RTミドルウェアサマーキャンプ2014

RTM on Androidの紹介

2014年8月5日

開発本部 第四開発部

中本啓之

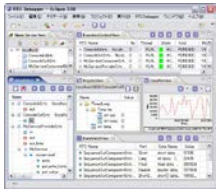
株式会社セック

Systems Engineering Consultants Co.,Ltd.

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

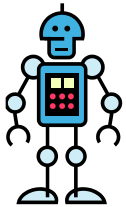
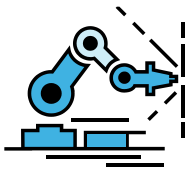



PC/タブレット向け

- OpenRTM.NET
- Android版RTミドルウェア RTM on Android™

ロボット層

- OpenRTM-aist
- HONDA RTM

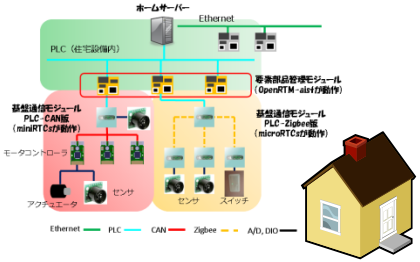

ロボット制御向け

- 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

- RTC-lite





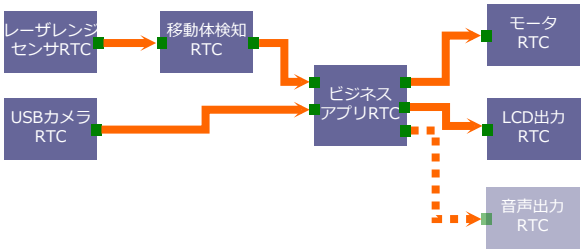
省資源マイコン向け

- miniRTCs-CAN
- microRTCs-Zigbee

RTミドルウェア導入のメリット

- **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

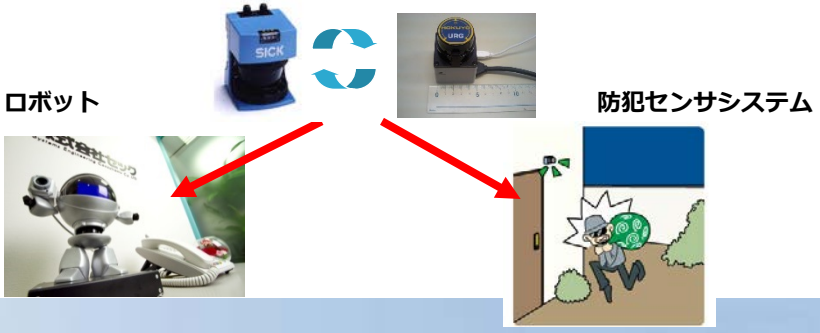
- **柔軟性の向上**
  - モジュール接続構成を変えるだけで様々なシステムを構築できる



- **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

- **再利用性の向上、選択肢の多様化**
  - 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
  - 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が容易（ソフト変更不要）



