

ET2008 11/19/2008



Java で遊ぼう！ Sun SPOT の紹介と応用事例

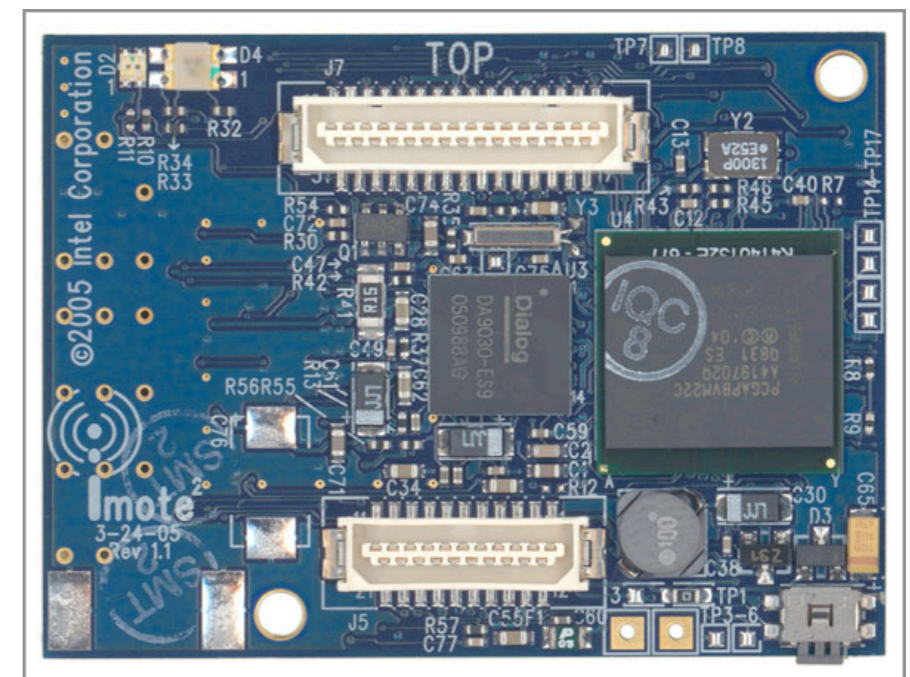
サン・マイクロシステムズ株式会社

草薙 昭彦・山口 浩

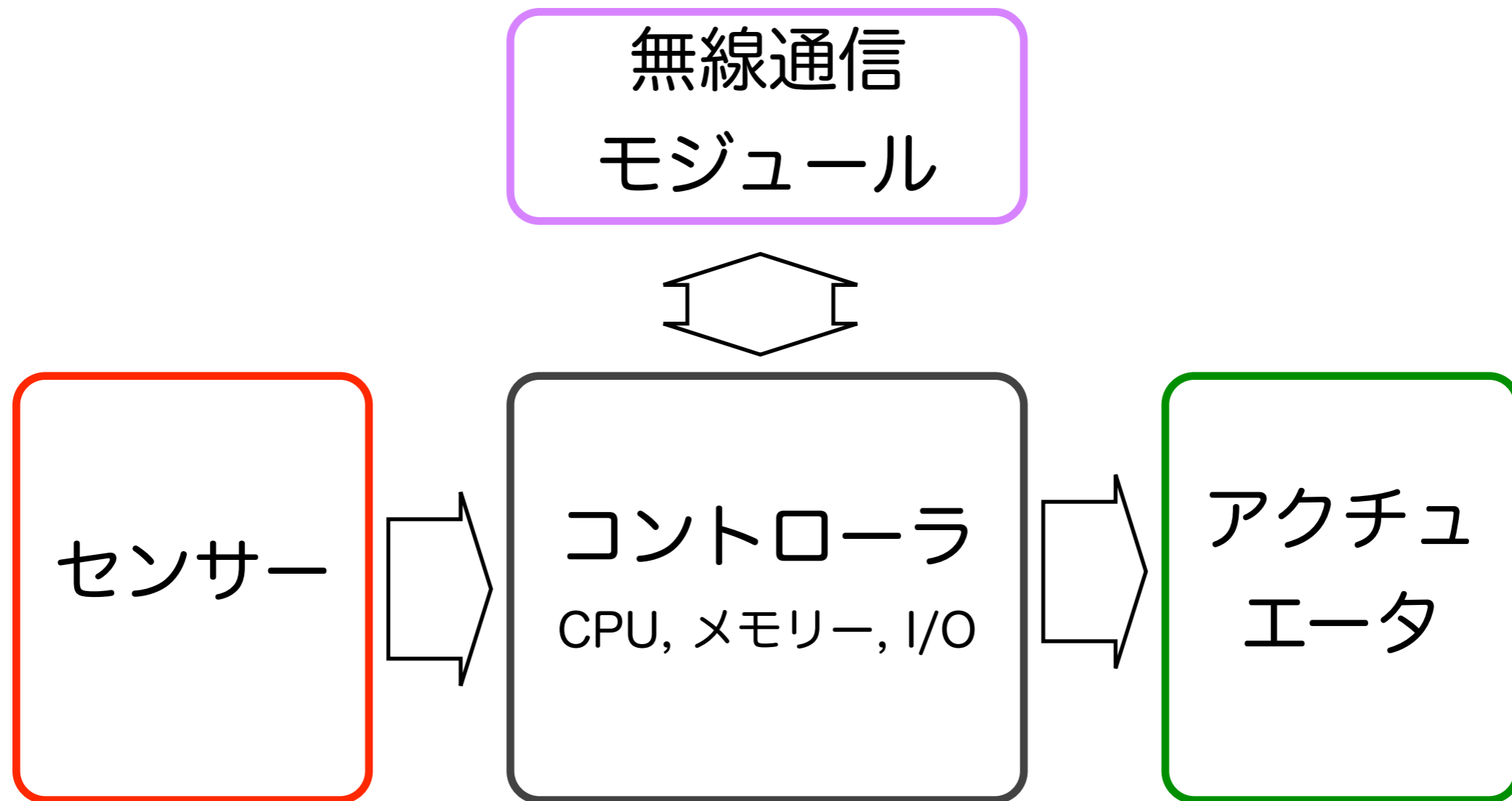


無線センサー分野の先駆者たち

- Smart Dust
 - > Kris Pister, 1998-2001
- Berkeley Motes, TinyOS
 - > Mica2 (8 bit, 4KB SRAM, 128KB Flash)
- Intel Mote
 - > Mote 2 (PXA271 Xscale)
 - > 32MB Flash & SDRAM
 - > Sensor Connectors



無線センサーデバイスの構造



アプリの開発は容易ではない

- ソフトウェアの開発環境
 - > 低レベル (C に近い) プログラミング言語
 - > 特殊で高価な開発環境
- 開発ツール
 - > 汎用ツールが利用できない
 - > 低レベルのデバッグ機能

多くの開発者には敷居が高い

Sun SPOT のアプローチ

- 小型センサーデバイス
 - > IEEE 802.15.4 無線
- Java プログラミング
 - > 豊富な API
- PC上の統合開発環境
 - > SPOT Manager & NetBeans IDE

