- ロボットシステムの事例紹介
  - 様々なRTミドルウェア
  - レーザレンジセンサシステム
  - インフォメーションロボットシステム
  - RT技術の住宅への適用
  - Android版RTミドルウェア「RTM on Android」
  - 機能安全対応RTミドルウェア「RTMSafety」

## RTミドルウェア

### ■ 特にRTミドルウェアに注力！



- ロボットラボラトリー主催  
「RTミドルウェア研究会ビジネスモデル検討会」
- 福岡ビジネス創造センター主催  
「ロボットビジネスセミナー  
～ロボット知能ソフトウェアプラットフォームの研究開発～」
- 書籍  
「はじめてのコンポーネントベース・ロボットアプリケーション開発  
～RTミドルウェア超入門～」
- Think IT 2009年6月特集記事  
「SEのためのRTシステム概論」 <http://thinkit.co.jp/article/950/1>
- ロボコンマガジン No.82 2012年7月特集記事  
「Android端末で動くロボットソフトウェアを作ろう」
- RTミドルウェア講習会
  - 名城大学 理工学部 電気電子工学科
  - 玉川大学 工学部 機械情報システム学科
  - 金沢工業大学 ロボティクス学科
  - 電気通信大学 高度ICT試作実験公開工房ピクトラボ [第1回ピクトラボ匠講演会](#)
  - RTミドルウェアサマーキャンプ2011/2012



## RTミドルウェアとは？

- RTミドルウェアは、ロボットを構成する要素（アクチュエータやセンサなど）やロボットを制御するソフトウェアのコンポーネント化技術
- 2008年4月に国際標準化団体OMG（Object Management Group）にて標準化
- 産総研がオープンソースのミドルウェアOpenRTM-aistを開発、提供



コンポーネントを組み合わせるだけでロボットを容易に開発

## RTミドルウェアとは？

### ■ RTミドルウェアとは、ロボットの部品化・共通化の技術

- ミドルウェア導入により、部品の再利用化が進む
  - ミドルウェア導入により、部品調達のオープン化が進む
- ロボット開発のコストダウン、産業化のための技術

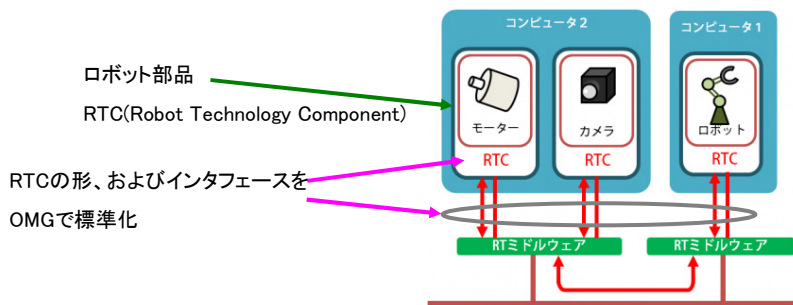
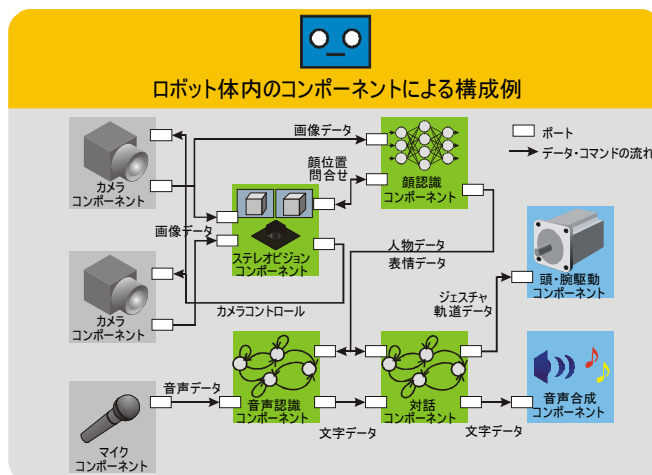


図1:RTミドルウェアの概念図

## モジュールを組み合わせたロボット構成



資料作成/提供：産業技術総合研究所

## RTミドルウェア導入のメリット1

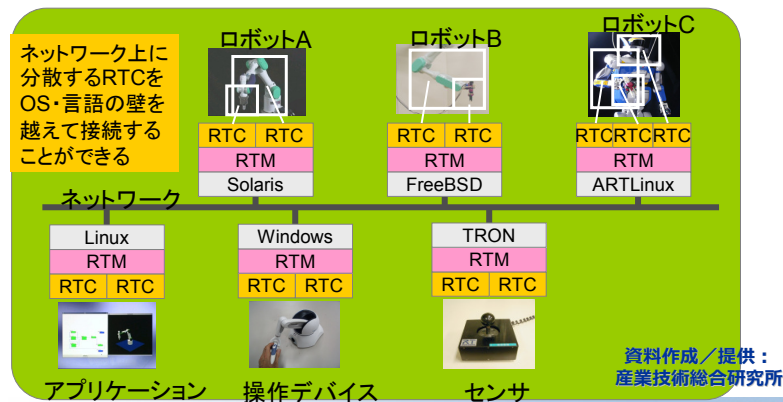
### ■ マルチプラットフォーム対応

- 単体のロボットだけでなく、ロボット周辺システムや、ロボット操作系のシステムともシームレスに結合する
- 省資源マイコンで動作し、センサーネットワークのプラットフォームにも適用可能
- ROSにはない機能

