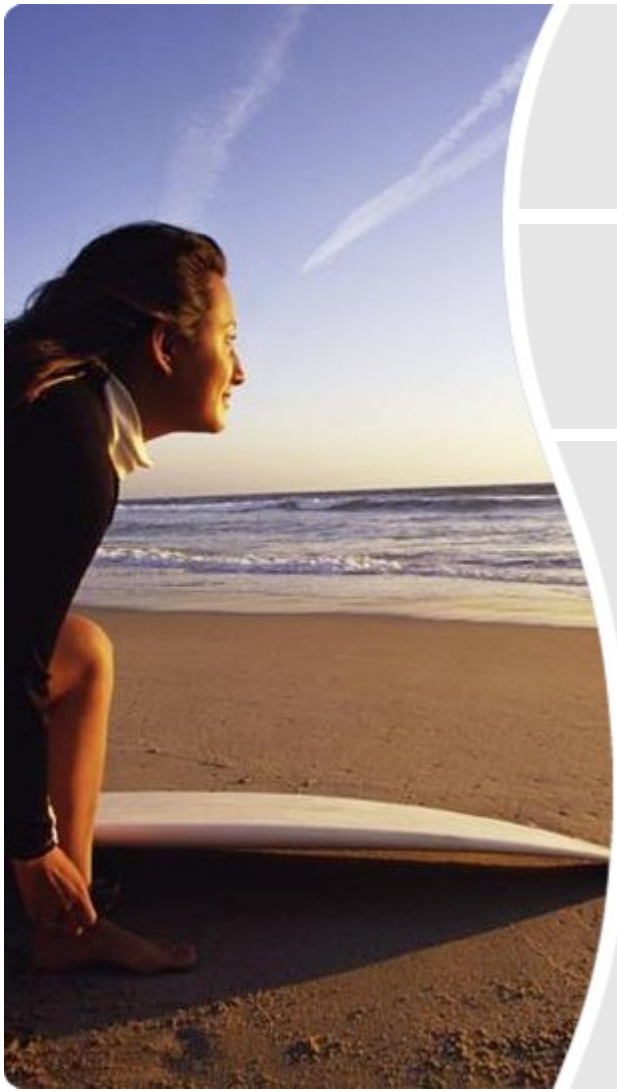
Have Fun with Sun SPOT

cookys（林秉毅）

CA of Feng Chia University

cookys@gmail.com　Coscup 2009

Agenda



Have Fun With Sun SPOT

What is Sun SPOT?

The Squawk Java VM

Java on Sun SPOT

What To Do/Where To Go

Who Am I?

- 林秉毅　Aka cookys
- 苦命菸酒生　in 逢甲
- Sun CA (Campus Ambassdor)

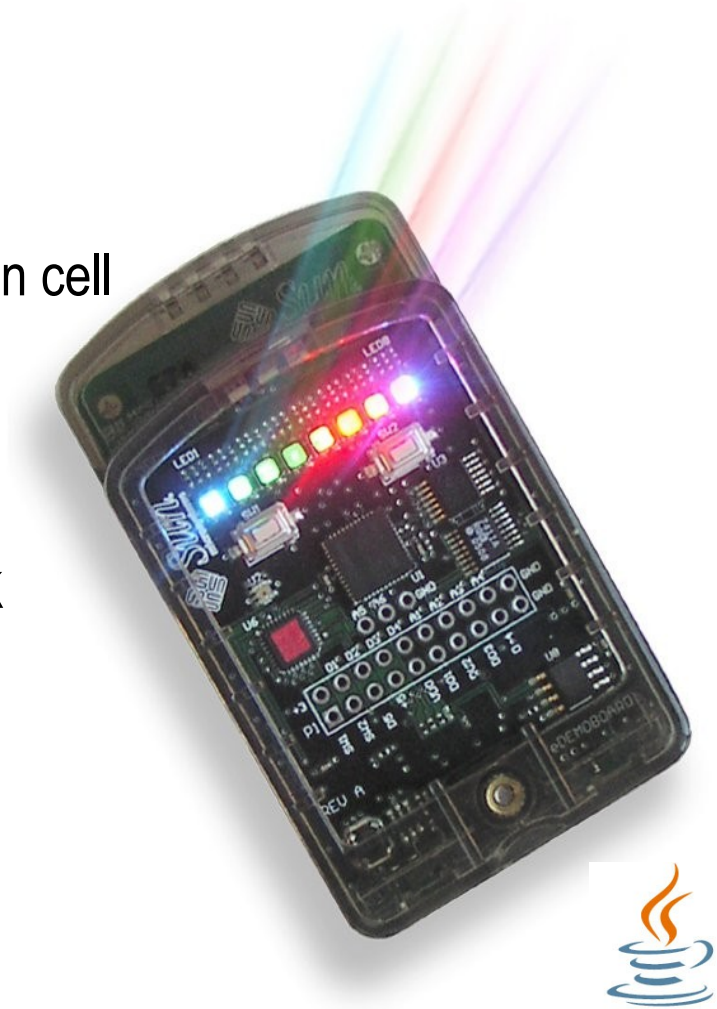
- Blog: http://blog.cookys.org
- Email: cookys@gmail.com