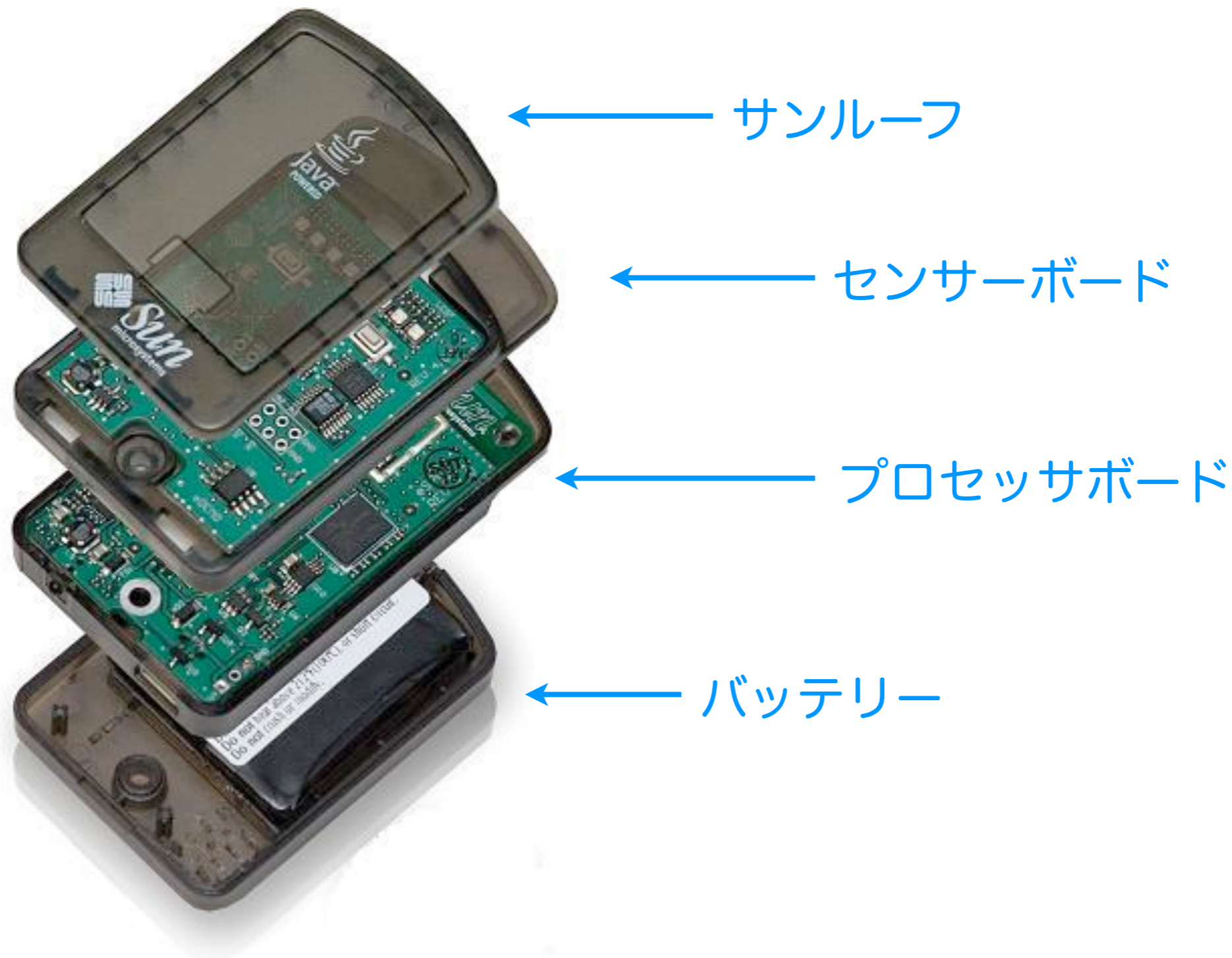


ハードウェア

Sun SPOT の外観

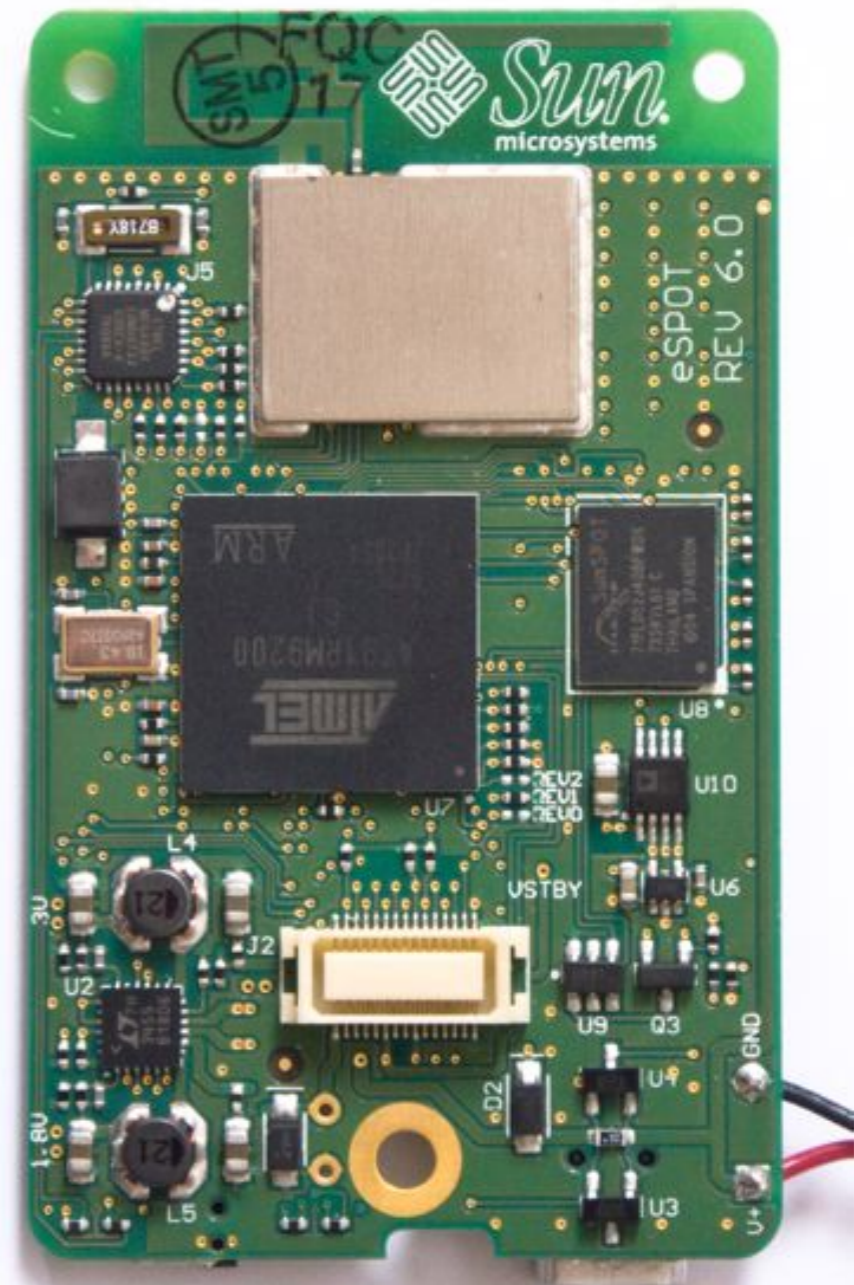


ドライバ1本で簡単に分解



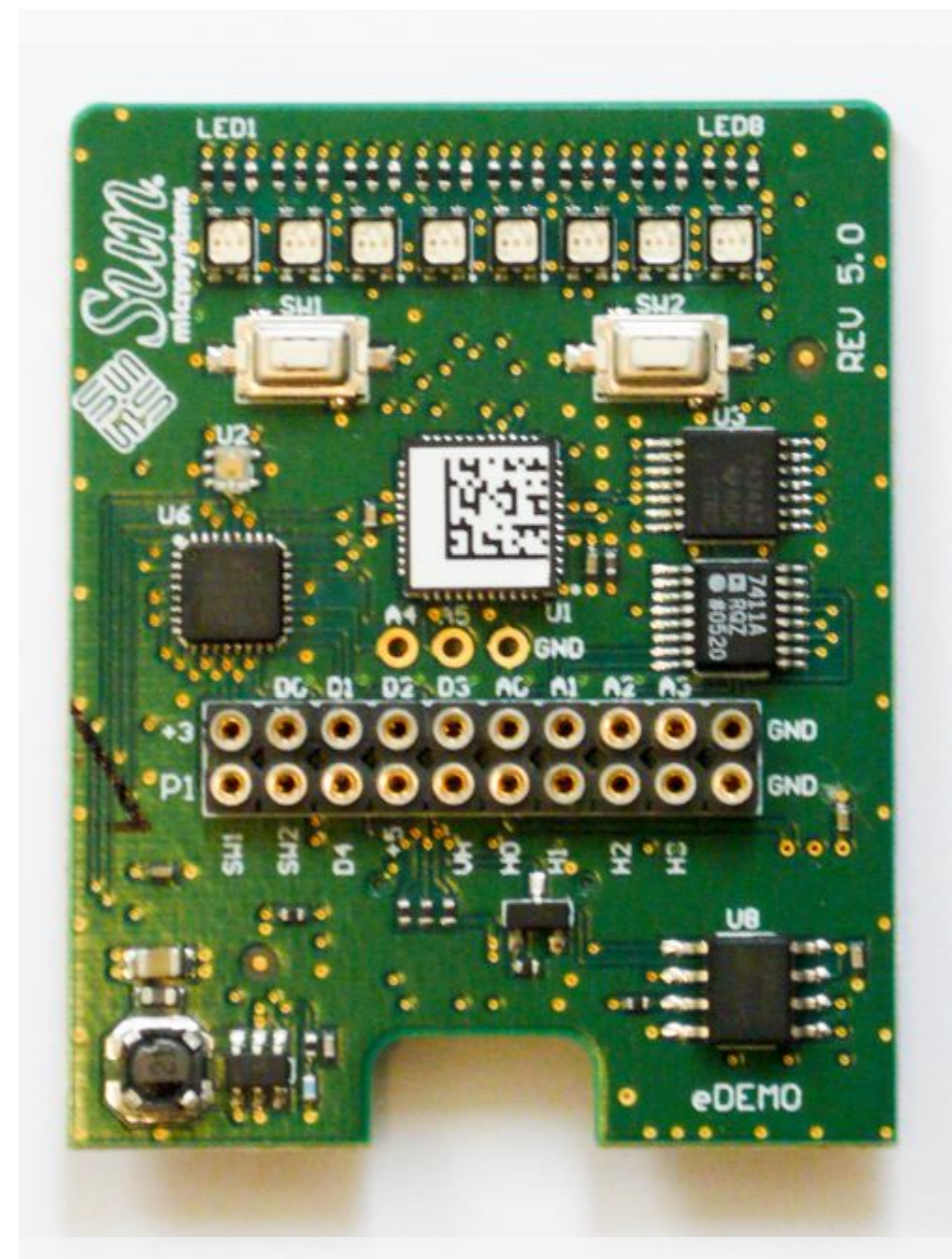
プロセッサボード

- 180MHz ARM-920T
- 512 KB RAM
- 4MB フラッシュ
- TI CC2420 Radio
 - > IEEE 802.15.4/2.4GHz
- LEDs
- センサーボードコネクタ



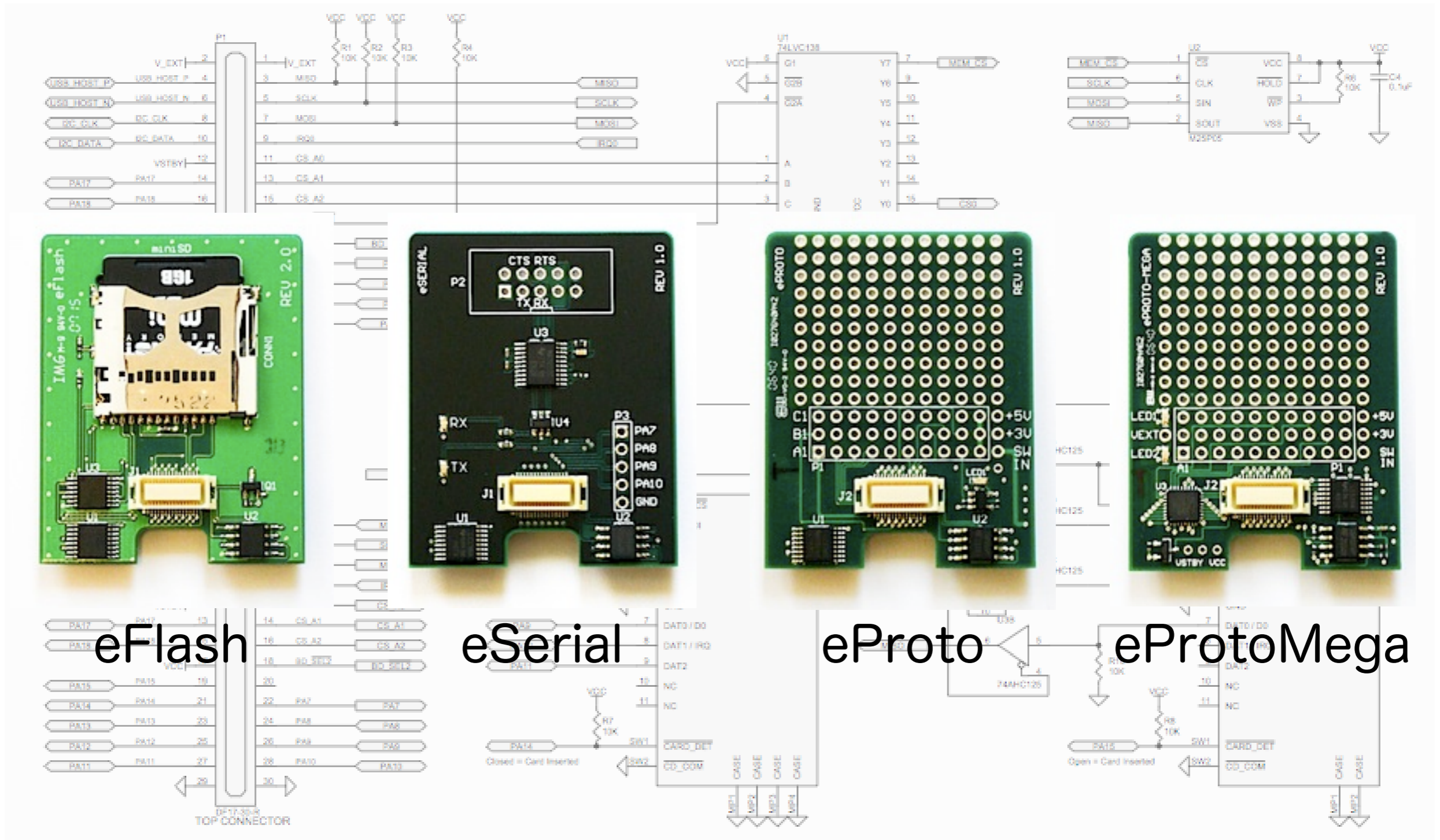
センサーボード (eDEMO)

- 3軸加速度センサー
- 光センサー
- 温度センサー
- 3色 LEDs (8個)
- プッシュボタン (入力)
- I/O ピン



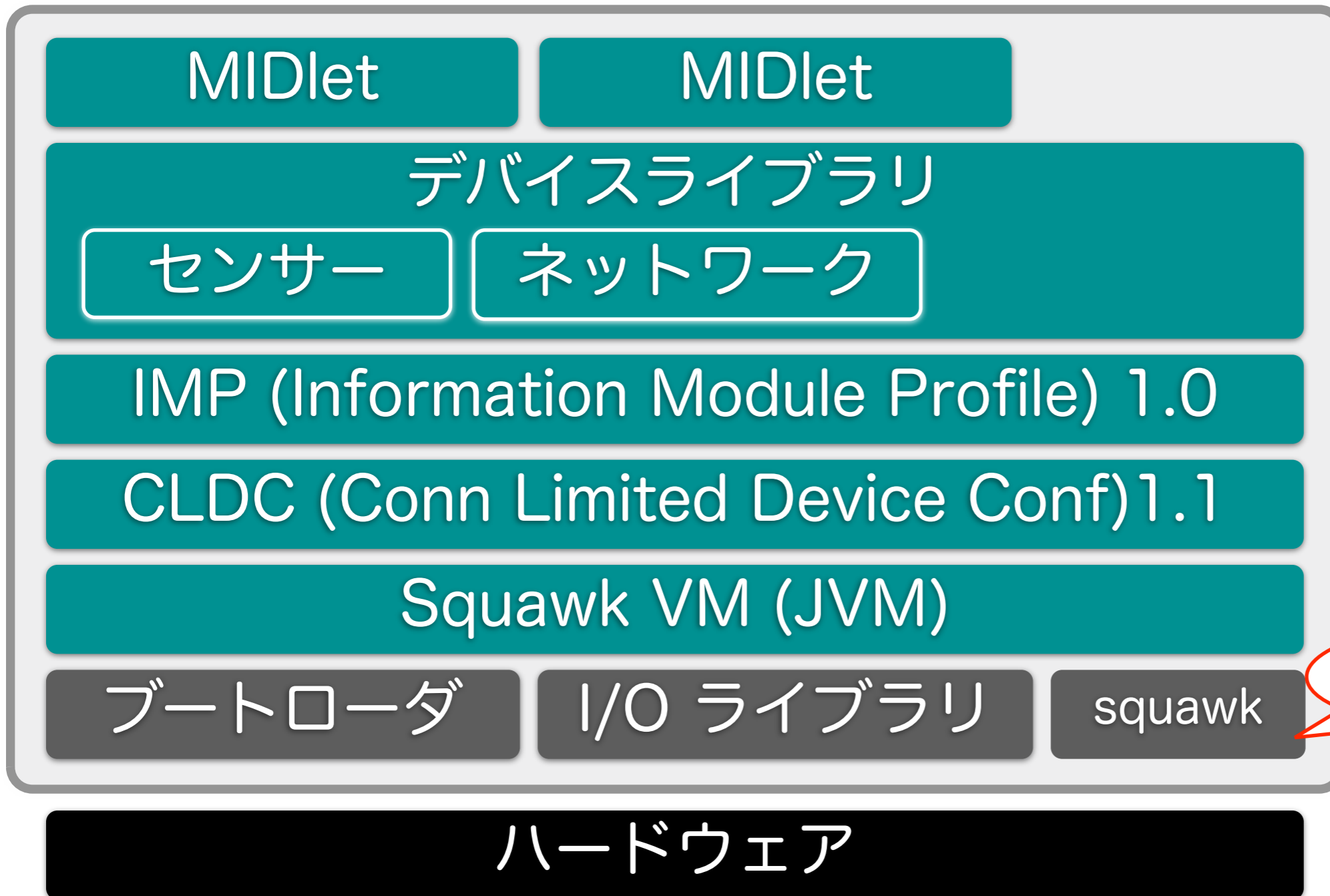
> アナログ入力(6), 汎用入出力(5), 高出力(4)

アドオンボード (プロト)