## RTミドルウェア導入のメリット2

### ■ ネットワーク分散

- ロボット体内LANやネットワークロボットなど、分散システムを容易に構築可能



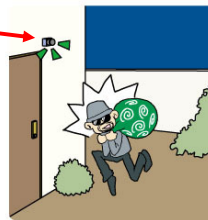
## RTミドルウェア導入のメリット3

### ■再利用率の向上

- 同じコンポーネントをいろいろなシステムに  
使いまわせる



ロボット組み



防犯センサシステム組み

## RTミドルウェア導入のメリット4

### ■選択肢の多様化

- 同じ機能を持つ複数のモジュールを試すこと  
ができる



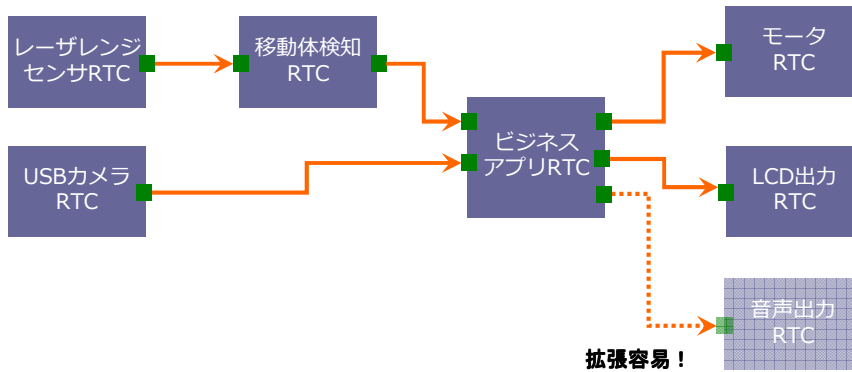
レーザレンジファインダの入れ替え容易  
(ソフト変更不要)



## RTミドルウェア導入のメリット5

### ■柔軟性の向上

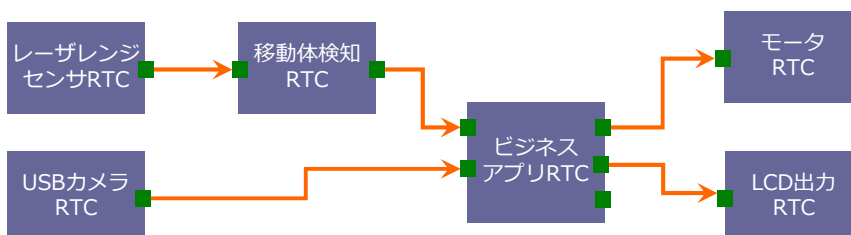
- モジュール接続構成を変えるだけで様々なシステムを構築できる



## RTミドルウェア導入のメリット6

### ■信頼性・堅牢性の向上

- モジュール単位でテスト可能なため信頼性が向上する
- システムがモジュールで分割されているので、一つの問題が全体に波及しにくい



## アジェンダ

- 株式会社セックのご紹介
- RTミドルウェア導入のメリット
- **ロボットシステムの事例紹介**
  - 様々なRTミドルウェア
  - レーザレンジセンサシステム
  - インフォメーションロボットシステム
  - RT技術の住宅への適用
  - Android版RTミドルウェア「RTM on Android」
  - 機能安全対応RTミドルウェア「RTMSafety」

## さまざまなRTミドルウェア

用途に合わせ、多様なRTミドルウェアを開発・提供しています。

### エンタープライズ層



PC/タブレット向け

OpenRTM.NET

Android版RTミドルウェア  
RTM on Android™

### ロボット層

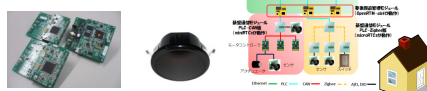


ロボット制御向け

OpenRTM-aist for  
VxWorks

機能安全対応  
RTミドルウェア

### エンベデッド/デバイス層



省資源マイコン向け

miniRTCs-CAN

microRTCs-Zigbee

## ロボットシステムの活用事例 ～レーザレンジセンサシステム～

### Laser Range Sensor System



障害物検知に

0～80mくらいまでの距離で屋内外での障害物検知が可能です。車、船、移動ロボットなどに搭載し、衝突回避に役立ちます。

資材管理に

倉庫などで、大型資材の管理が可能です。特に、日照条件の変化が激しく、画像処理では、困難な場所での管理に役立ちます。

受付・案内に

人を検知して呼び掛けを行い、人の位置に応じて案内を変えることが可能です。企業の受付システムとして、観光などの案内システムとして使えます。

防犯対策に

人を検知してカメラで自動追尾が可能です。1台のセンサで複数のカメラで高速に追尾することが可能なため、新しい防犯カメラとして使えます。

動線計測に

人流の動線を計測することで、マーケティングツールとして使用可能です。動線データはパソコンに蓄積され、通過人数や滞留時間の分析に使えます。

SEC Systems Engineering Consultants Co., Ltd.

## ロボットシステムの活用事例 ～レーザレンジセンサシステム～

レーザレンジファインダを使って不審者を検知し、カメラで自動追跡・撮影するロボット技術を活用した次世代の防犯システムです。



株式会社セック  
SEC Systems Engineering Consultants Co., Ltd.

Copyright© 2012 Systems Engineering Consultants Co., LTD. All rights reserved.

## ロボットシステムの活用事例

### ～インフォメーションロボットシステム～

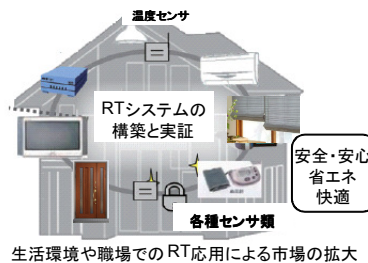
- インフォメーションロボットシステムは、人間などの移動体を捕捉し高精度に追跡を行うことができるロボット
- 頭部に搭載した液晶ディスプレイを使って、ユーザーに様々な情報を伝えることが可能
- RTミドルウェアを採用し、高い拡張性と保守容易性を備える
- 防犯システムをはじめ、自動受付システムや店舗内の動線計測システムなど、様々な分野に応用することが可能



## ロボットシステムの活用事例

### ～RT技術の住宅への適用～

- 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
  - NEDOからの委託研究（2008～2010年）
  - RT（Robot Technology）システムの開発基盤の標準化
  - RTの活用事例の創出
- 窓、家電、センサなどの住宅設備をRTミドルウェアにより連携
  - 国際標準規格のRTミドルウェアを用いて、様々な住宅設備を連携
  - 住宅のRT化により人間の生活支援や安心安全、省エネルギー化を実現



## ロボットシステムの活用事例 ～RT技術の住宅への適用～

- ロボット部品（センサやアクチュエータ）やIT家電をRTミドルウェアで連携させ住宅をRT化
- IT+RTで今までにないサービスや機能を提供
- RTミドルウェアにより、複数ベンダのシステムをシームレスに連携