RTM on Android 開発の狙い

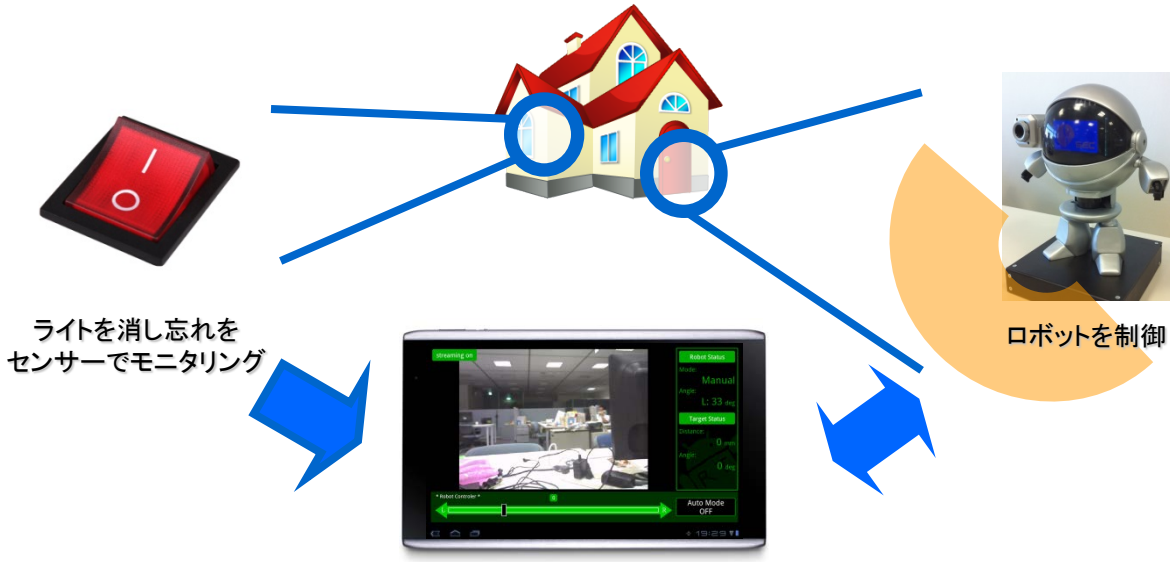
RTC開発者／利用者の裾野拡大！！

これに伴うRTミドルウェアの普及促進！！

RTM on Androidとは

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。



Android端末をロボットシステムのフロントエンドに活用！

RTM on Androidとは

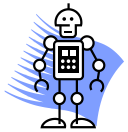
RTM on Android　とは

- ✓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



RTM on Androidの活用事例

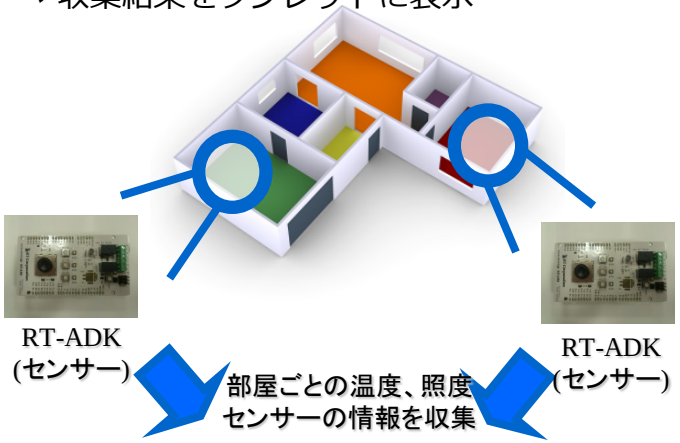
SCENE1: ロボットの遠隔制御・監視

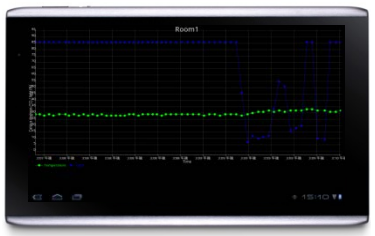
Android端末を用いてロボットを遠隔制御
- ✓タブレットからロボットを操作
- ✓ロボットのカメラ映像をタブレットで表示

コントローラとして利用

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





情報表示端末として利用

RT×Androidでロボット／センサーを制御・監視

AndroidにRTMを載せるとは？

- そもそもRTMとは
  - RTCを作成するために
  - RTCがRTCとして動作するために
  - 存在するミドルウェア
  - ．．．としてのソフトウェア
  - 実行単位はRTMではなくRTC
- Android上でRTCを作成する？
  - そのようなシーンは想像できない
- Android上で動作するRTCとは？
  - 単なるAndroidアプリの１形態に過ぎない