ソフトウェア

ソフトウェアスタック



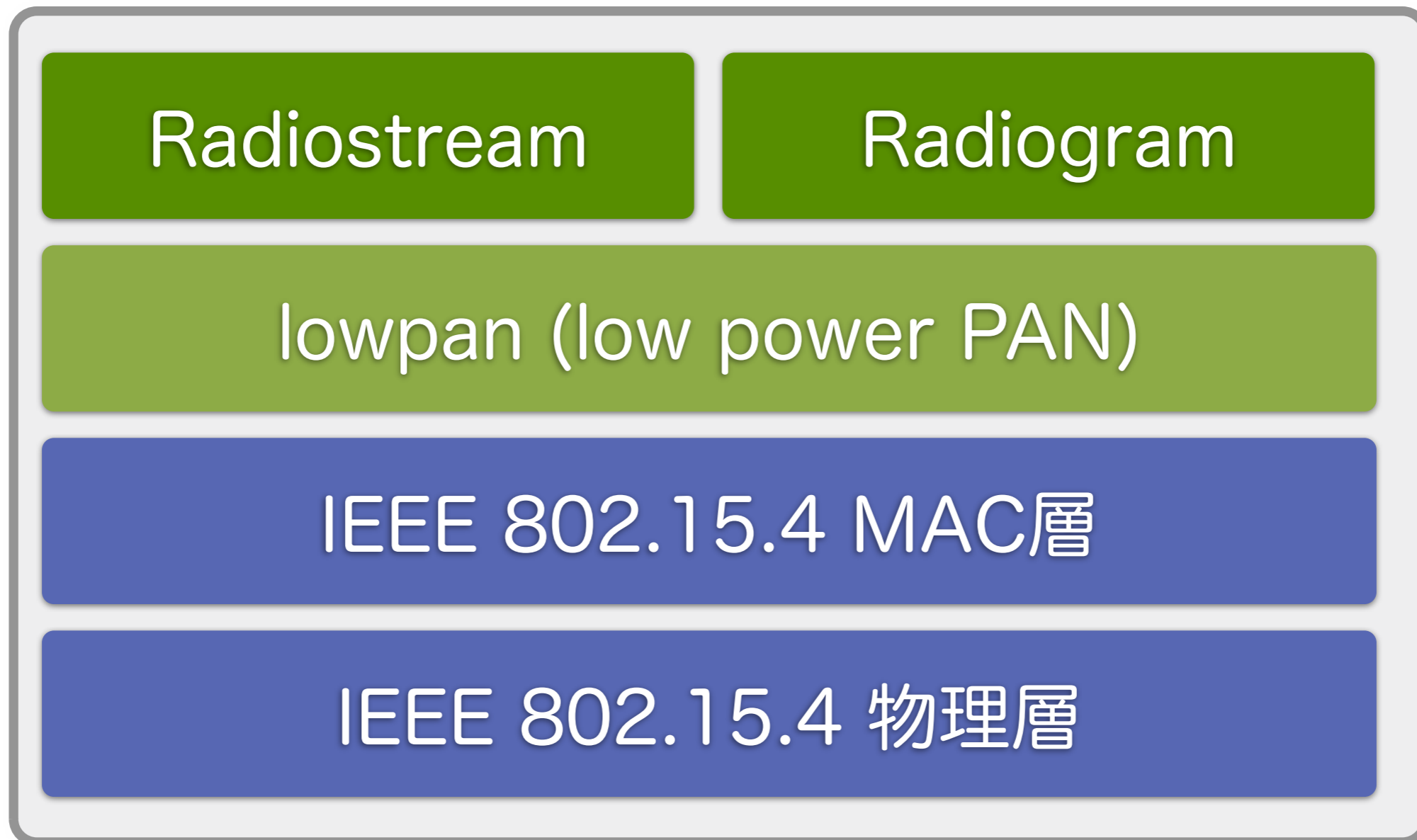
Java
アプリケーション

Javaクラス
ライブラリ

ハードウェア依存

■ C

無線のプロトコルスタック

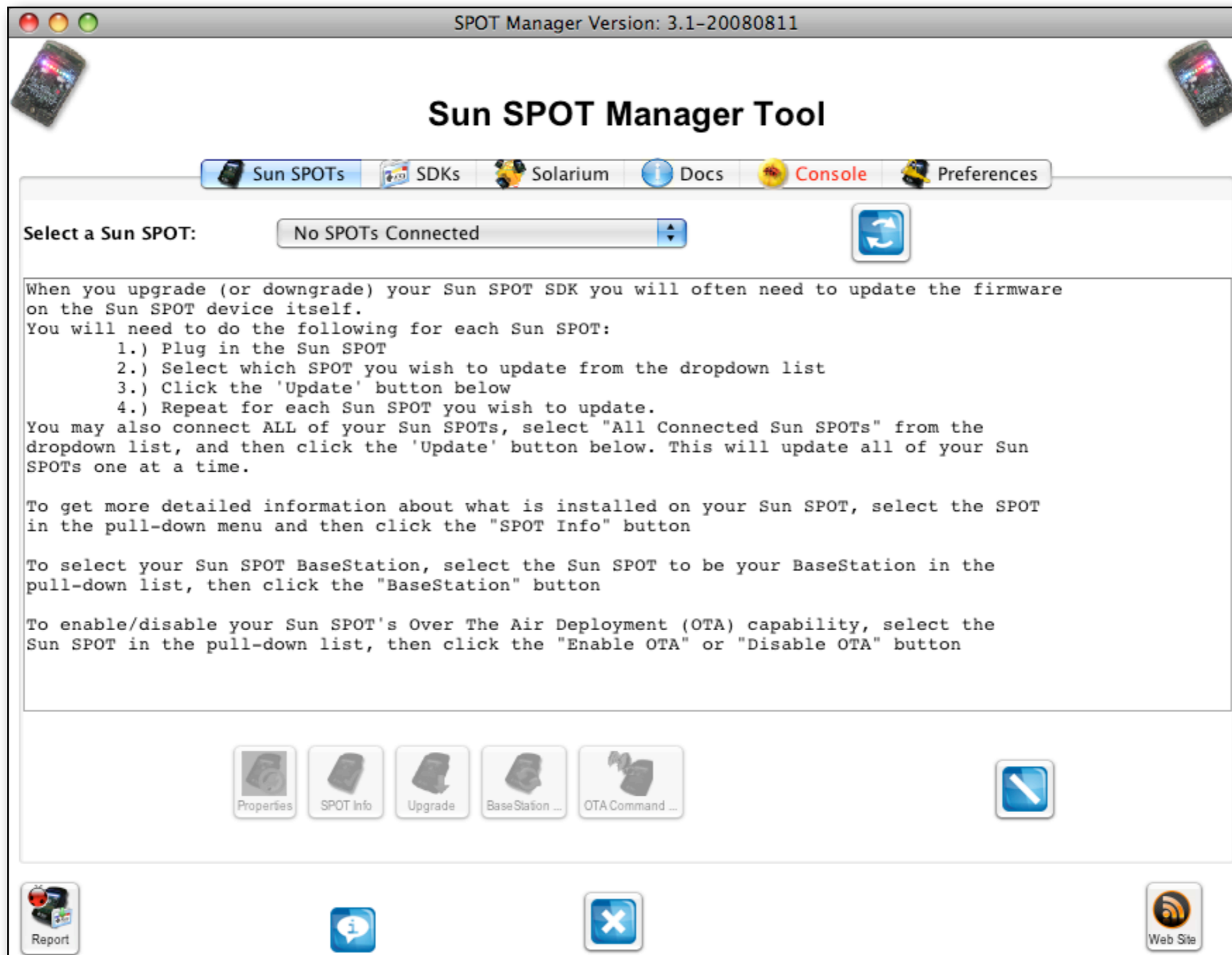


Squawk Java VM

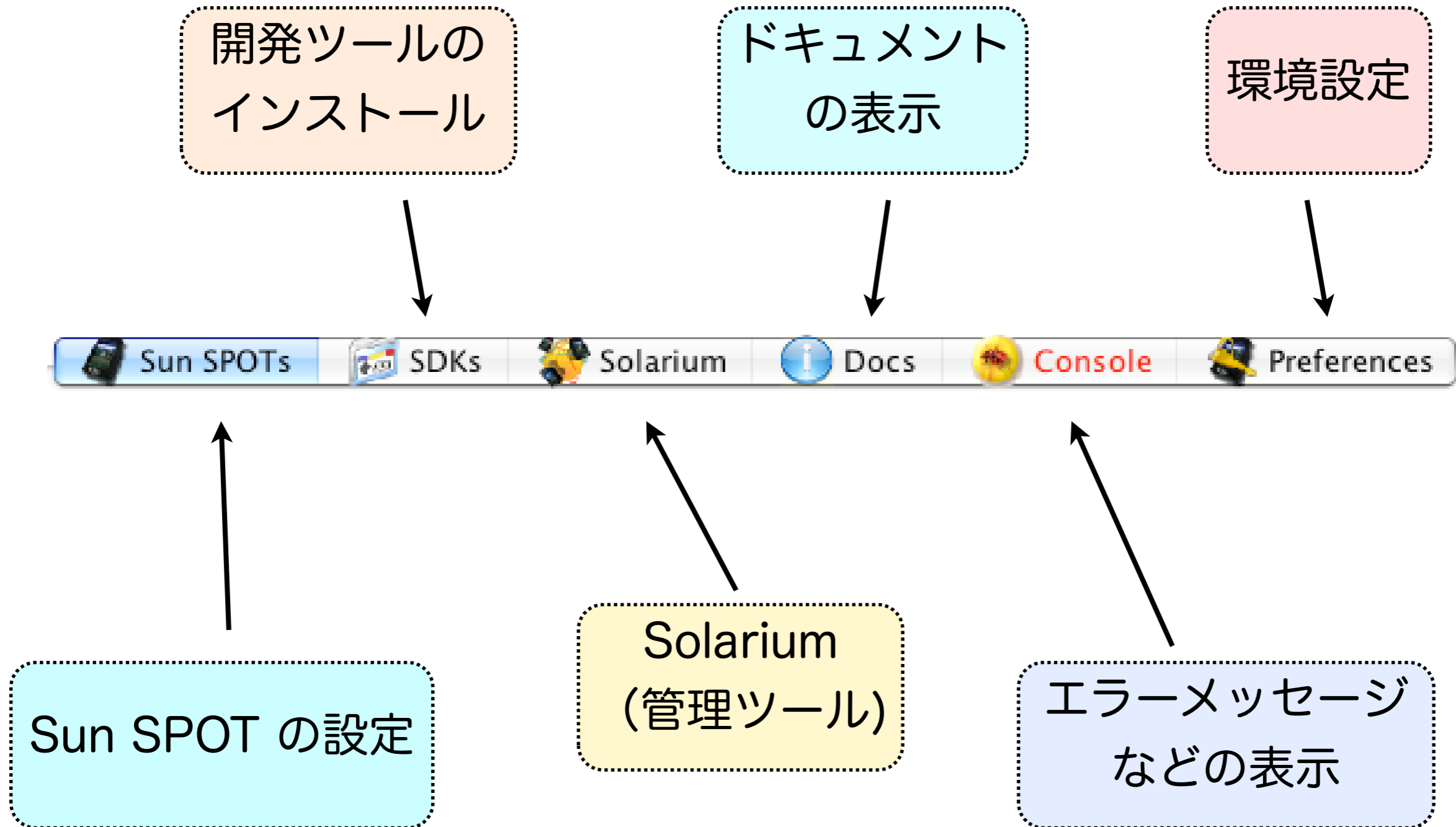
- Java ME CLDC 1.1 準拠
- OS 不要 (ARM 上で直接動作)
- ほとんどが Java 言語による実装
- リアルタイム性
 - > 0.15ms (平均)
 - > 13ms (GC 時)
- オープンソース (squawk.dev.java.net)