## ロボットシステムの活用事例 ～RT技術の住宅への適用～



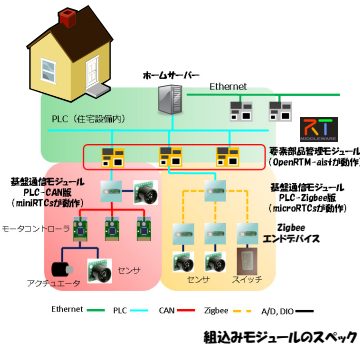
## ロボットシステムの活用事例 ～RT技術の住宅への適用～

- 窓やドア、家電、センサなどに組み込みモジュールと軽量RTミドルウェアを搭載しRTモジュール化



住宅内のネットワーク上に分散配置されたRTモジュールが、RTミドルウェア仕様で連携・協調することで、環境分散型の住宅RTシステムを実現

- ホームコントローラによる集中管理ではなく、**階層的な分散管理**のアーキテクチャを採用
- ネットワークプロトコルとして、窓開閉などのリアルタイム制御、通信の信頼性が必要なところは**CAN**、配置の自由度が必要なところは無線ネットワークの**Zigbee**、住宅内の基幹ネットワークには**PLC**を採用



組み込みモジュールのスペック

専用品管理モジュール	基幹通信モジュール(CAN/Zigbee)	モータコントローラ (CAN)	Zigbeeエンドデバイス
CPU TI AM1808(375MHz)	CPU SH7214/SH2(100MHz)	CPU ARM7(60MHz)	CPU TI CC2530(16MHz)
ROM 2GB	ROM 1MB	ROM 256KB	ROM 256KB
RAM 256MB	RAM 128KB	RAM 16KB	RAM 8KB
OS Linux	OS TOPPERS ASP	OS なし	OS なし
RTM OpenRTM-aist	RTM miniRTCs / microRTCs	RTM miniRTCs	RTM microRTCs

オープンな規格を採用し、様々なメーカーの機器が連携可能に！



Copyright© 2012 Systems Engineering Consultants Co., LTD. All rights reserved.

27

## ロボカップ@ホーム

- ロボカップ@ホームのタスク例
  - Robot Inspection and Poster Session(RISP)
  - Follow Me
  - Go Get It!
  - Who Is Who
  - General Purpose Service Robot
  - Shopping Mall



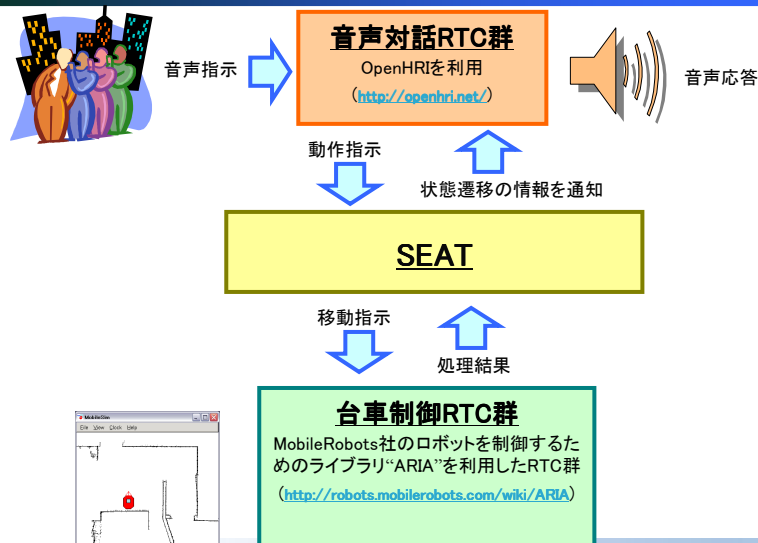
Copyright© 2012 Systems Engineering Consultants Co., LTD. All rights reserved.

28

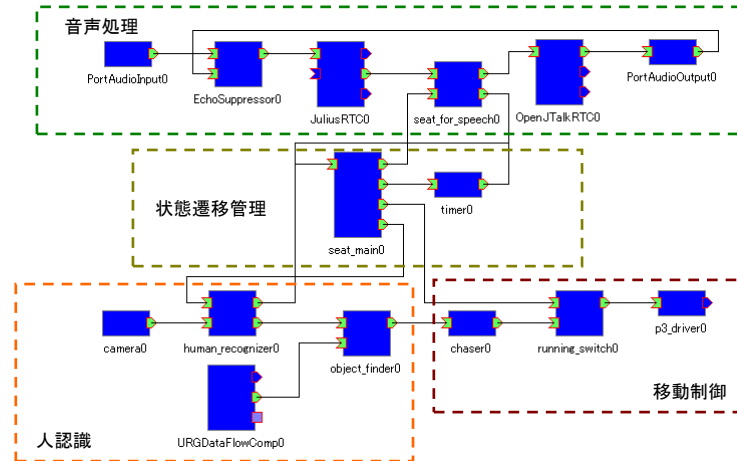
# ロボカップ@ホーム

モジュール分類	モジュール名	モジュール補足	利用可能なライブラリ
自律移動	大域的経路計画	地図からの経路生成	知能化PJモジュール
	局所的経路計画	軌道追従、衝突回避	知能化PJモジュール
	障害物検知	LRF	知能化PJモジュール
	自己位置推定	オドメトリ、SLAM	知能化PJモジュール
	地図更新	SLAM	知能化PJモジュール
	人検知・追跡	カメラ	知能化PJモジュール
	モータ出力	動作指令	知能化PJモジュール
	センサ入力	LRF、(ステレオ)カメラ	—
コミュニケーション	音声認識	日本語、英語	OpenHRI
	対話スクリプト	—	
	音声合成	—	
	音声出力	スピーカー	—
	顔認識・認証	見知らぬ人の顔と名前を覚える	オムロン OKIAO Vision OpenCV
マニピュレーション	ジェスチャー認識	身振り、手振りの認識	オムロン OKIAO Vision OpenCV
	センサ入力	カメラ、マイクロフォンアレイ	音声認識モジュール
	順・逆運動学計算	—	OpenRAVE
	軌道生成	—	OpenRAVE
	ハンドアイマニピュレーション	—	OpenRAVE
	物体探索・移動・認識・把持	ペットボトル、ドア、冷蔵庫、缶	OpenRAVE
	バンケット台	—	OpenCV
システム	オブジェクトの発見・識別	—	OpenCV
	センサ入力	TOF、ステレオカメラ	—
	コントローラ	マニピュレーション制御	—
	タスク管理	シナリオによるタスク実行制御	シナリオエディタ フィロソフィー
システム	モジュール操作	RTシステムエディタの自動化	rtsshell rtcshell
	システム管理	動作状況管理、ログ管理	—

