

第23回 マイクロマシン/MEMS展
第3回 ROBOTECH 次世代ロボット製造技術展

ROBOTECH RTミドルウェア講習会
RTミドルウェアAndroidワークショップ

~ RTM on Android を利用してAndroidでRTCを動かそう ~



2012年 7月13日

株式会社セック
開発本部 第四開発部
川口 仁、池添明宏

アジェンダ

- 準備状況などの確認
- RTM on AndroidのサンプルRTC開発環境展開
- RTM on Androidの概要説明
 - アーキテクチャ
 - RTC開発方法概要(サンプルRTCコードを参考に)
 - 使用上の注意点 など
- RTM on AndroidのサンプルRTCを動かそう
 - ビルド・インストール
 - OpenRTM-aistサンプルコンポーネントとの通信
- 機能追加演習
 - 入力ポート追加、新規データ入力処理追加
- 余興(ちょっとしたデモアプリ紹介)
- 質疑



準備状況などの確認

■ 機材の準備

■ ノートPC

- OS: Windows XP (Vista, 7も可)
- Eclipseが動作する程度のスペックが必要です
- 無線LANが使用できること
- USBポートの空きが1つ以上あること

■ Android端末 (2.3.3 ~ 3.2)

■ PC-Android端末接続用USBケーブル

■ ワイヤレスネットワーク接続確認

■ アクセスポイント名

■ パスワード



準備状況などの確認

■ ソフトウェアの準備

■ OpenRTM-aist(C++) 1.0.2-RELEASE

- omni-ORB, RT SytemEditor, サンプルRTCを含む

■ Androidアプリ開発環境

- JDK, Android SDK (Platform2.3.3), Eclipse[Javaの開発環境を含む]