Androidアプリとして
RTCを作成し

Android端末上で
上記RTCが動作する

ことを可能にする
こと

RTM on Android の効能

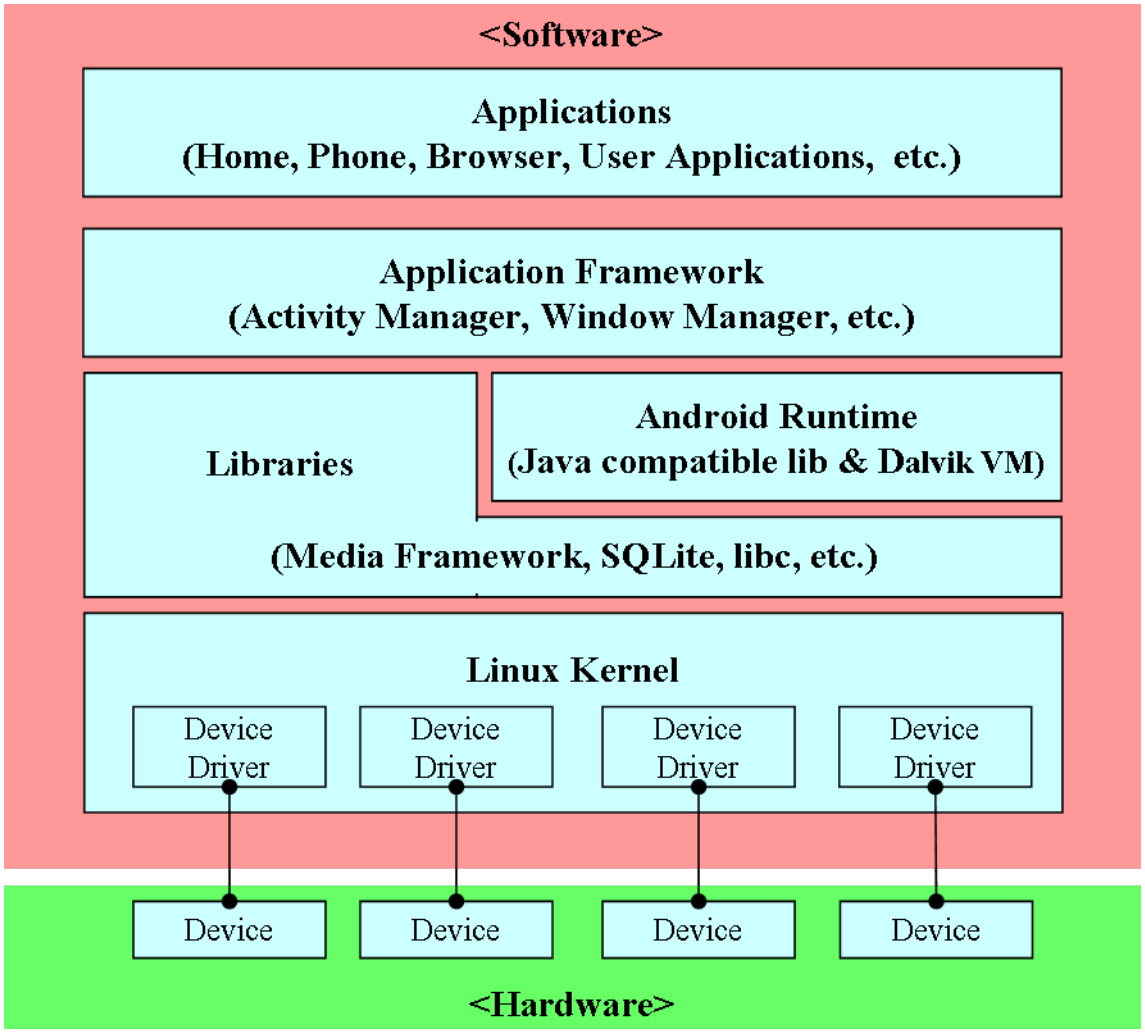
- RTミドルウェアやRTCの知識が乏しくても、通常の**Androidアプリケーション開発スキルを持つ人であれば、容易にRTCの開発が可能**となる
- RTM on Androidを利用して開発したRTCは、ごく普通の**Androidアプリケーションとして扱うことができ、実行時のシステムリソースへの負荷も軽い**
- RTM on Androidを利用して開発したRTCは、OMGにより標準化されたRTC標準仕様Ver1.0に従った軽量RTCのOpenRTM拡張モデルとしての基本的な振る舞いに対応し、**他プラットフォーム上のRTCともシームレスに相互接続**できる