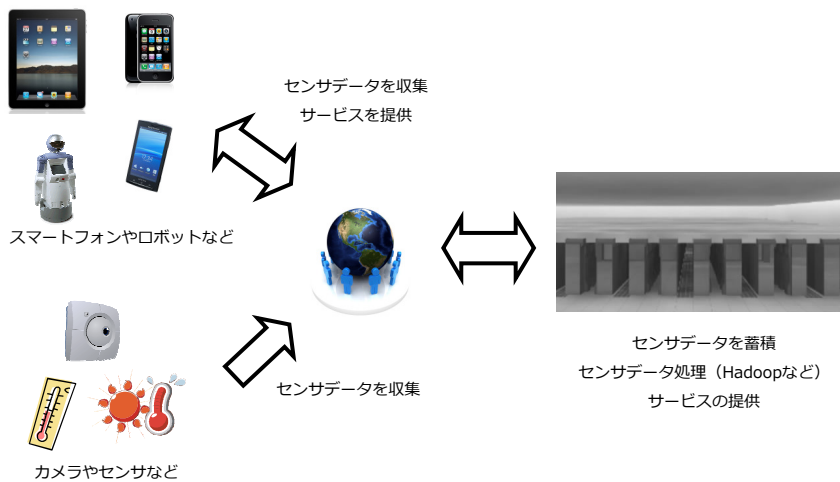
# Follow Meのコンポーネント構成



# Follow Meのコンポーネント構成



# ロボットシステムの活用事例 ～ロボットシステムとクラウドの連携～





## ロボットシステムの活用事例 ～RTM on Android～



RTM on Androidは、Android端末に対応したRTミドルウェアです。  
RTM on Android を用いることで、ロボットやセンサーがAndroid端末と連携するシステムを迅速かつ安価に作成することが可能になります。






## ロボットシステムの活用事例 ～RTM on Android～



### RTM on Android とは

- ✓OMGで承認された国際規格であるRTC SpecificationをAndroid上に実現
  - Androidタブレットを用いることで、場所を選ばずにロボットの制御・監視が可能になります
- ✓OpenRTM-aist-1.0と相互運用可能
  - RTミドルウェアを使用した既存ロボット/センサが利用できるため、開発コストを下げ、開発期間を短くすることができます
- ✓分散ミドルウェアCORBAには産総研が開発したRtORBを採用

### 例えば、こんな使い方

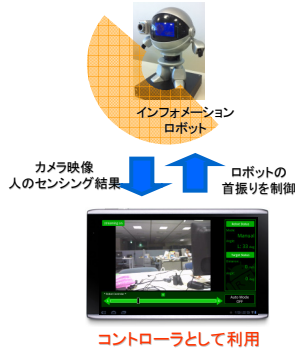
- ✓産業用ロボットの生産管理端末として 
- ✓スマートハウスで部屋ごとの電力消費量や家電の状態を確認する端末として 
- ✓ロボットの遠隔操作やモニタリングを行う端末として 

# ロボットシステムの活用事例 ～RTM on Android～



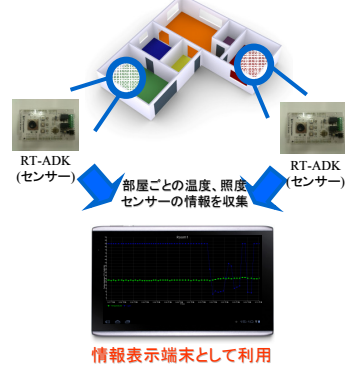
## SCENE1: ロボットの遠隔制御・監視

- Android端末を用いてロボットを遠隔制御
- ✓タブレットからロボットを操作
- ✓ロボットのカメラ映像をタブレットで表示



## SCENE2: センサーネットワーク

- Android端末をセンサネットワークのデバイスとして活用
- ✓部屋情報をセンサーと接続したAndroid端末から収集
- ✓収集結果をタブレットに表示



RT × Androidでロボット／センサーを制御・監視



Copyright© 2012 Systems Engineering Consultants Co., LTD. All rights reserved.