■ Android端末接続用USBドライバ

- ADBコマンドが使えるようにしておく

■ RTM on Android

- RTMonAndroid-1.0.1.zipのダウンロード



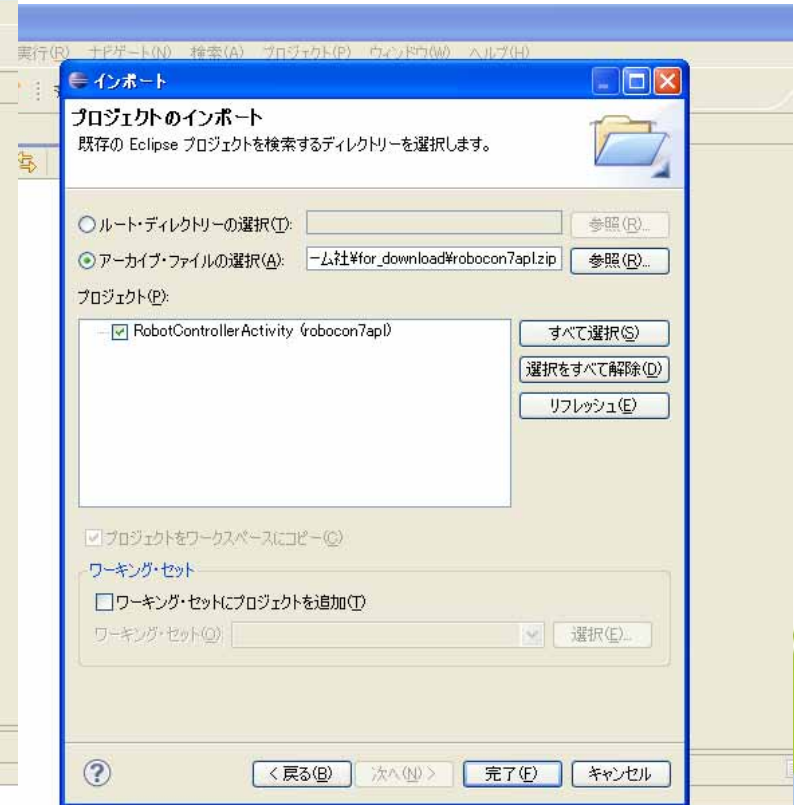
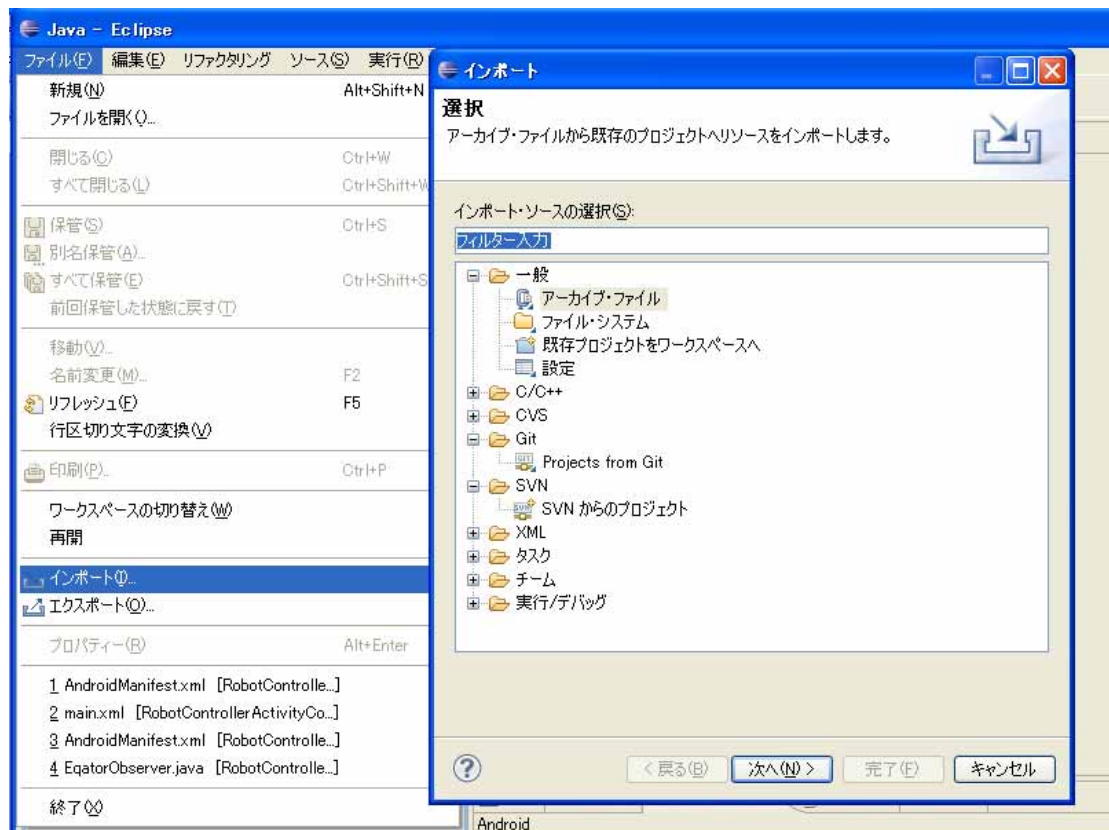
RTM on AndroidのサンプルRTC開発環境展開