RTCとAndroidにおけるコンポーネントの対応

- RTCは，必ずしもUIを必要としない
- 他RTCとの通信は継続的に実施できる必要がある

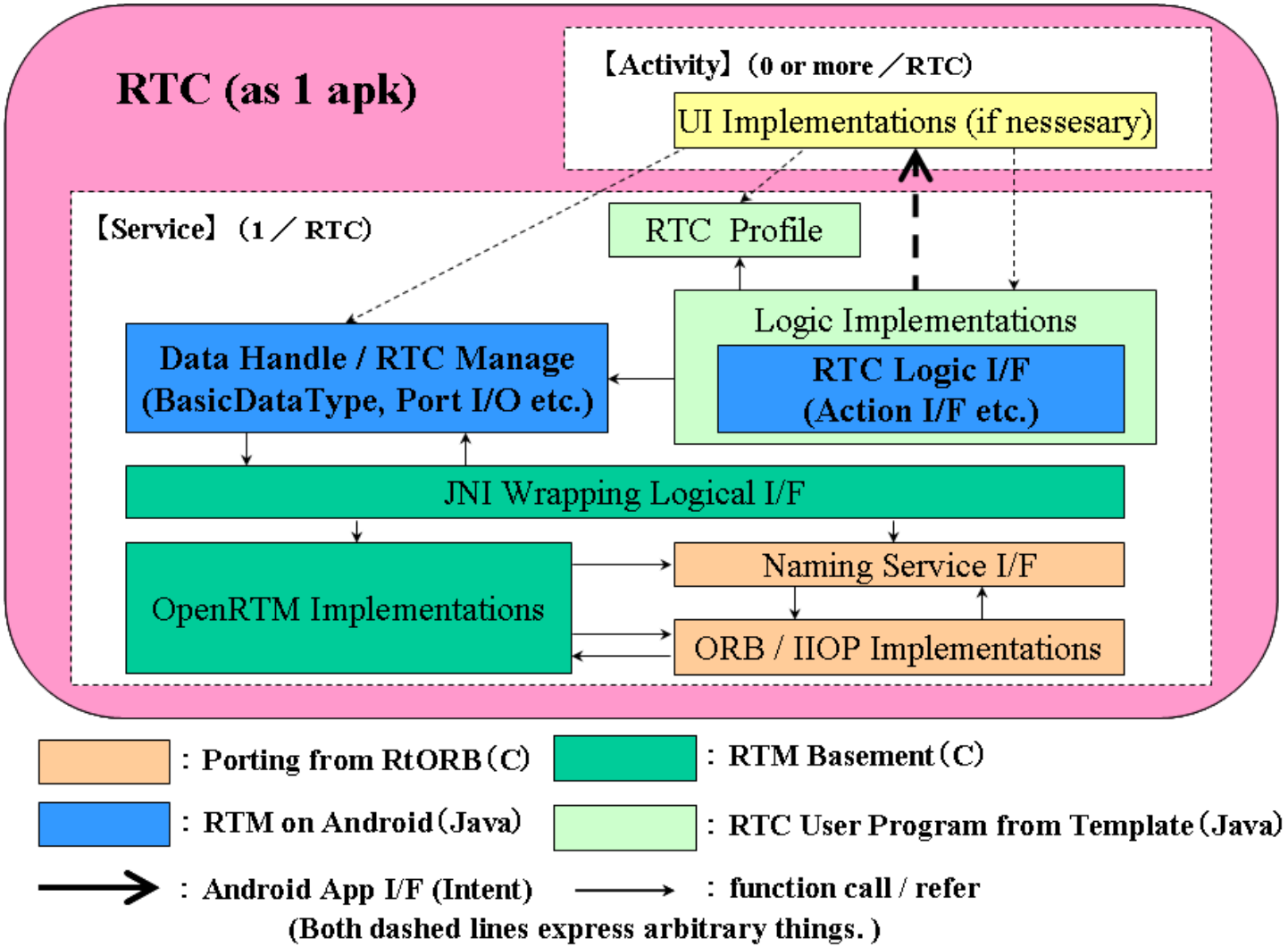
- RTM on Androidでは，RTCがAndroid上でServiceとして動作
  - UIが必要な場合は別途Activityを必要なだけ追加
  - これら全体で一つのアプリケーション単位（apk）
  - 一つのAndroid端末上にて 複数のRTCを同時にActiveな状態で稼動可能