35

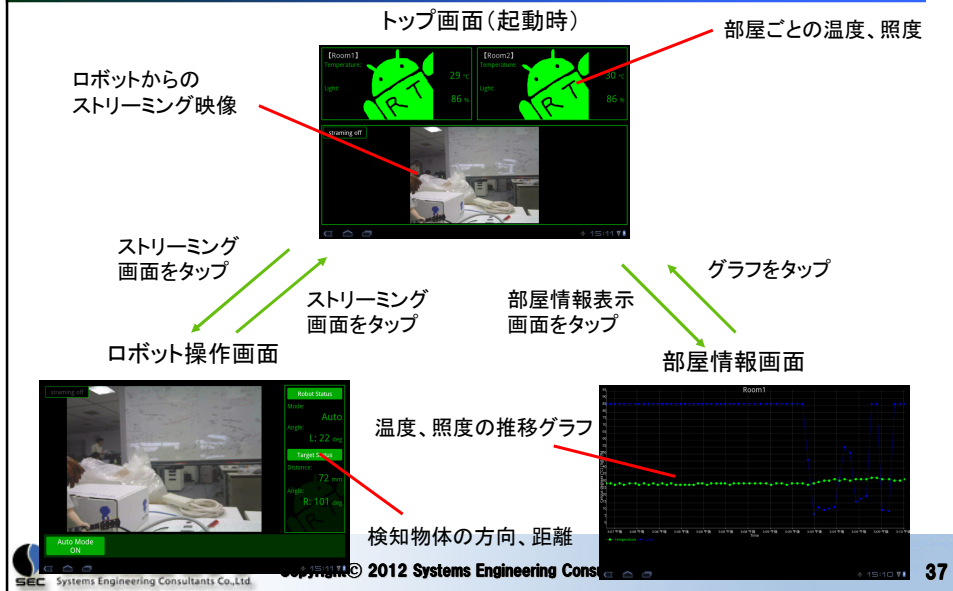
# ロボットシステムの活用事例 ～RTM on Android～



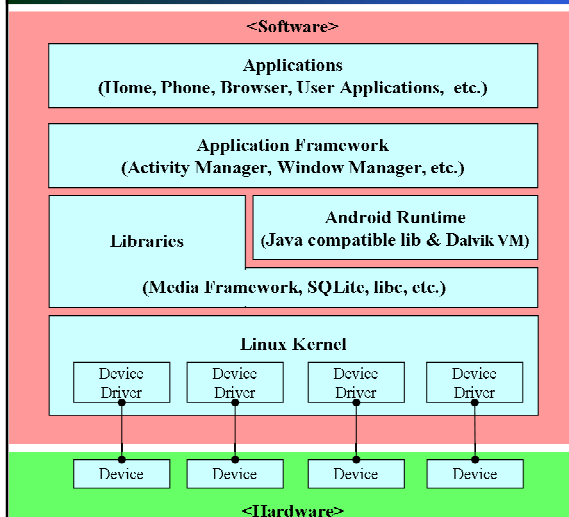
Copyright© 2012 Systems Engineering Consultants Co., LTD. All rights reserved.

36

# ロボットシステムの活用事例 ～RTM on Android～

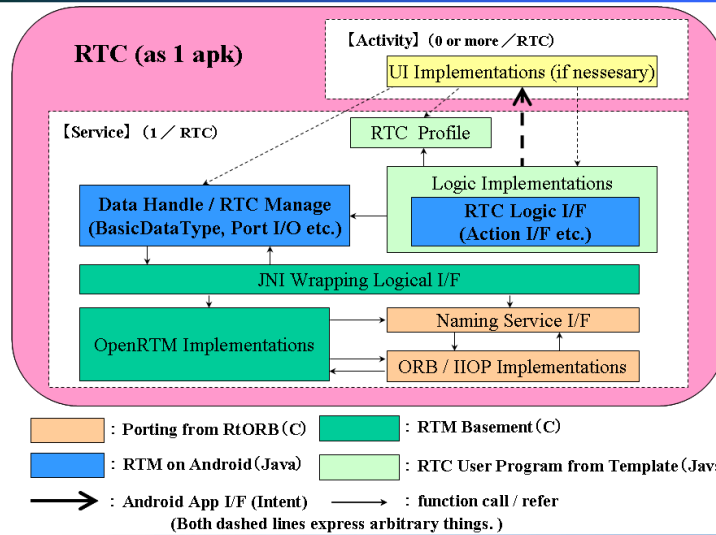


# Androidアーキテクチャとアプリケーション



- Android Runtimeに専用VM (Dalvik VM)
  - 各アプリケーションは、それぞれが独立したLinuxのプロセス
  - 各アプリケーションに一つのDalvik VMインスタンスが対応して動作
- 一般的Androidアプリケーションは、UIを持つコンポーネントであるActivityとして実装される
  - Activityは画面の最上位に表示される場合にのみ動作
- UIを持たず、バックグラウンドで継続的に動作するServiceも代表的なコンポーネント単位

# RTM on Android のアーキテクチャ



# RTM on Android でのRTC開発

- **ごく普通のAndroidアプリ開発手法と同じ**
  - Google社により提供されているEclipse用SDKを利用
  - アプリケーション単位にAndroidプロジェクトを作成
  - ソースの**編集からビルドまで**を実施
  - RTCとしての動作は**シンプルなAPI記述**で実現可能



```

10 public static final String default_name_name = "10";
11 public static final String name = "SimpleRTM";
12 public static final String implmentation_id = "Simple";
13 public static final String type = "SimpleComponent";
14 public static final String description = "Simple RTC";
15 public static final String version = "1.0";
16 public static final String vendor = "Systems Engineer";
17 public static final String category = "Sample";
18
19 public static final float execute_rate = 10.0F; // 1
20 // PLEASE EDIT RIGHT COLUMN --- end ---
21
22 // DO NOT DELETE STATIC MAP DECLARATION BELOW
23 public static Map<String, String> profiles = new Map<>();
24 put ( RTPThread.NAME_SERVER, MyRTC_Profile.default
25 put ( RTPThread.NAME_SERVER, MyRTC_Profile.name );
26 put ( RTPThread.IMPLEMENTATION_ID, MyRTC_Profile.im
27 put ( RTPThread.TYPE, MyRTC_Profile.type );
28 put ( RTPThread.DESCRPTION, MyRTC_Profile.description
29 put ( RTPThread.VERSION, MyRTC_Profile.version );
30 put ( RTPThread.VENDOR, MyRTC_Profile.vendor );
31 put ( RTPThread.CATEGORY, MyRTC_Profile.category );
32 put ( RTPThread.EXECUTE_RATE, String.valueOf( My
33 );
    
```

## RTM on Android使用上の注意点

- 画面オフや回転への注意
  - Androidバージョン2系では、画面オフ時や回転時にonDestroy()やonCreate()が内部で呼ばれてしまう
  - このため、RTCの起動・終了方法によっては、意に反して連動して終了してしまう危険性がある
  - 対処方法(実際にサンプルプロジェクトを修正)
    - 画面レイアウト定義への指定にて画面オフ抑止
    - マニフェストへの指定にて回転を抑止
- omni-ORBとの親和性への注意
  - omni-ORBは、デフォルトで一定時間後にタイムアウト処理が走る
  - RTM on Androidが採用しているRtORBはこれに未対応
  - 対処方法(実際にOpenRTM-aistのサンプルRTC用rtc.confを修正)
    - omni-ORBを利用するRTCのコンフィグレーション指定にて、タイムアウトを抑止
- RT SystemEditorとの親和性への注意
  - RTM on Androidを利用したRTCは、RT SystemEditor上での操作感が悪く、RT SystemEditorが無応答に陥ることもある
  - 対処方法(本日は適用外)
    - RT SystemEditorを使わない