■ Eclipseを起動

Eclipseの文字コードは、UTF-8としておいて下さい。

■ インポート

既存プロジェクトをワークスペースへ
アーカイブファイルの選択



RTM on Androidの概要説明

■ 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アプリとして
RTCを作成し

Android端末上で
上記RTCが動作する

■ Android上でRTCを作成する？

- そのようなシーンは想像できない！

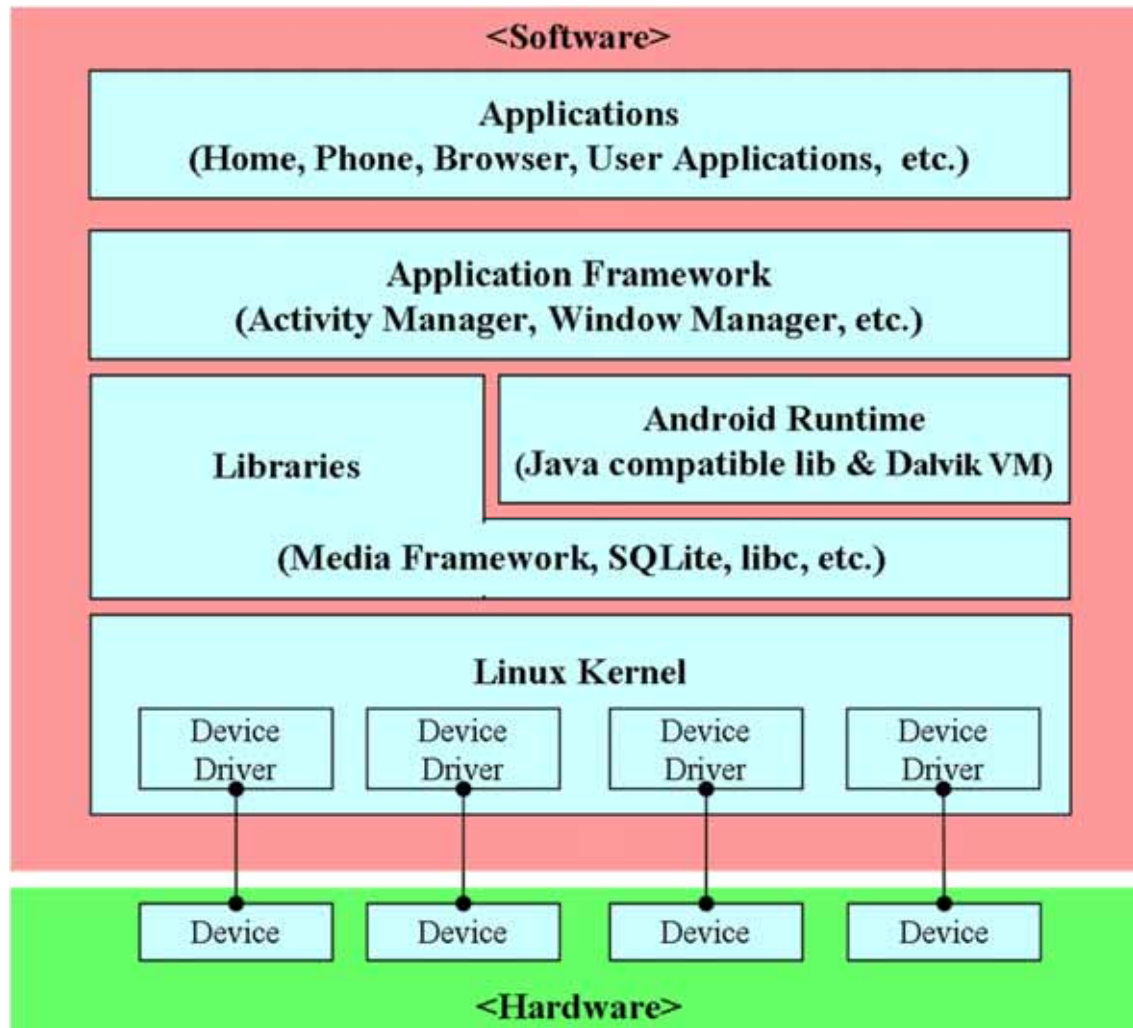
ことを可能にすること

■ 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)

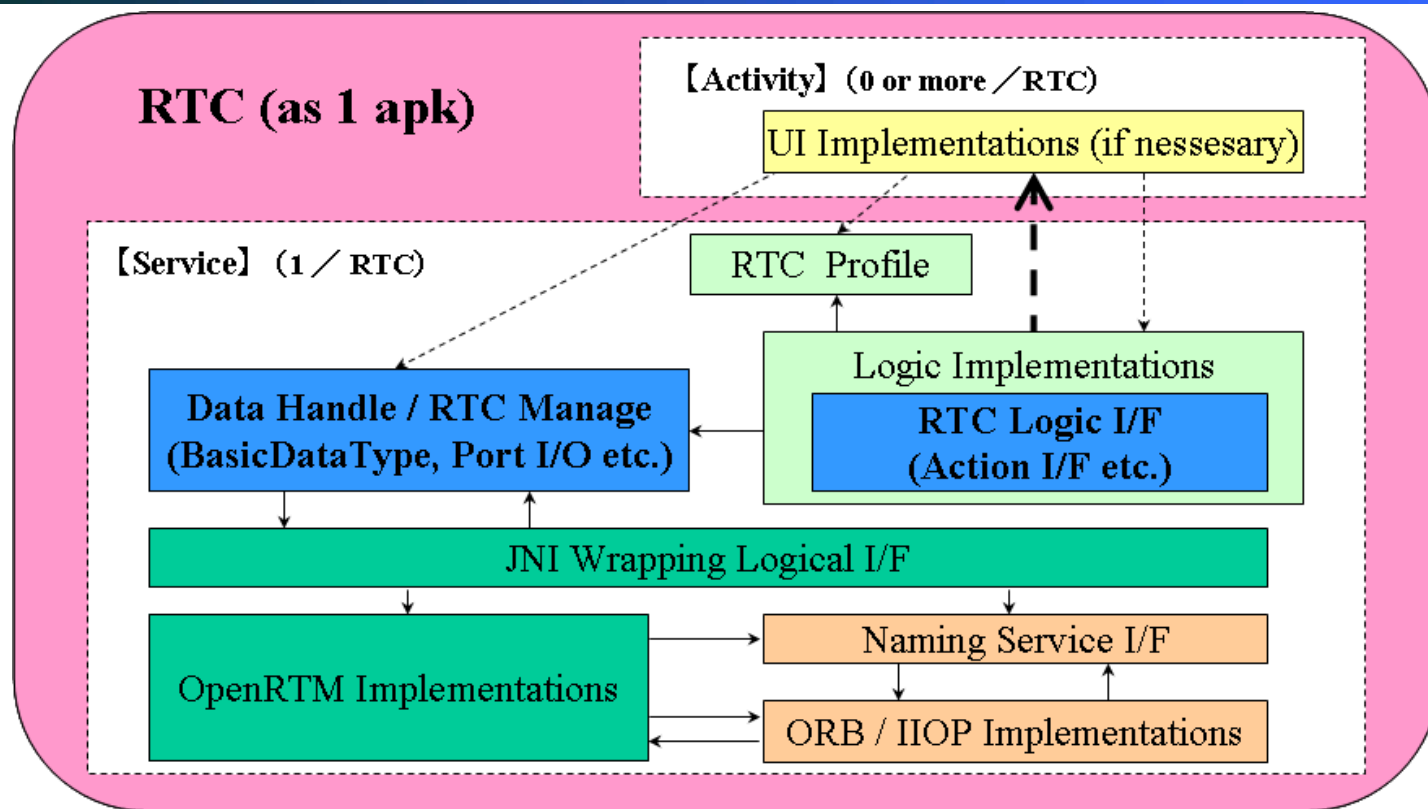
- 各アプリケーションは、それぞれが独立したLinuxのプロセス
- 各アプリケーションに一つのDalvik VMインスタンスが対応して動作