管理ツール

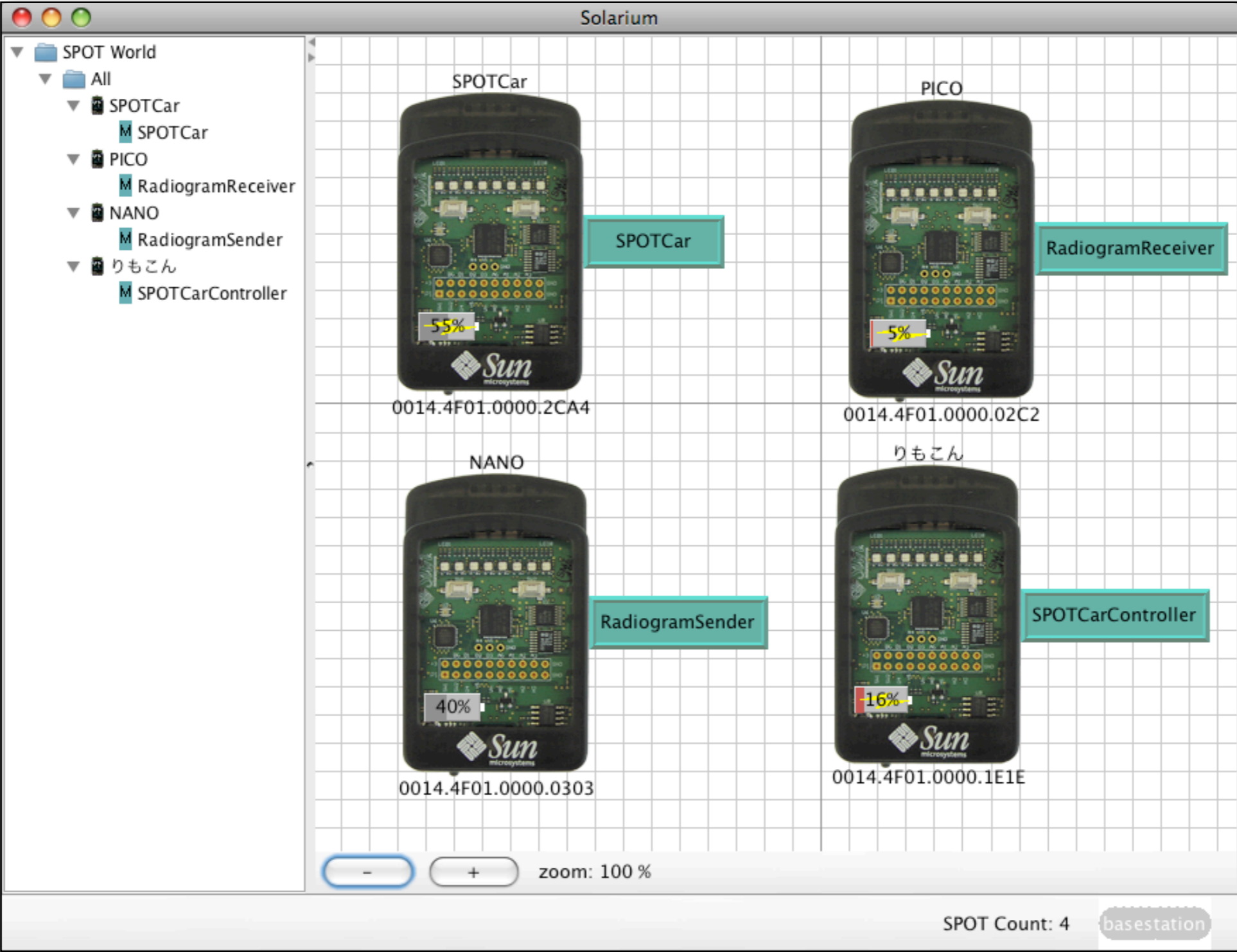
Sun SPOT Manager Tool



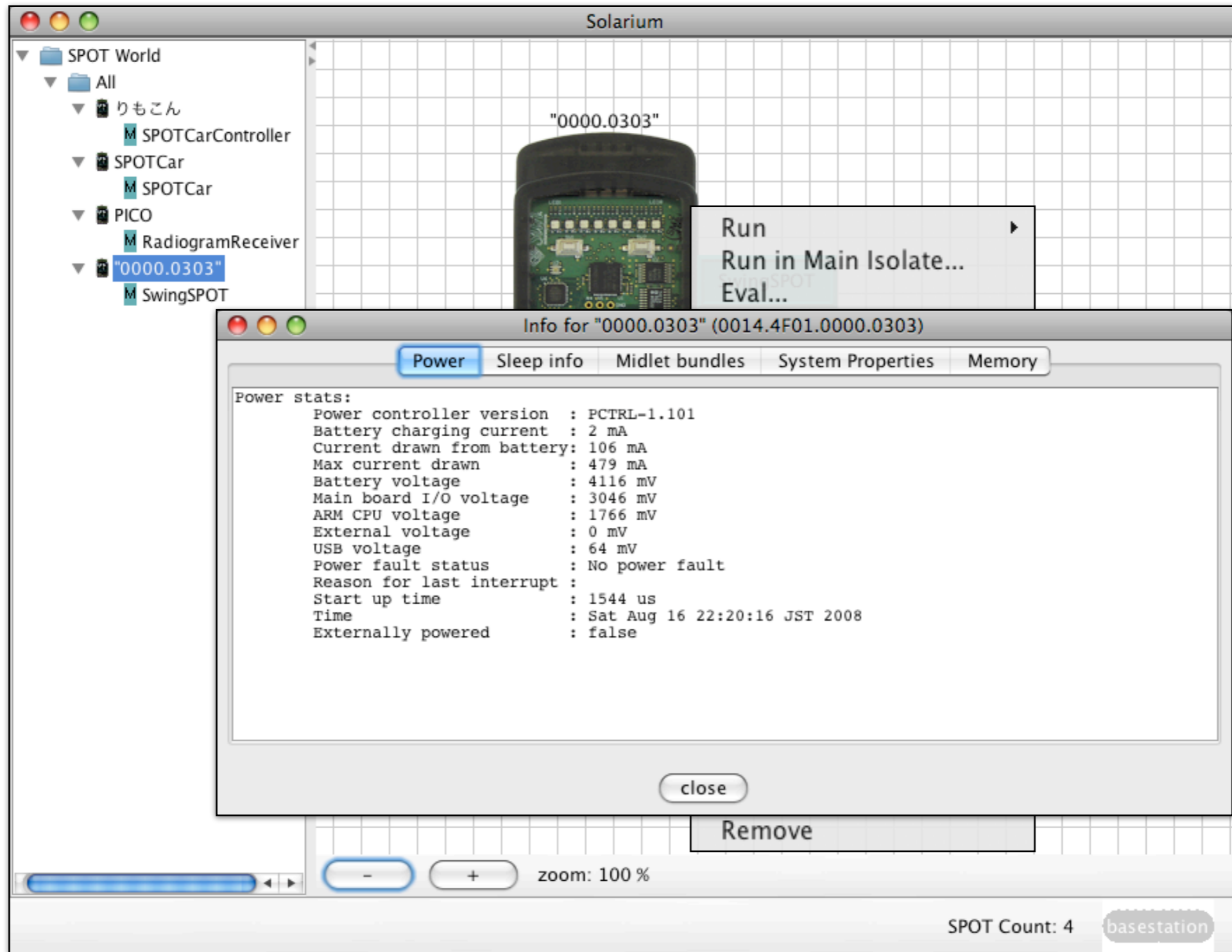
SPOT Manager Tool の機能



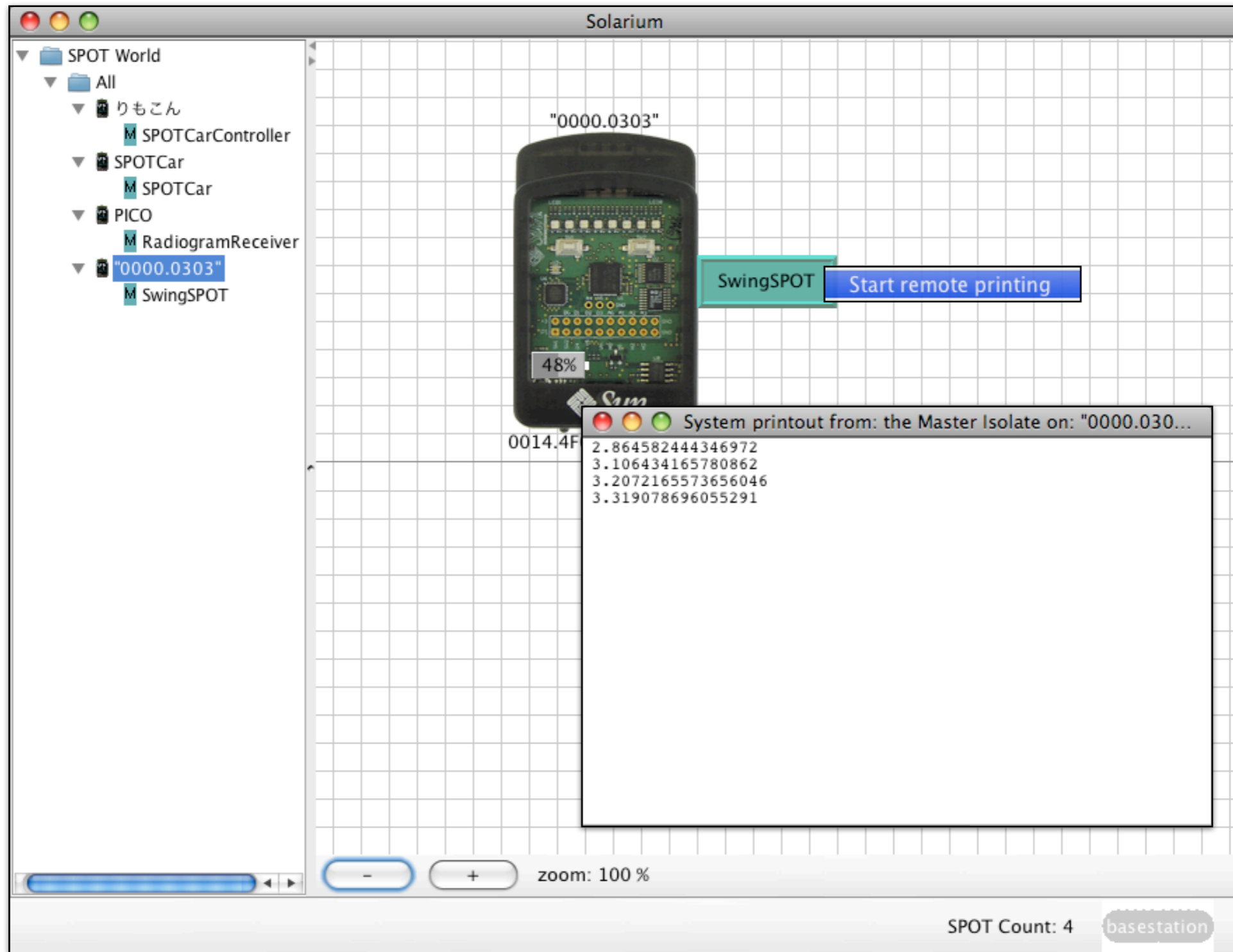
Solarium



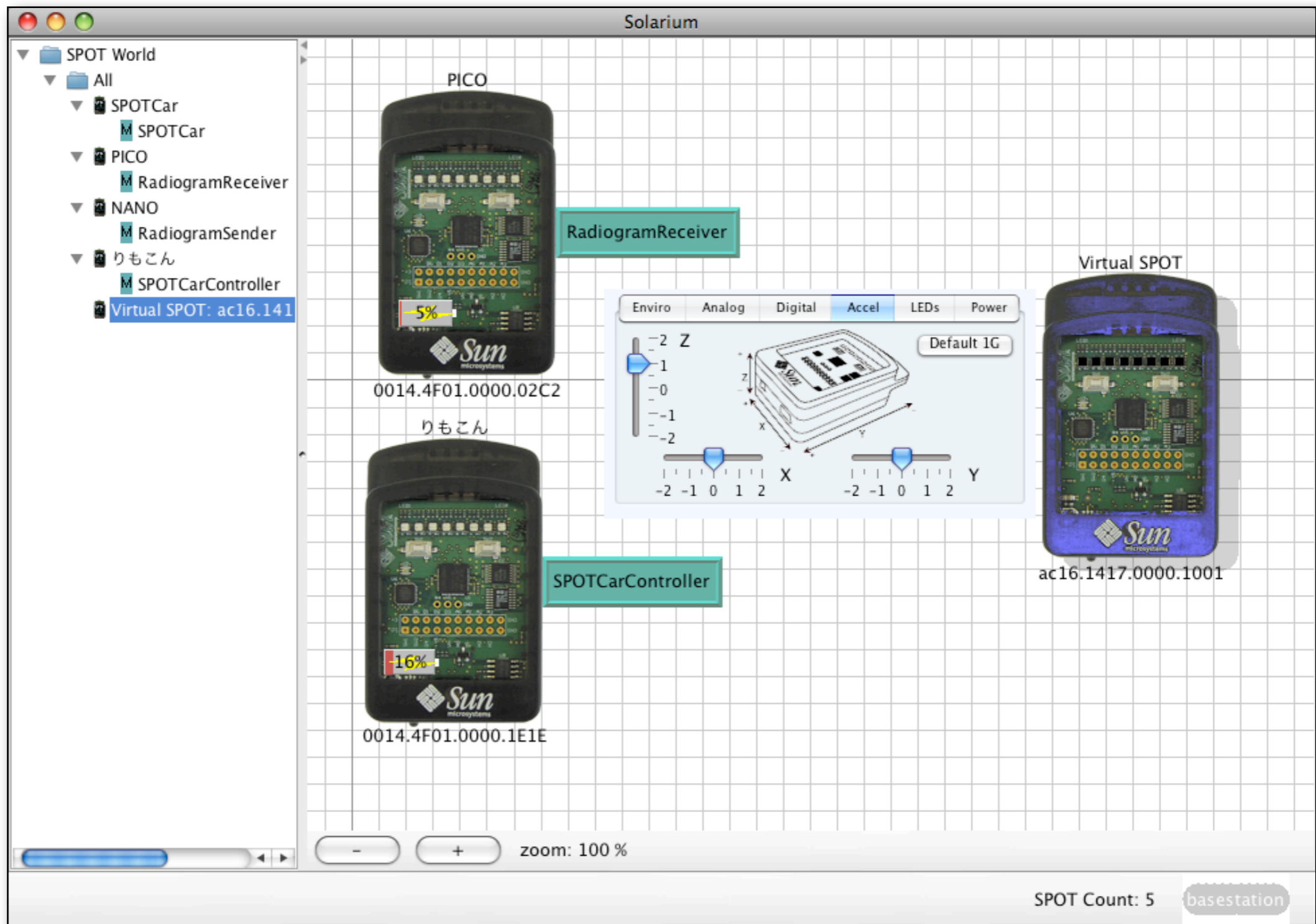
Solarium: Get Info



Solarium: Remote Printing



Virtual SPOT (エミュレータ)



プログラム開発

テンプレートの利用



The screenshot shows the NetBeans IDE 6.1 interface. In the foreground, the 'New Project' wizard is open, showing the 'Steps' pane with instructions: '1. Choose' and '2. Name a'. The 'Files' pane on the left shows a project tree with 'SunSpotApplication' selected, containing a 'src' directory with 'org.sunspotworld' and 'StartApplication.java'. The main editor window displays the source code for 'StartApplication.java'.

```

1  /*
2   * StartApplication.java
3   *
4   * Created on 2008/08/08 15:13:54;
5   */
6
7  package org.sunspotworld;
8
9  import com.sun.spot.peripheral.Spot;
10 import com.sun.spot.sensorboard.EDemoBoard;
11 import com.sun.spot.sensorboard.peripheral.ISwitch;
12 import com.sun.spot.sensorboard.peripheral.ITriColorLED;
13 import com.sun.spot.peripheral.radio.IRadioPolicyManager;
14 import com.sun.spot.io.j2me.radiostream.*;
15 import com.sun.spot.io.j2me.radiogram.*;
16 import com.sun.spot.util.*;
17
18 import java.io.*;
19 import javax.microedition.io.*;
20 import javax.microedition.midlet.MIDlet;
21 import javax.microedition.midlet.MIDletStateChangeException;
22
23 /**
24  * The startApp method of this class is called by the VM to start the
25  * application.
26  *
27  * The manifest specifies this class as MIDlet-1, which means it will
28  * be selected for execution.