## RTM on Androidでの独自型の使用

- 独自型を使用する場合は、データクラスを用意する  
※データのアライメントに注意する必要がある

参考：[http://www.openrtm.jp/wiki/hara/ja/RtORB/RtORB\\_CDR.html](http://www.openrtm.jp/wiki/hara/ja/RtORB/RtORB_CDR.html)

```
import java.util.List;
public class CameraImage implements
    Marshalizable {
    private static final String dataType =
        "CameraImage";

    public RTCTime tm = null;
    public short width;
    public short height;
    public short bpp;
    public String format;
    public double fDiv;
    public List<Byte> pixels;

    public CameraImage() {
        tm = new RTCTime(0, 0);
        pixels = new ArrayList<Byte>(0);
    }

    public CORBA_CdrData marshal() {
        Marshalizer mslzr = new Marshalizer();
        mslzr.marshalLong(getTm().getSec());
        mslzr.marshalLong(getTm().getNsec());
        mslzr.marshalShort(width);
        mslzr.marshalShort(height);
        mslzr.marshalShort(bpp);
        mslzr.marshalShort(0);
    }
}
```

```
mslzr.marshalString(format);
mslzr.marshalDouble(fDiv);
mslzr.marshalByteSeq(pixels);

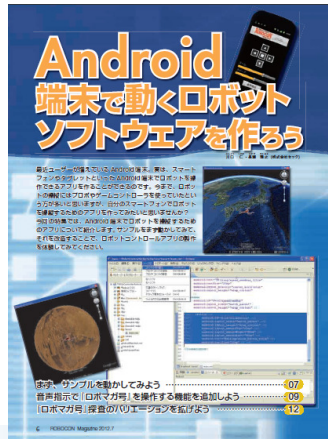
CORBA_CdrData cdr = new CORBA_CdrData();
cdr.setData(mslzr.get());
return cdr;
}

public void demarshal(CORBA_CdrData
    cdrData) {
    Marshalizer mslzr = new
        Marshalizer(cdrData.getData());
    int sec = mslzr.demarshalLong();
    int nsec = mslzr.demarshalLong();
    tm.set(sec, nsec);

    width = mslzr.demarshalShort();
    height = mslzr.demarshalShort();
    bpp = mslzr.demarshalShort();
    mslzr.demarshalShort(0);
    format = mslzr.demarshalString();
    fDiv = mslzr.demarshalDouble();
    pixels = mslzr.demarshalByteSeq();
}
}
```

## デモアプリ紹介

- **ロボコンマガジン(オーム社) 2012年7月号特集のアプリを紹介します。** <http://www.ohmsha.co.jp/robocon/>



上記オーム社サイトより  
ダウンロード可能

## 機能安全対応RTミドルウェア



RTMSafety

- **機能安全に対応した高信頼RTミドルウェアの必要性**
  - 生活支援分野などサービスロボットにおける機能安全の必要性
  - IEC61508、ISO26262、ISO13482など国際標準規格への対応
  - 機能安全対応のためのコストダウン、安全モジュールの再利用による開発効率の向上
  - 商用で利用できる高品質なRTミドルウェアへのニーズ



- **IEC61508の製品認証** 準拠の高信頼RTミドルウェアを開発

## 安全性の確保にはコストがかかる！

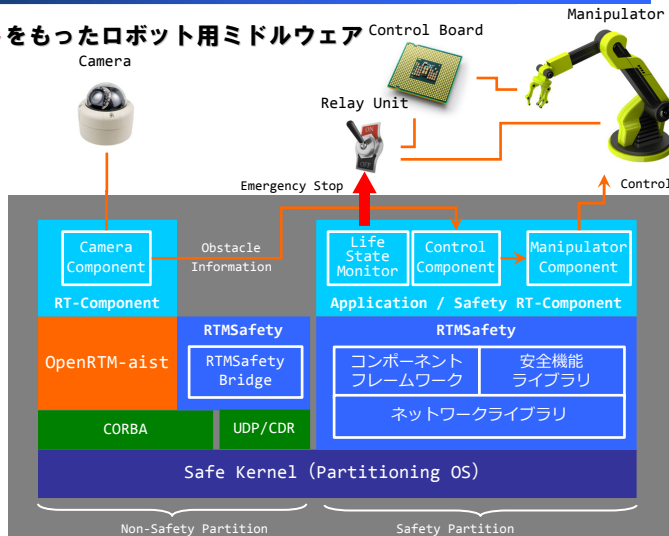
- 品質を確保するにはコストがかかる
- 同様に、安全なロボットを開発するためにはコストがかかる！
- 安全を確保しつつ、ロボットを低コストで開発するためには、安全性確保のためのフレームワークが必要

## RTMSafety



### 世界初の安全コンセプトをもったロボット用ミドルウェア

- ◆IEC61508 SIL3 Capableの製品認証を取得
- ◆ロボット用コンポーネント(RTC)とCPU負荷を均一化するフレームワークを提供
- ◆RTCの生存状況を監視する機能(Safety Function Library)を装備
- ◆GIOP / CDR準拠の軽量通信プロトコルを実装し、様々なネットワークプロトコルに対応可能(Network Protocol)
- ◆OpenRTM-aistと連携する機能を搭載(RTM Safety Bridge)



# RTMSafety



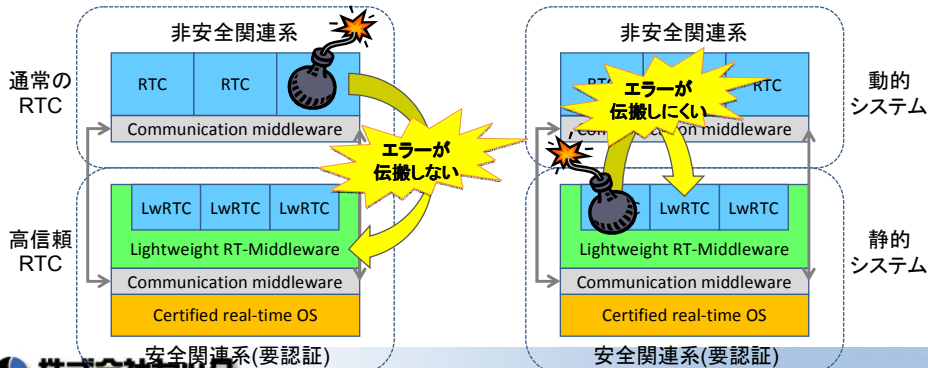
- 認証機関exidaにてIEC 61508の製品認証(SIL3)を取得
  - 日本のソフトウェア会社としては、IEC 61508の認証取得は**2例目**
- 2012年5月より**製品として販売開始**