Androidアーキテクチャとアプリケーション



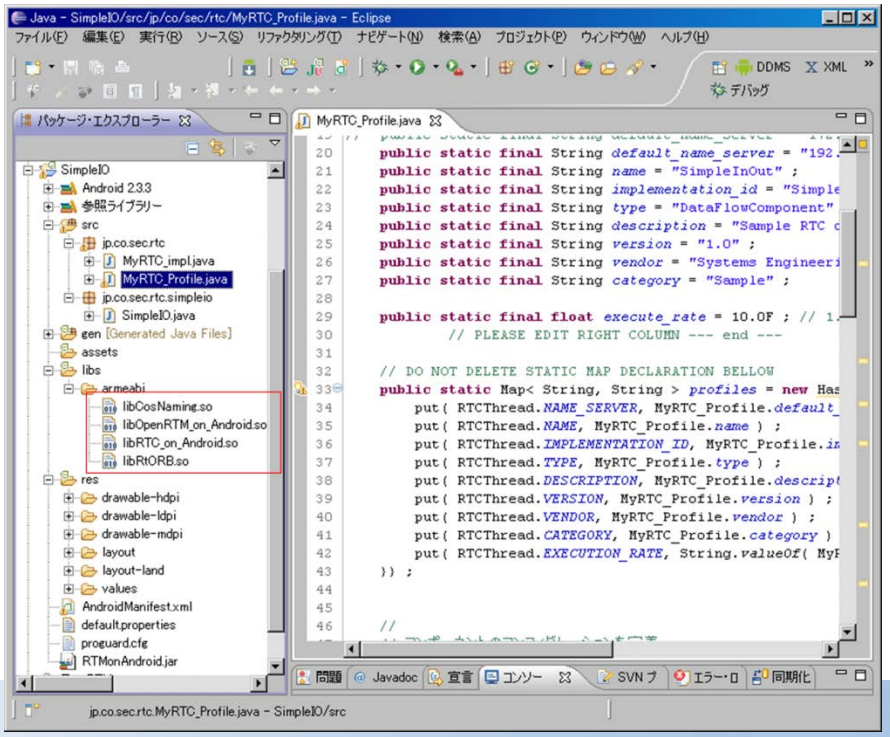
- Android Runtimeに専用VM（Dalvik VM）
  - 各アプリケーションは，それぞれが独立したLinuxのプロセス
  - 各アプリケーションに一つのDalvik VMインスタンスが対応して動作
- 一般的Androidアプリケーションは，UIを持つコンポーネントであるActivityとして実装される
  - Activityは画面の最上位に表示される場合にのみ動作
- UIを持たず，バックグラウンドで継続的に動作するServiceも代表的なコンポーネント単位

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



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
    - 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
    - Rtshellを使うか、Connectorを作成する

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.openrtp.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>();
  }