```


API ドキュメントの表示



The screenshot shows the NetBeans IDE 6.1 interface. On the left is the 'Files' view showing a project structure with 'Telemetry' and 'LightSensor.java'. The main editor shows a Java file with line numbers 7-33. A Javadoc window is open, displaying the API documentation for 'com.sun.spot.sensorboard.IDemoBoard'. The window title is 'IDemoBoard (SunSPOT API V4.0 (preliminary))'. The URL is 'file:///Users/yamaguch/SunSPOT/sdk/doc/javadoc/com/sun/spot/ser...'. The documentation includes navigation links, a summary, the interface name, implementing classes, and a method summary table.

Overview Package Class Use Tree Deprecated Index Help SunSPOT API V4.0 (preliminary)

[PREV CLASS](#) [NEXT CLASS](#) [FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#) DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

com.sun.spot.sensorboard

Interface IDemoBoard

All Known Implementing Classes:

[EDemoBoard](#)

```
public interface IDemoBoard
```

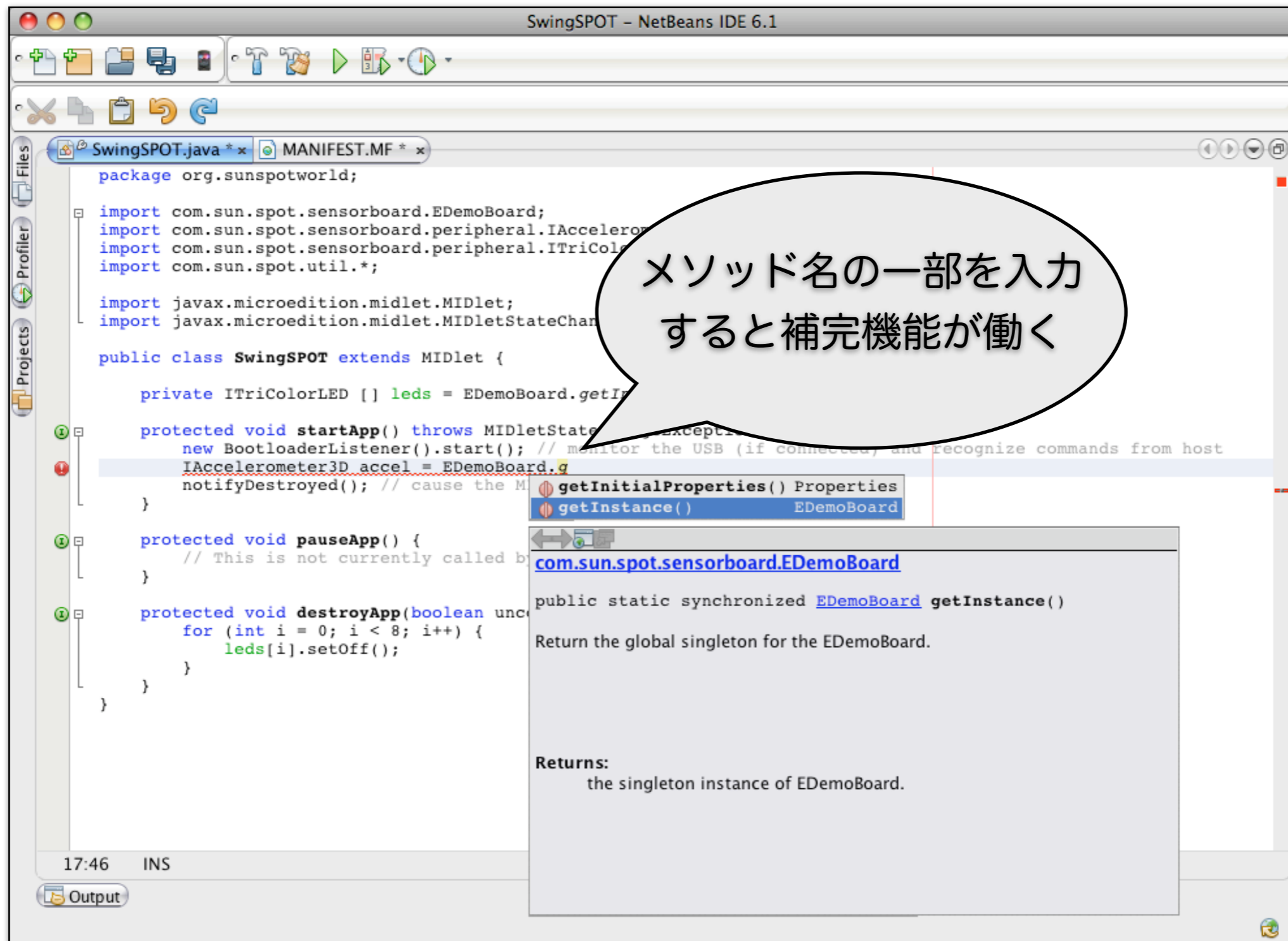
Definition of the generic interface for a SunSPOT demo board. Defines methods to access sensors & effectors that are specific to the DemoBoards.

Method Summary

IAccelerometer3D	getAccelerometer() Method to get built in accelerometer.
ITemperatureInput	getADCTemperature() Method to get internal temperature sensor on the ADC from the edemo board.
IIOPin[]	getIOPins() Method to return an array of the builtin bidirectional digital pins.

Display a menu

エディタの補完機能



メソッド名の一部を入力すると補完機能が働く

```

package org.sunspotworld;

import com.sun.spot.sensorboard.EDemoBoard;
import com.sun.spot.sensorboard.peripheral.IAccelerometer;
import com.sun.spot.sensorboard.peripheral.ITriColorLED;
import com.sun.spot.util.*;

import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;

public class SwingSPOT extends MIDlet {

    private ITriColorLED [] leds = EDemoBoard.getInstance();

    protected void startApp() throws MIDletStateChangeException {
        new BootloaderListener().start(); // monitor the USB (if connected) and recognize commands from host
        IAccelerometer3D accel = EDemoBoard.getInstance();
        notifyDestroyed(); // cause the MIDlet to be destroyed
    }

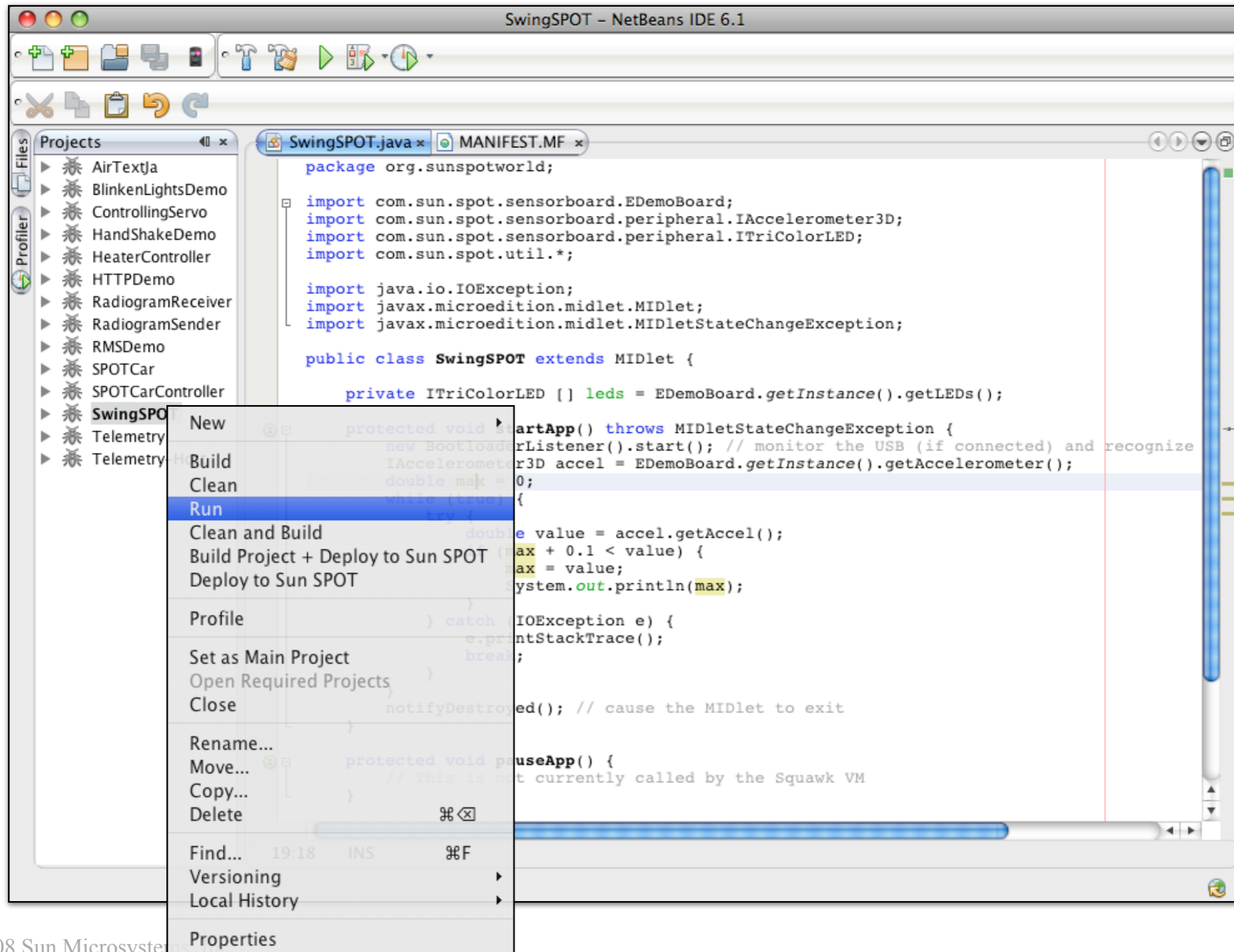
    protected void pauseApp() {
        // This is not currently called by the MIDlet
    }

    protected void destroyApp(boolean unconditional) throws MIDletStateChangeException {
        for (int i = 0; i < 8; i++) {
            leds[i].setOff();
        }
    }
}

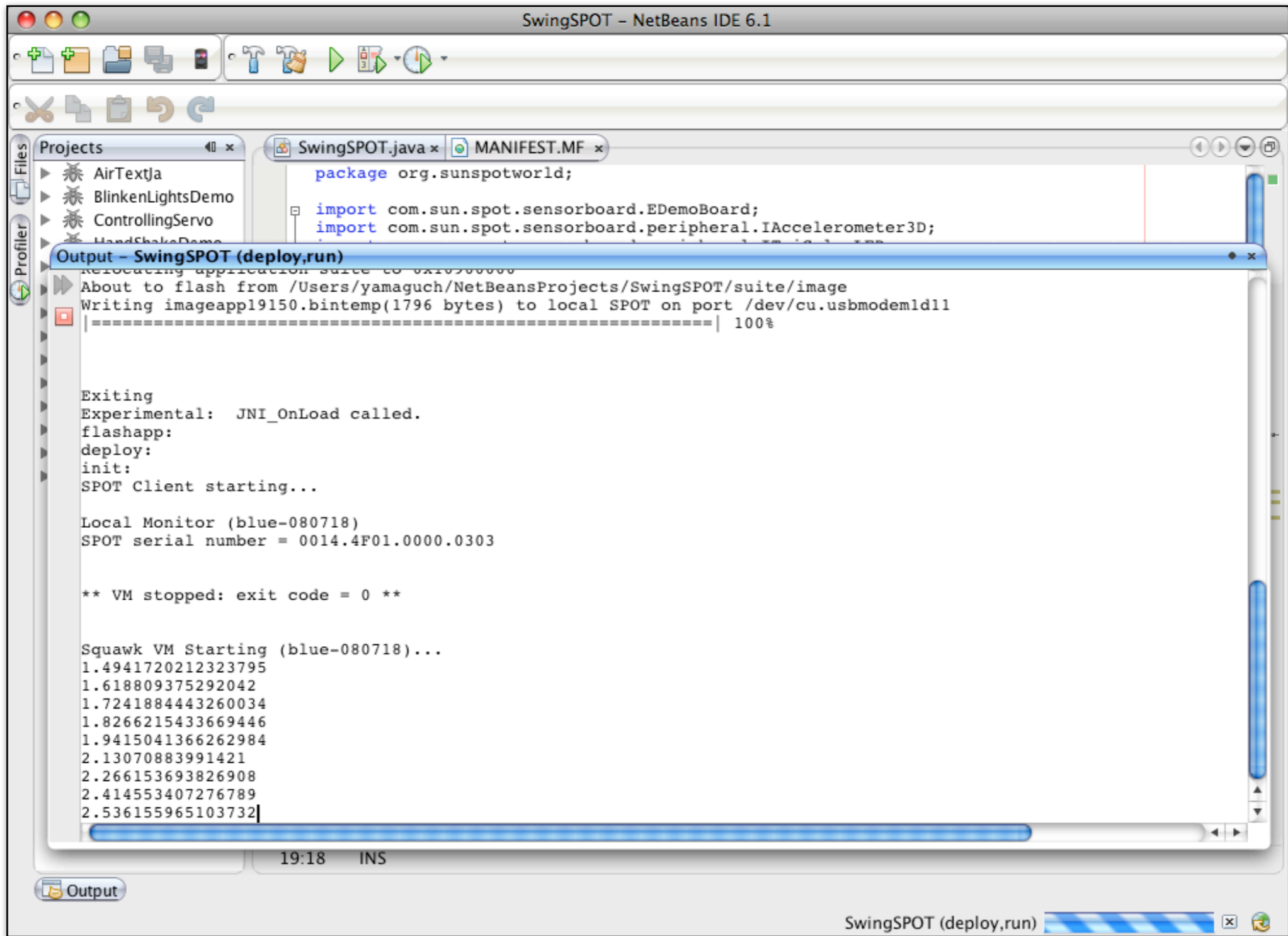
```

getInitialProperties() Properties
getInstance() EDemoBoard
[com.sun.spot.sensorboard.EDemoBoard](#)
 public static synchronized EDemoBoard getInstance()
 Return the global singleton for the EDemoBoard.
Returns:
 the singleton instance of EDemoBoard.