# RTMSafetyの構成とメリット

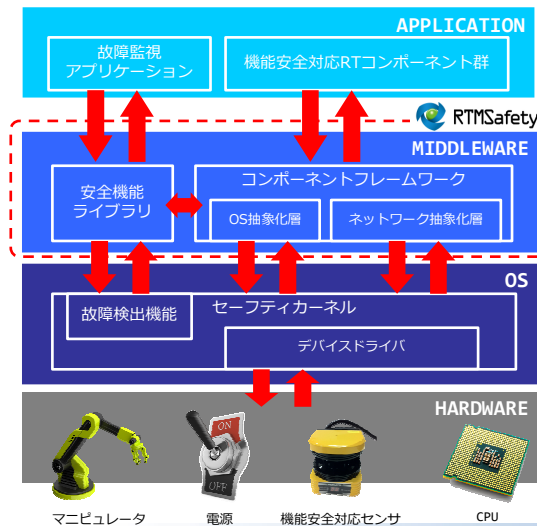


- |   |  |
|---|--|
| <p>非安全関連系が安全関連系に影響を及ぼさない</p> <ul style="list-style-type: none"> <li>• 非安全関連系が認証不要になる</li> </ul> | <p>安全関連系のコンポーネント間の影響が最小限</p> <ul style="list-style-type: none"> <li>• コンポーネント変更時の認証負荷を減らす</li> </ul> |
|---|--|





# RTMSafetyの構成とメリット



•機能安全対応のRTコンポーネントを再利用することで開発を効率化

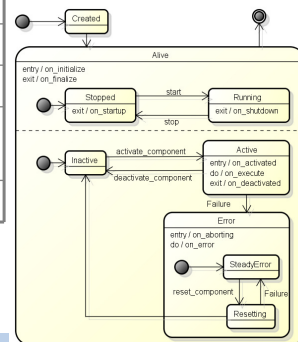
ミドルウェア層まで機能安全に対応していると...

•汎用的なアプリケーションフレームワークや共通の安全機能を利用することで開発効率の向上が見込める  
•新規開発部分を少なくすることで、安全の積み上げが容易になり、機能安全認証にかかるコストも低減できる

# OpenRTM-aistとの比較



	OpenRTM-aist	RTMSafety
Component Model	Base on OMG RTC Specification	Base on OMG RTC Specification (Light Weight RTC)
Data Port	○	○
Service Port	○	×
Configuration	○	×
Composite Component	○	×
Language	C++, Python, Java	C (MISRA C)



# RTMSafetyデータシート



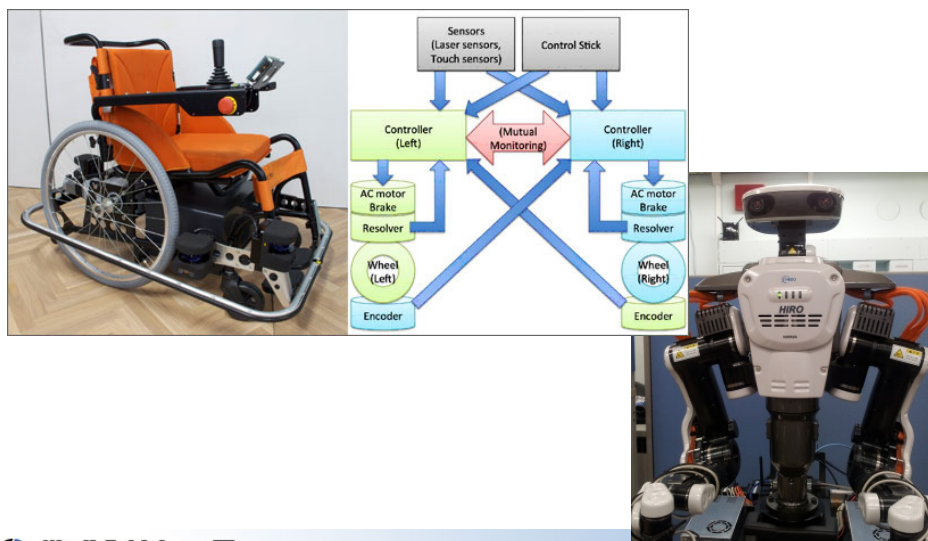
OS	QNX Neutrino RTOS Safe Kernel 1.0 TOPPERS / ASP 1.3.1 VxWorks Cert Platform (対応を検討中) ※他のOSへの移植も承ります。別途ご相談ください。
CPU	x86 family SH2 family ※他のCPUに移植することも可能です。別途ご相談ください。
ライセンス	開発ライセンス ランタイムライセンス ※価格は別途お問い合わせください。



Copyright© 2012 Systems Engineering Consultants Co., LTD. All rights reserved.

51

# RTMSafety適用事例

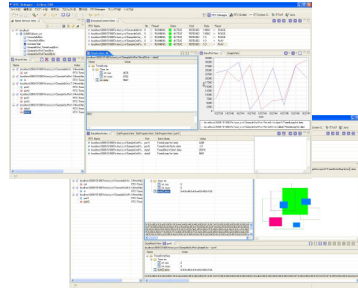


Copyright© 2012 Systems Engineering Consultants Co., LTD. All rights reserved.

52

## RTコンポーネントデバッガ

RTコンポーネントデバッガは、RTコンポーネントの動作を検証するためのテストツールです。RTコンポーネントデバッガを使うと、RTコンポーネントのテストを効率的におこなうことができます。

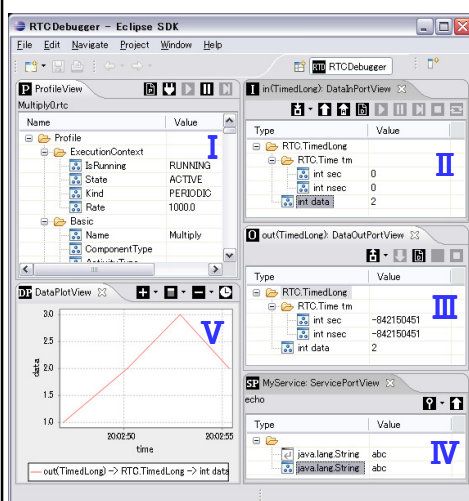


**RTコンポーネントデバッガの機能**

- ▶ RTコンポーネントの状態制御
- ▶ データポートへのデータ入出力
- ▶ サービスポートの呼び出し
- ▶ データの視覚化
- ▶ データの記録/再生 etc...