■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



- : Porting from RtORB (C)
 - : RTM Basement (C)
 - : RTM on Android (Java)
 - : RTC User Program from Template (Java)
 - : Android App I/F (Intent)
 - : function call / refer
- (Both dashed lines express arbitrary things.)



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

A screenshot of the Eclipse IDE showing the source code for MyRTC_Profile.java. The code defines a class with various static final fields and a static map. The fields include default_name_server, name, implementation_id, type, description, version, vendor, and category. The static map 'profiles' is populated with values for these fields. The code is written in Java and includes comments for configuration and execution rate.

```
20 public static final String default_name_server = "192
21 public static final String name = "SimpleInOut" ;
22 public static final String implementation_id = "Simple
23 public static final String type = "DataFlowComponent"
24 public static final String description = "Sample RTC c
25 public static final String version = "1.0" ;
26 public static final String vendor = "Systems Engineeri
27 public static final String category = "Sample" ;
28
29 public static final float execute_rate = 10.0F ; // 1
30 // PLEASE EDIT RIGHT COLUMN --- end ---
31
32
33 // DO NOT DELETE STATIC MAP DECLARATION BELOW
34 public static Map< String, String > profiles = new Has
35 put( RTCThread.NAME_SERVER, MyRTC_Profile.default_
36 put( RTCThread.NAME, MyRTC_Profile.name ) ;
37 put( RTCThread.IMPLEMENTATION_ID, MyRTC_Profile.im
38 put( RTCThread.TYPE, MyRTC_Profile.type ) ;
39 put( RTCThread.DESCRPTION, MyRTC_Profile.descript
40 put( RTCThread.VERSION, MyRTC_Profile.version ) ;
41 put( RTCThread.VENDOR, MyRTC_Profile.vendor ) ;
42 put( RTCThread.CATEGORY, MyRTC_Profile.category ) ;
43 put( RTCThread.EXECUTION_RATE, String.valueOf( My
44
45
46
```

Android用RTCの開発方法概要

- RTCの仕様をプロファイルソースに記述
- RTCの起動・停止周りをメインクラスなどに記述
 - ネームサーバの場所などの管理
 - サンプルでは、画面から入力
 - メニュー設定にするとAndroidアプリらしくなる
 - RTCの起動・停止タイミング等の管理
 - サンプルでは、ボタン押下で制御
 - アプリ起動・終了と連動させる方法もある