コンパイル+デプロイ+実行



実行結果



応用

コンテナ型データセンターに



A 3D Accelerometer. Rigid body simulation: records movement of the Blackbox container in transit.



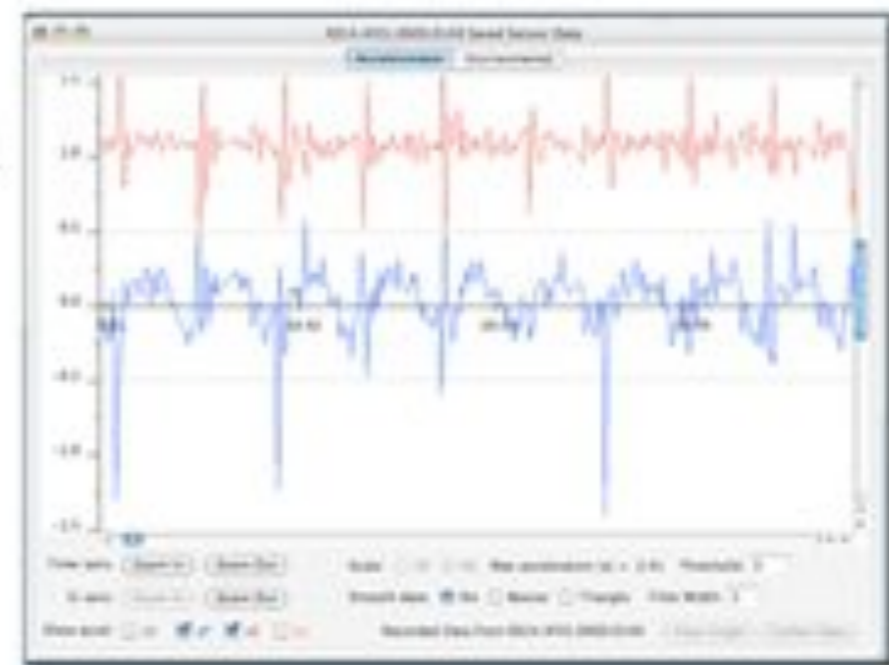
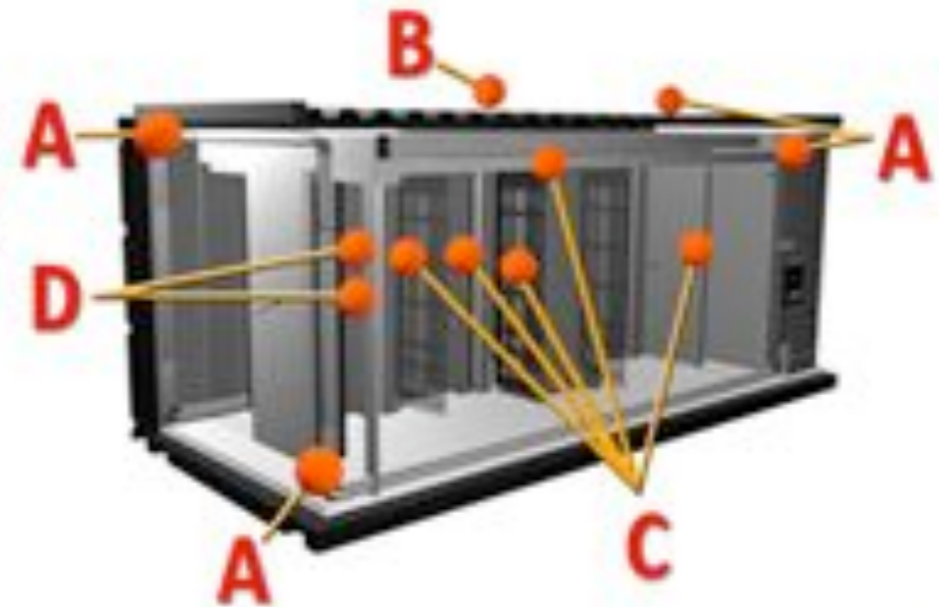
B GPS. Tracks Blackbox location.



C Temperature, humidity and light. Monitors environmental conditions of systems in transit.

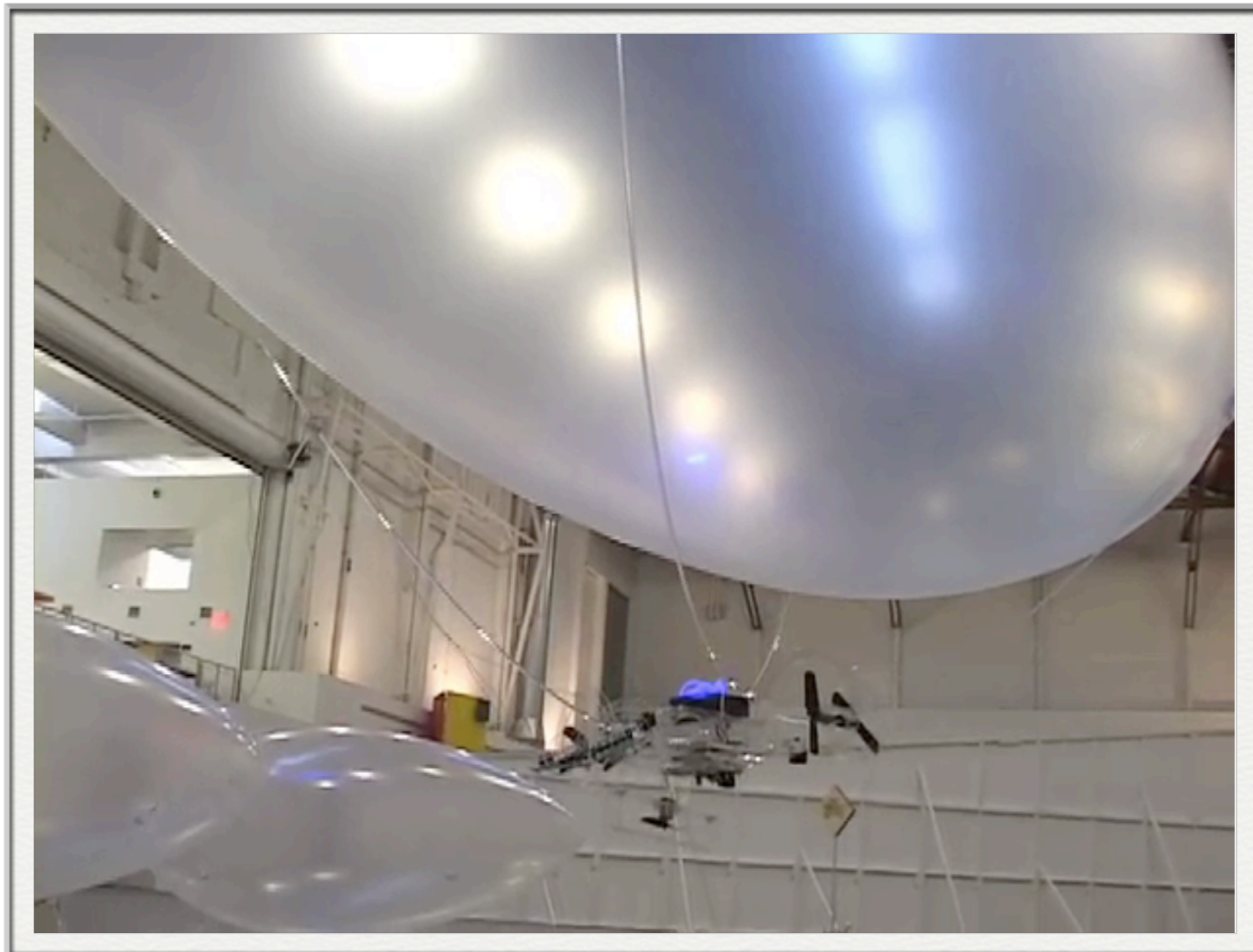


D Data loggers. Redundant data collection.



<http://www.youtube.com/watch?v=aeLPCn31oIM>

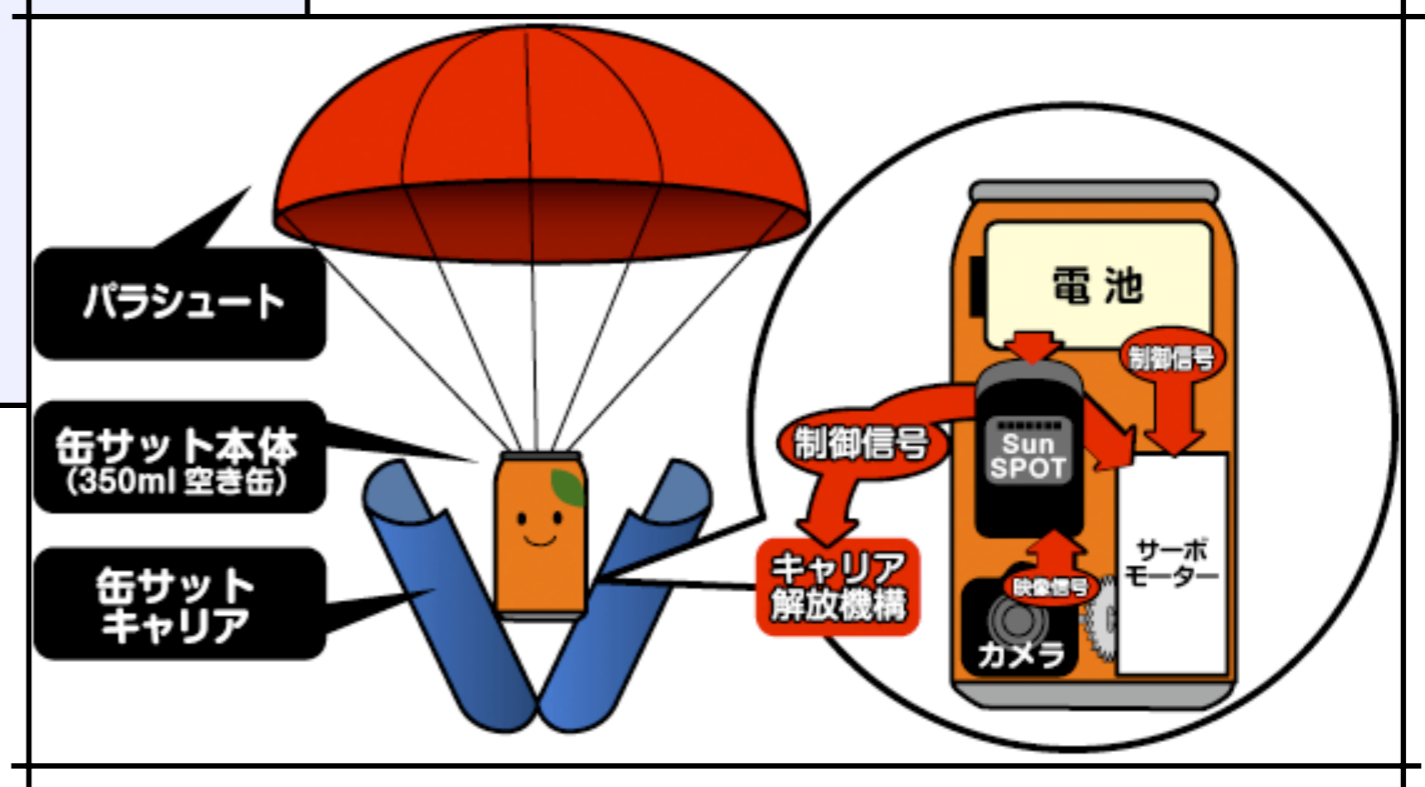
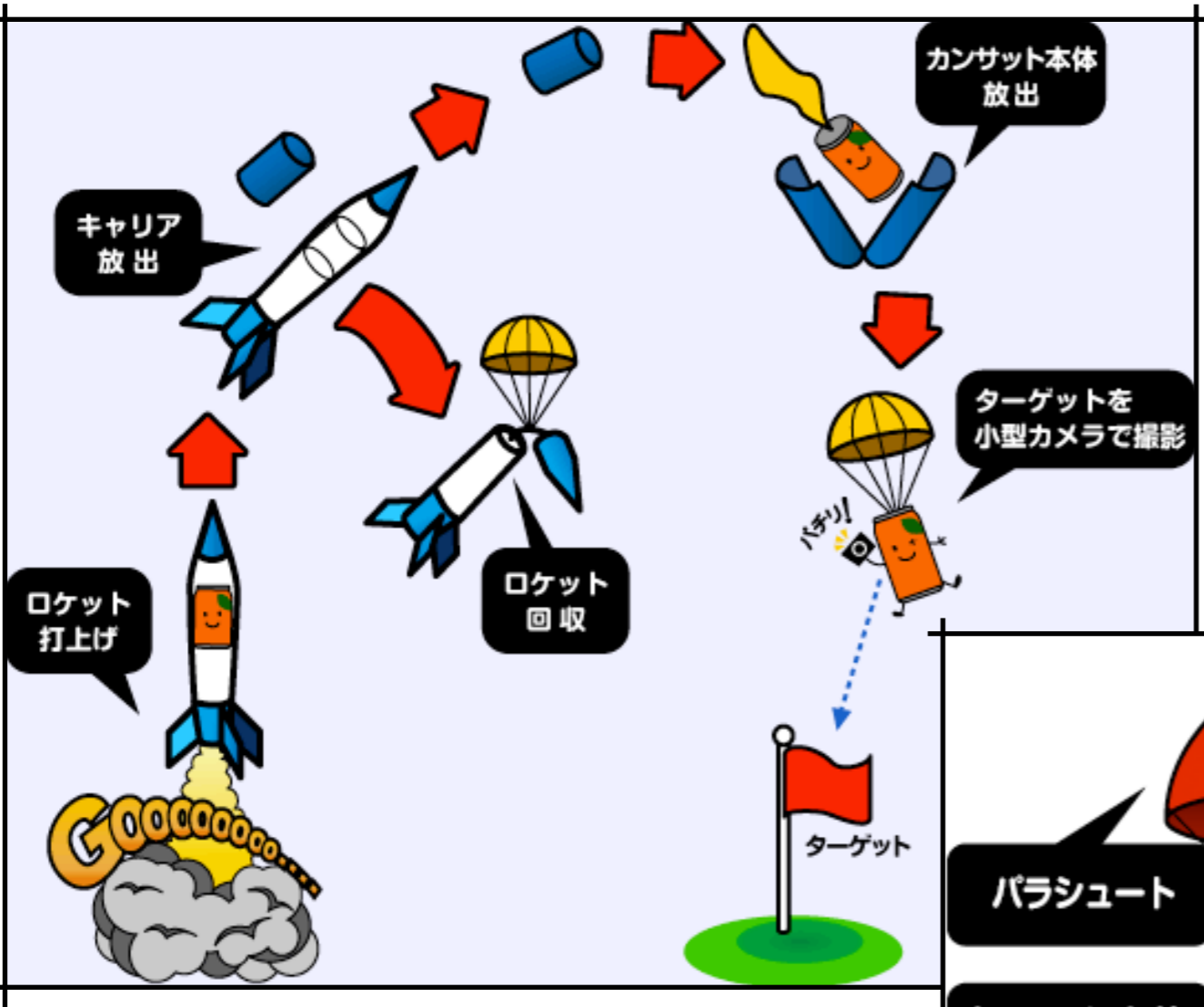
自律型飛行船 (ALAVs)