RTコンポーネントデバッガの動作環境	
OS	Javaの動作する各種OS
Java	JDK 6 Update 10以降
Eclipse	Eclipse 3.2以降
RTC仕様	OMG RTC Specification 1.0
RTミドルウェア	OpenRTM-aist-0.4.2/1.0.0/1.1.0 OpenRTM.NET-0.4/1.0/1.1

## RTCデバッガ オーバービュー



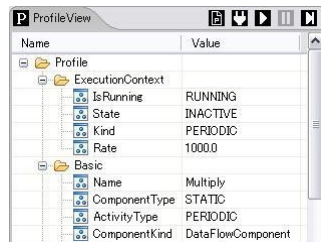
Eclipse Plug-in  
として開発

- I プロファイルビュー
- II データ入力ポートビュー
- III データ出力ポートビュー
- IV サービスポートビュー
- V データプロットビュー

## 実行コンテキスト制御機能

### ■ ExecutionContextの制御

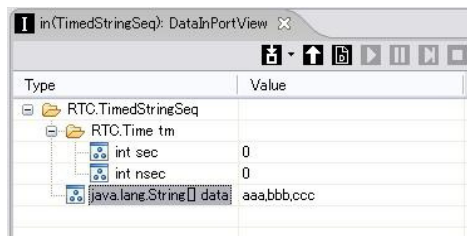
- 一時停止、ステップ実行(OnExecute1回実行)



プロファイルビュー

## データ入力ポート検証機能

- 画面で入力したデータを送信
- MIN/MAX/インターバルを指定しての自動送信

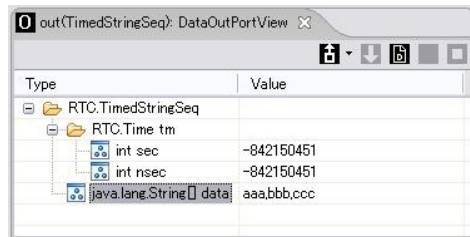


データ入力ポートビュー

TimedStringSeq型データの送信

## データ出力ポート検証機能

- RTCデバッガで受信したデータをリアルタイムに表示



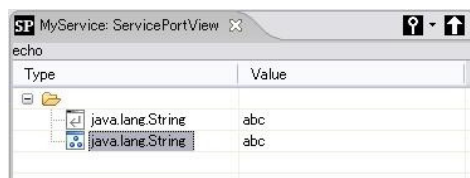
Type	Value
RTC.TimedStringSeq	
RTC.Time tm	
int sec	-842150451
int nsec	-842150451
java.lang.String[] data	aaa,bbb,ccc

データ出力ポートビュー

TimedStringSeq型データの受信

## サービスポート検証機能

- サービスポート(Provider)のインタフェースを、画面で引数を入力して呼び出し
- 戻り値を表示



Type	Value
java.lang.String	abc
java.lang.String	abc

サービスポートビュー

string echo(in string msg)の呼出

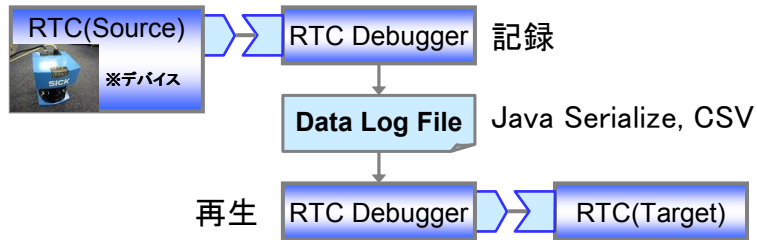
## データ記録/再生機能

通常



一般的に、再現は容易ではない

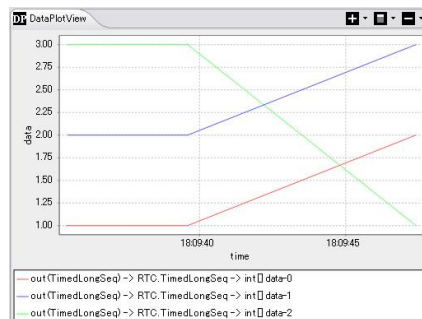
RTCデバッガを利用



再現性の確保、ハード使用面の省力化

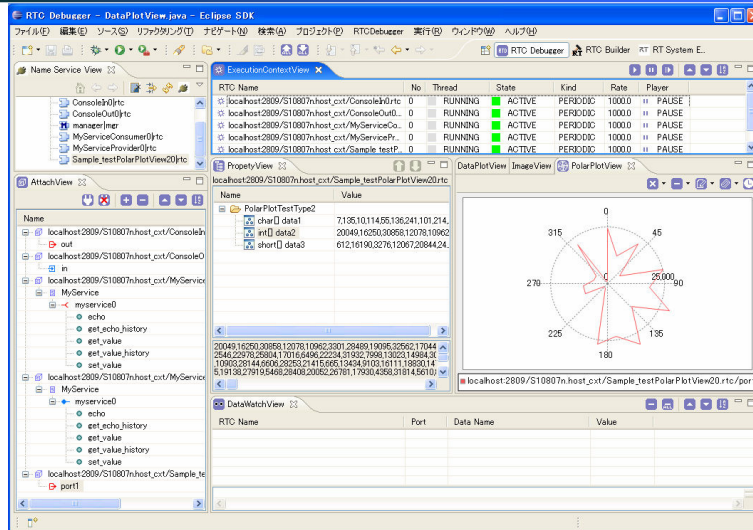
## データプロット機能

- 縦軸(データ) : Auto or 固定
- 横軸(時間) : ユーザ指定(最新 X ミリ秒間)



データプロットビュー

# データプロット機能



# ロボットサイト

- ロボットサイトにて、NEDO「次世代ロボット知能化技術開発プロジェクト」をはじめ、当社の研究開発成果を公開しています。