RTCとしての実装を処理実装ソースに記述

- 入出力ポートの生成登録やデータハンドリングなど

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 サンプルRTCを動かそう

■ ビルド

- Eclipseの設定が自動ビルドの場合
- 強制的にマニュアルでビルドする場合

■ インストール

- Eclipseから簡易インストール・・・本日はこれ
- 配布する場合は、必ず署名をつけてapkを作ること
 - これは、RTM on AndroidではなくAndroidアプリの作法

■ 実行

- 簡易インストール直後は直ちに起動
 - これも、普通のAndroidアプリと同じ



ConsoleIn, ConsoleOutとつながろう

- PCのIPアドレス確認
- PCでomniネーミングサービス起動
- PCでRT SystemEditor起動
- PCでConsoleIn, ConsoleOutを起動
- Android端末でサンプルアプリ (RTMonAndroid) を起動
 - NameServerのIPアドレスを設定
 - START RTCボタン押下
- RT SystemEditorでポート接続し、Activate
 - ConsoleIn RTMonAndroid ConsoleOut
 - All Activate
- ConsoleInから数値を入力
- Deactivate、停止



機能追加演習

■ 講師のAndroid端末の傾きセンサの値を取得して表示しよう

- TimedFloatSeq型の入力データポートを追加
 - 追加データポート名規定
 - 追加データポート生成、登録
- 追加データポートからのデータを取得
 - ポートからのデータ取得
 - TimedFloatSeq型からのデータを取り出し
- 取得データの表示を指示
 - 傾きセンサ値に対するキーワード規定
 - 既存の仕組みを利用した取り出しデータ表示指示

修正対象クラス

jp.co.sec.rtcパッケージ

RTMonAndroidProfile

RTMonAndroidImpl



コード修正箇所 (行番号は目安)

追加データポート名規定

- RTMonAndroidProfile -

```
26 //  
27 // コンポーネントのデータポートを定義  
28 //  
29 // << in port >>  
30 public static final String InPort1= "S_in_port_1";  
31 public static final String Inport2= "SensorIn_port";  
32
```



コード修正箇所 (行番号は目安)

追加データポート生成、登録

- RTMonAndroidImpl -

```
19 // InPort
20 private InPort<TimedLong>inPortLong;
21 private InPort<TimedFloatSeq>sensorInPort;
22
...
129 private void initInPort(){
130     String pName = RTMonAndroidProfile.InPort1;
131     TimedLong tm = new TimedLong();
132     inPortLong = new InPort<TimedLong>(pName, tm);
133     rtcService.addInPort(inPortLong);
134
135 // センサ情報入力ポート生成、登録
136 sensorInPort = new InPort<TimedFloatSeq>(
137 RTMonAndroidProfile.Inport2, new TimedFloatSeq());
138 rtcService.addInPort(sensorInPort);
139 }
```



コード修正箇所 (行番号は目安)

ポートからのデータ取得、TimedFloatSeq型からのデータを取り出し

- RTMonAndroidImpl -

```
154 private void ioControl(){
155     Logger4RTC.spec("SPEC_LOG", "RTM_11j: onExecute in Application Started.");
156     String str = "";
157     try{
158         // センサ情報入力
159         while (sensorInPort.isNew()) {
160             TimedFloatSeq tfs = sensorInPort.read();
161             str = tfs.getTm().toString() + "¥n";
162             List<Float> sensorValueList = tfs.getData();
163             int num = sensorValueList.size();
164             for (int i=0; i<num; i++){
165                 str += sensorKey[i] + ":" + sensorValueList.get(i) + "¥n";
166             }
167         }
```



コード修正箇所 (行番号は目安)

傾きセンサ値に対するキーワード規定

- RTMonAndroidImpl -

```
23 // OutPort
24 private OutPort<TimedLong>outPortLong;
25
26 // センサ情報ポート通知内容
27 privateString[]sensorKey = {"Direction","Pitch","Roll"};
28
```

既存の仕組みを利用した取り出しデータ表示指示

- RTMonAndroidImpl -

myRTC myActivityへの名称変更リファクタリング実施

```
175         str = tl.getTm().toString() + " | "+n;
176 //         myActivity.textDraw(str);// 画面表示
177     }
178 if (str.length()>0) myActivity.textDraw(str);// 画面表示
179     }
```

コメントアウト



機能追加演習

- 機能追加修正ができた方
 - ビルド
 - インストール
 - 講師のところへ
 - 講師のRTCと接続確認

修正が成功していれば、
このような表示が出ます。



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/>



講師が執筆しました

上記オーム社サイトより
ダウンロード可能



質 疑