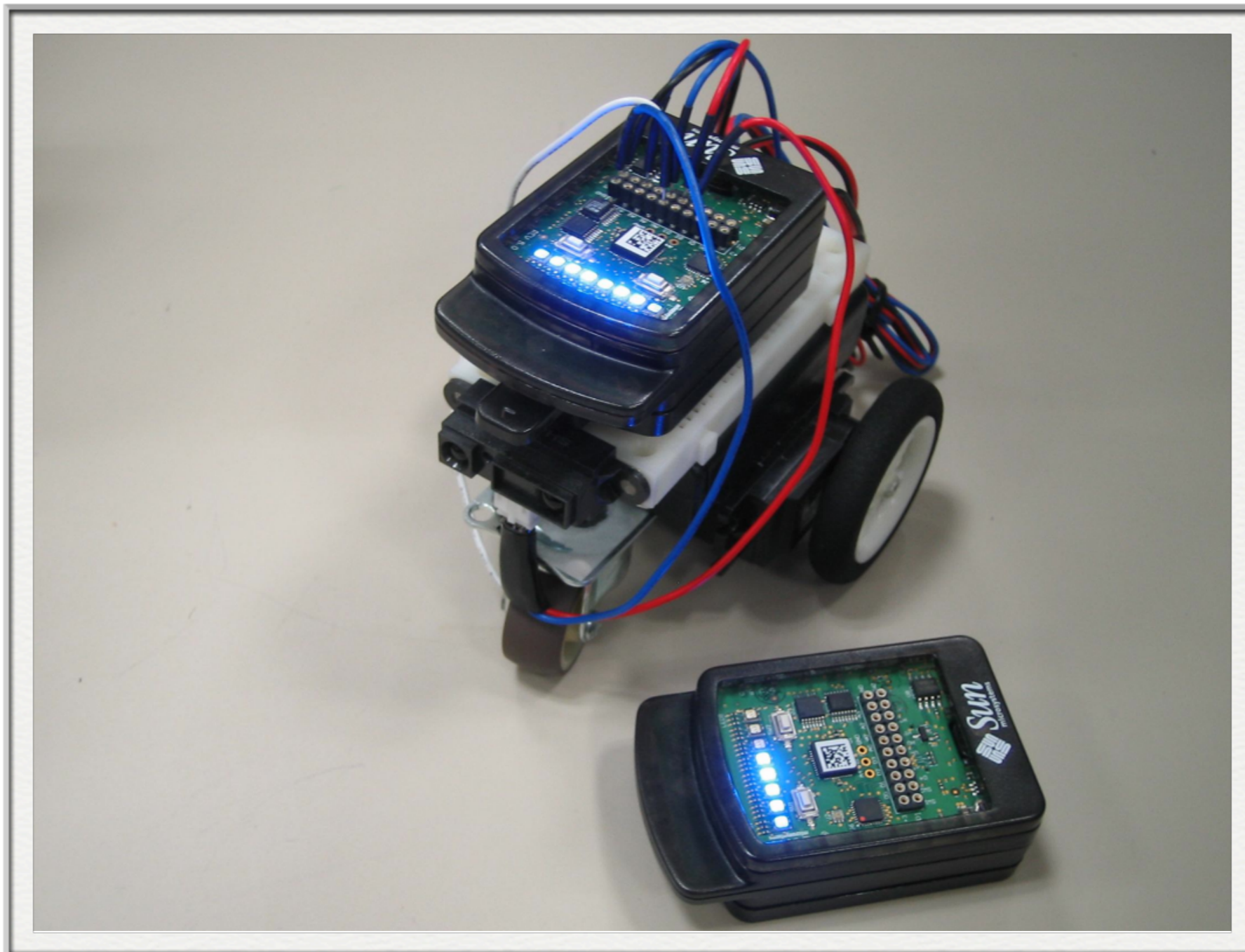
Sun SPOT 宇宙へ ... !?