  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();
    format = mslzr.demarshalString();
    fDiv = mslzr.demarshalDouble();
    pixels = mslzr.demarshalByteSeq();
  }
}
```

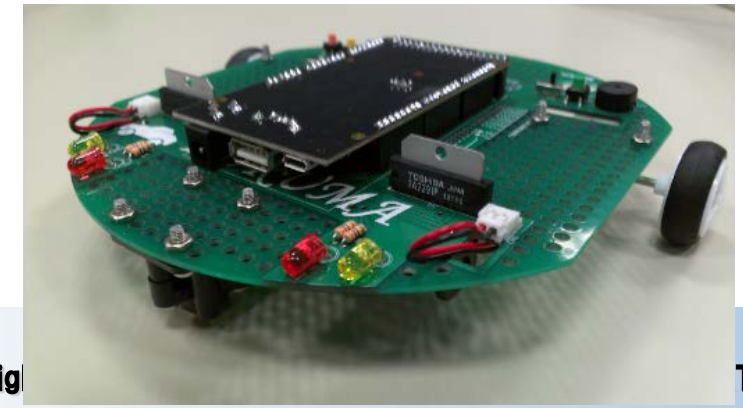
ロボコンマガジン連載

- ロボコンマガジン（オーム社）で、2012年7月号～2013年1月号に「RTM on Android」の連載記事

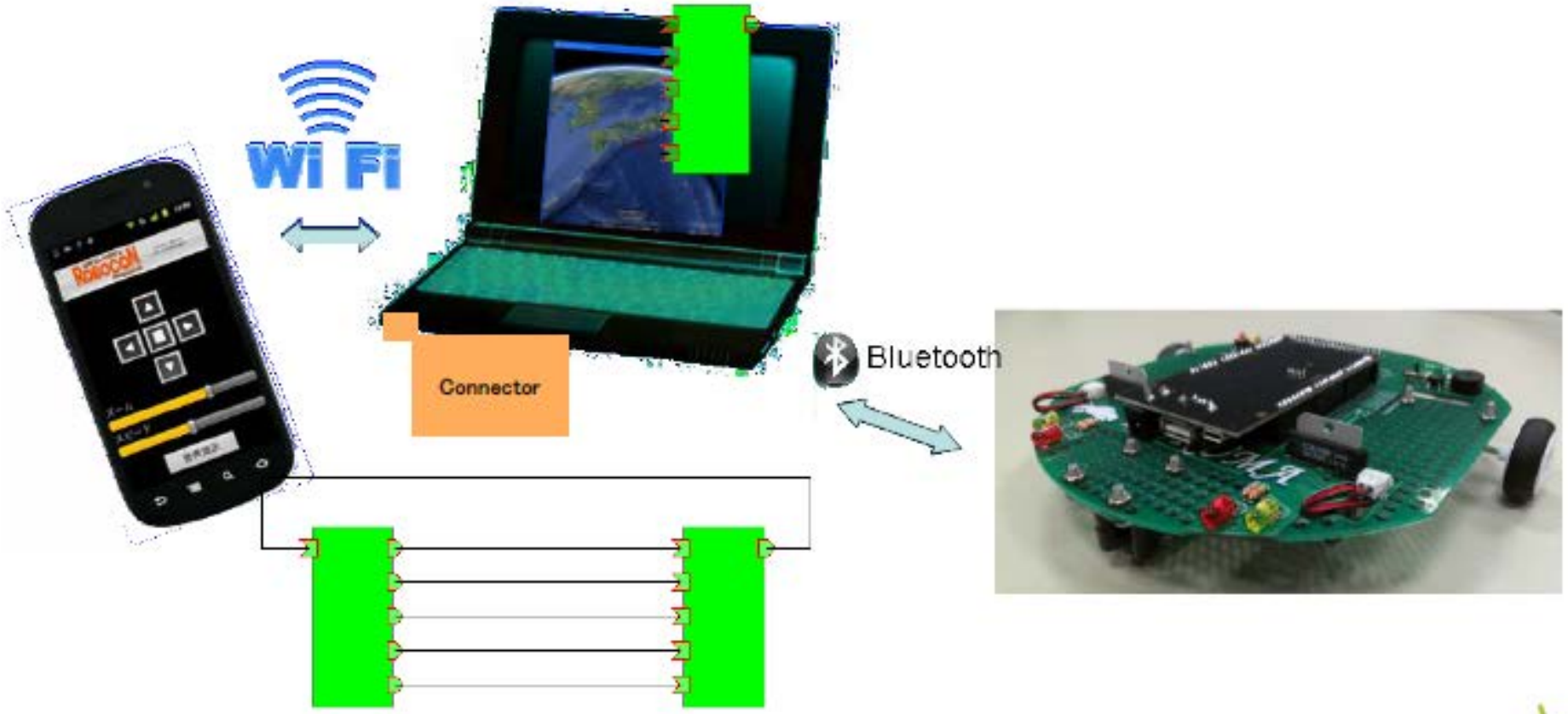
http://www.ohmsha.co.jp/robocon/

ロボコンマガジン連載

- OpenRTM.NETで実装したWindows PC上で動作するRTCをAndroid端末からコントロールするRTCの開発
  - Google Earth内を移動するバーチャル飛行体
  - Arduino用移動台車 KURUMA Shield（ピルクス社）



デモシステム構成



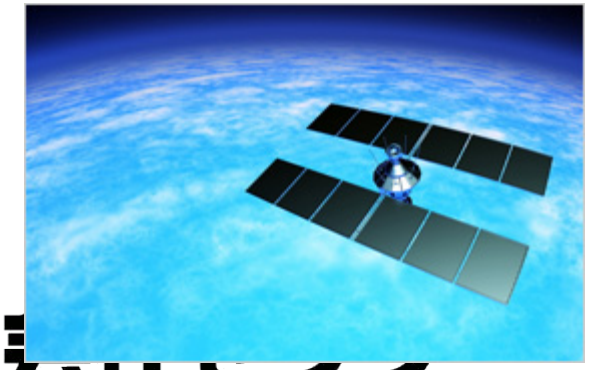
KURUMA Shield コントロール時の接続

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。

e-mail ： robot@sec.co.jp
nakamoto@sec.co.jp

QCD & I = Customer Satisfaction



株式会社セック

Systems Engineering Consultants Co.,Ltd.