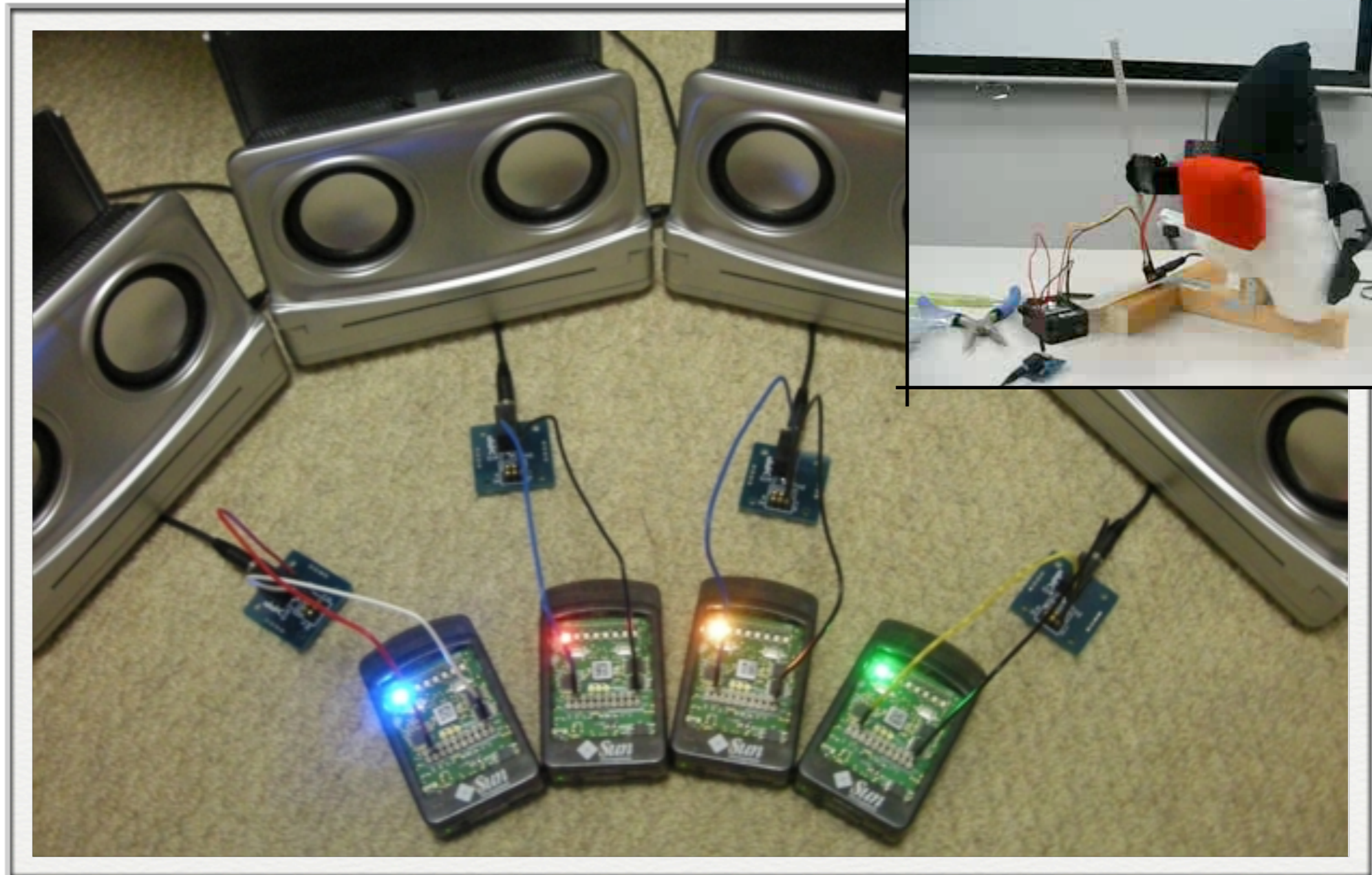
缶サット甲子園



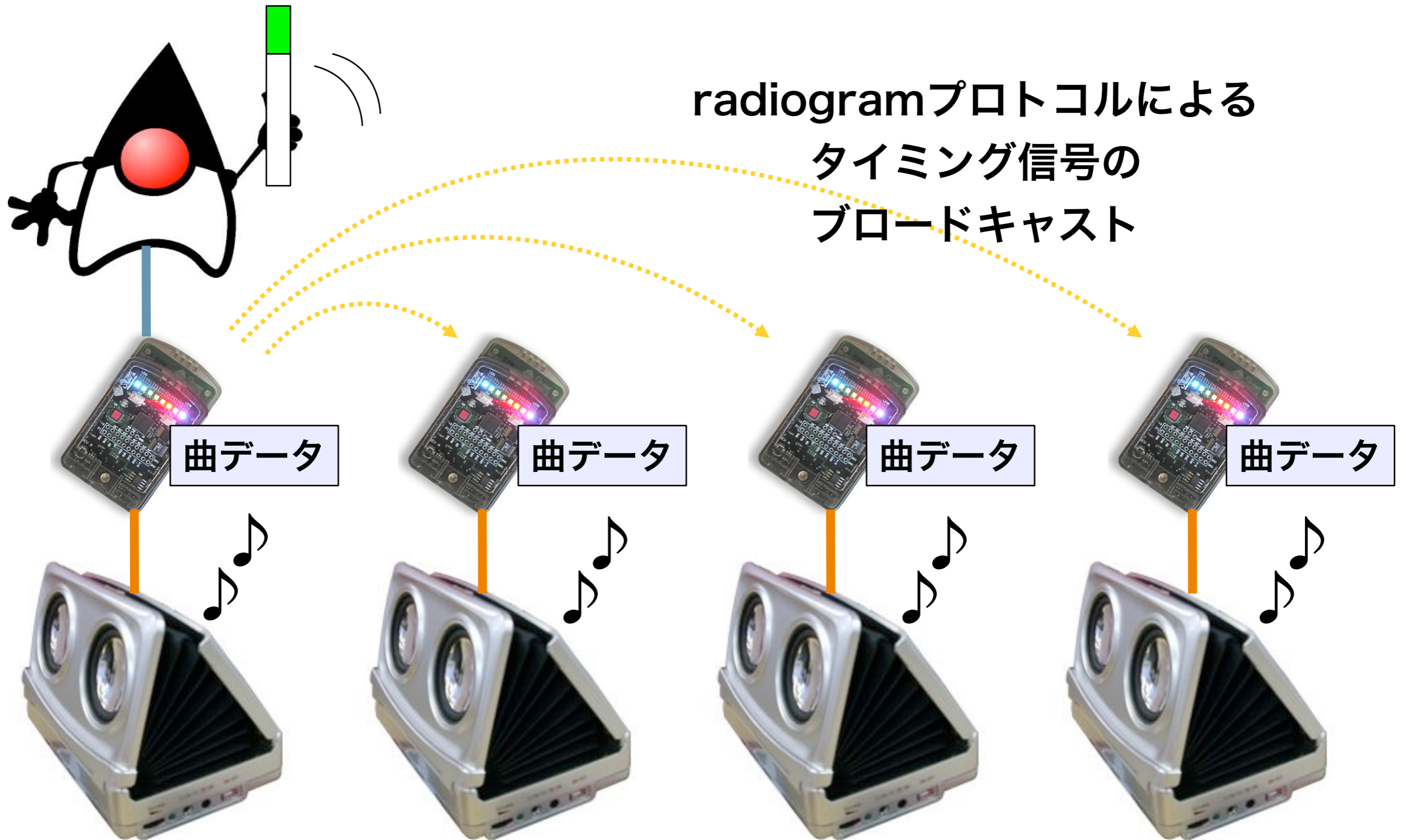
Sun SPOT リモコンカー



Sun SPOT カルテット



Sun SPOTカルテット



Javaソースコードの断片

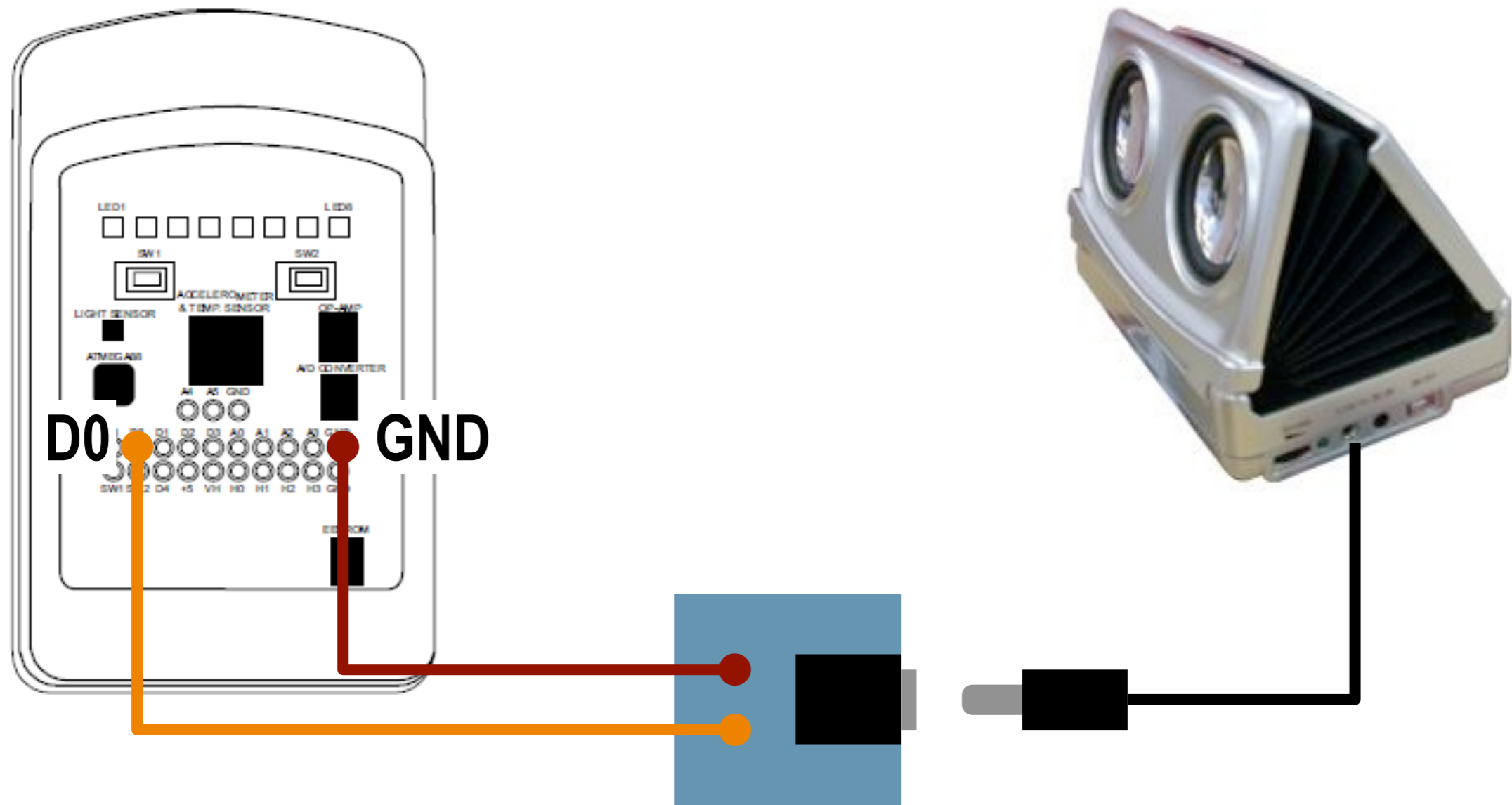
トーンジェネレータ

```
int tone_map[] = {9091, 8581, 8099, ...};
ToneGenerator toneGen =
    new ToneGenerator(board.getIOPins()[EDemoBoard.D0]);
toneGen.setPeriod(tone_map[note]);
toneGen.setDuration(duration);
toneGen.beep();
```

サーボ

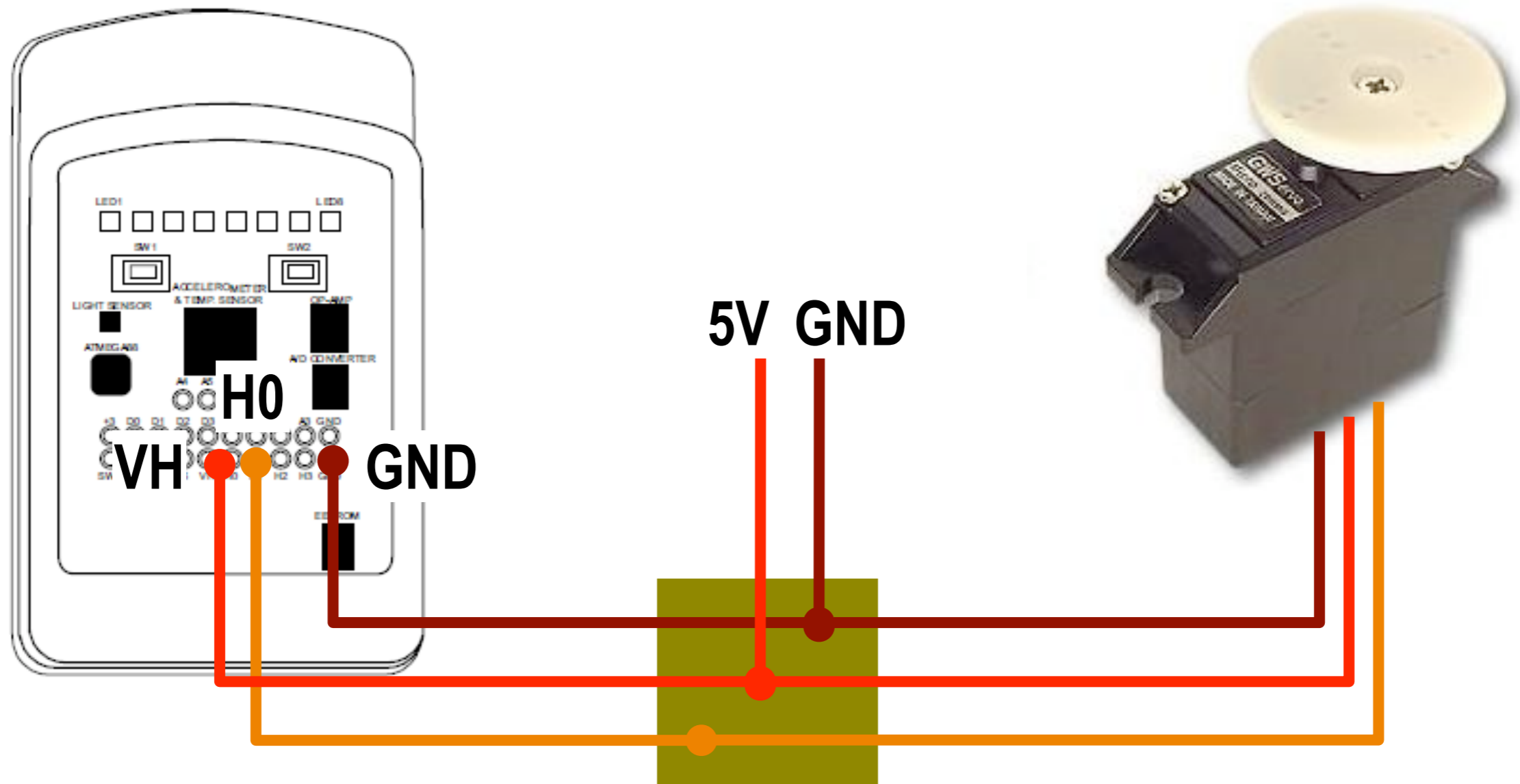
```
IServo servo =
    new Servo(board.getOutputPins()[EDemoBoard.H0]);
servo.setBounds(1200, 1500);
servo.setPosition(0.0f);
Utils.sleep(500);
servo.setPosition(1.0f);
```

スピーカーの接続



イヤフォンジャック変換基板

サーボモーターの接続



www.sunspotworld.com

SunSpotWorld - Home of Project Sun SPOT

http://www.sunspotworld.com/ Google

EXPERIMENTAL TECHNOLOGY FROM SUN MICROSYSTEMS LABORATORIES

PROJECT SUN SPOT

SUN SMALL PROGRAMMABLE OBJECT TECHNOLOGY

ABOUT SUN SPOTS

INFORMATION & RESOURCES

BUY A SUN SPOT

LATEST FORUM POSTINGS

- How to choose seed value for Random
Wed Sep 17, 2008 3:43 pm
- what are USART and UART
Thu Sep 18, 2008 5:18 am
- SpotManager does not start

SUN SPOT NEWS & EVENTS

- Sun SPOT SDK v4.0 (Blue) is now available (Sep 17)
- Discount Pricing for schools and US students (Apr 15)
- Now available in Canada (Apr 15)
- Sun SPOTs Go Open Source - an interview with Arshan Poursohi (Feb 12)

SUN SPOT IN EDUCATION

SUN SPOT FORUMS

java.net PARTNER

Contact Us | Copyright 2008 Sun Microsystems, Inc. All rights reserved. | Privacy | Terms of Use | Trademarks

Interface 10月号

インターフェース 3000円～5000円

GCC&μITRON&TCP/IPプロトコルスタックを使いこなす

**ColdFire基板で学ぶ
ネットワーク機器開発技法**

- GNUクロス開発環境の構築
- μITRON 4.0仕様準拠OS TOPPERS/JSPの移植
- FATファイルシステムFatFs for TOPPERSの移植
- TCP/IPプロトコルスタックTINETの実装
- P2P地震情報とColdFire基板の応用

● 注目点 シリアルATAコアとシミュレーション

● 注目点 Windows Vista時代のデバイスドライバ開発

● 注目点 FR60基板を用いたデジタル・フォトフレーム

● 注目点 **Javaを使ったSun SPOT
センサ・ネットワーク**

Oct. 2008
10

<http://www.cqpub.co.jp/interface/>
<http://www.kumikoni.net/>

Interface

Javaを使ったオープン・ソース・プロジェクト

Sun SPOTでセンサ・ネットワーク事始め

草薙 昭彦/町田 修一/山口 浩

3軸加速度/温度/湿度センサ, 8個の3色LEDをもつ無線センサ・ネットワーク・デバイス「Sun SPOT」と、それを使った簡単な応用例を解説する。Sun SPOTのハードウェア仕様としては、ARM-920T(180MHz)、512KバイトRAM, 4Mバイト・フラッシュROMなどから構成される。またソフトウェア仕様としてはJavaテクノロジーを採用し、JavaVMとして無線センサ・ネットワーク開発環境の現状、リソース制約の厳しいデバイス向けのSquawk VMを搭載している。ハードウェア、ソフトウェア共にオープン・ソースとして公開され、開発時のハードルを低くしている。(編集部)

1. 無線センサ・ネットワークと Sun SPOT

● 無線センサ・ネットワークと、それを実現するデバイス

私たちの身の回りにはセンサがあふれています。温度センサや光センサ、タッチ・センサ、人感センサなど、小さなセンサ部品がさまざまな製品に埋め込まれ、毎日の生活を便利にしています。

さて、これらのセンサがそれぞれに小さな頭脳を持ち、互いに通信し合うようになったらどうなるかを想像してみてください。小さな駆動装置が組み込まれているものもあります。数も数個、数十個ではなく、何千、何万と増えていったら、どうでしょう？ 私たちのような生命体も、同じようなモデルだと見せるかもしれません。あるいはジュラシック・パークの原作者として有名なマイケル・クライトンの「プレイ (Prey)」というSF小説を思い出される方もいるでしょう。

無線センサ・ネットワークは、このような領域を対象として10年ほど前から研究が始まりました。その後、センサも多種多様なものが登場するようになり、無線チップやプロセッサなどの小型高性能化と低価格化が進んだ結果、現在ではユビキタス社会を実現するための重要な技術として注目されるようになってきました。

本稿では、このような無線センサ・ネットワークを構成するデバイスを取り上げます。基本的な構造は図1のようなもので、入力を行うセンサと出力を行うアクチュエータ、無線通信モジュール、そしてこれらを制御するコントローラから構成されています。コントローラ部分はマイクロプロセッサやメモリを備える小さなコンピュータで、この部

分をプログラムすることで「小さな頭脳」を実現します。

● 無線センサ・ネットワーク開発環境の現状

コントローラ部分のプログラミングには、周辺のデバイスに使われている電子回路に関する知識と合わせて、使用しているマイクロプロセッサのアセンブリ言語や組み込み機器向けのC言語の知識が要求されます。そのため、無線センサ・デバイスを提供するベンダの多くが同時に開発ツールを提供していますが、これらはベンダごとに異なることが多いので、使いこなせるようになるまでにはある程度の訓練が必要です。さらに無線ネットワークに関する知識も必要で、特にパソコンと連携させる場合には、パソコン上のアプリケーション開発に関する知識も必要となります。このようなことから、無線センサ・ネットワーク・デバイスに関連するアプリケーションの開発は、一般的な組み込みアプリケーションの開発と比べ難易度が高いものと考えられていました。

一方、パソコン上のアプリケーションやサーバ系アプリケーションに目を向けてみると、さまざまなオープン・ソース化の影響も手伝って、無償で汎用性が高く使い勝手

図1 無線センサ・ネットワーク・デバイス

Interface Oct. 2008 **KEYWORD** 無線センサ・ネットワーク・デバイス, Java, ARM-920T, 3軸加速度/温度/湿度センサ, Squawk VM, IEEE 802.15.4, CLDC, NetBeans 163



Enjoy Java with Sun SPOT!

Akihiko.Kusanagi@Sun.COM
Hiroshi.Yamaguchi@Sun.COM