What is Sun SPOT?

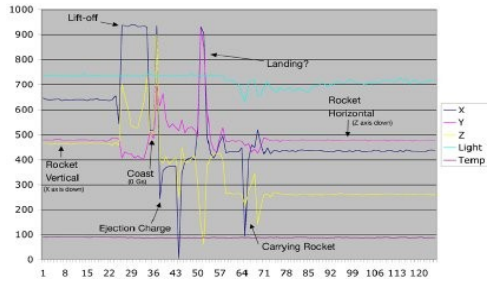
太陽．點

What is Sun SPOT?

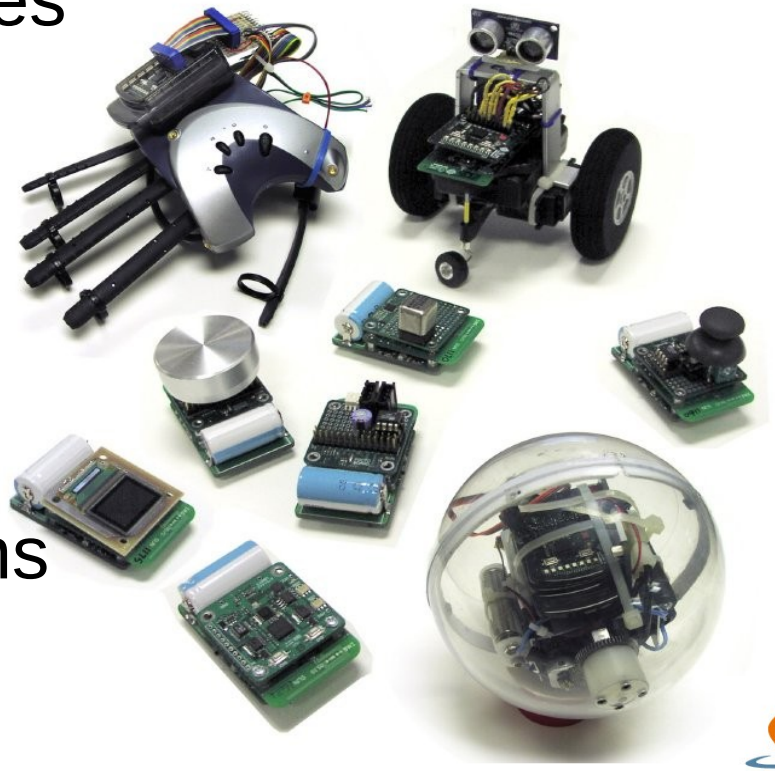
- **S**mall **P**rogrammable **O**bject **T**echnology
  - > Java on small devices
    - > Sun has licensed Java on over 1.5 billion cell phones
  - > Programming the world with Java
  - > Research project from Sun Labs
  - > Platform for wireless sensor network applications
    - > Helping to build the “network of things”



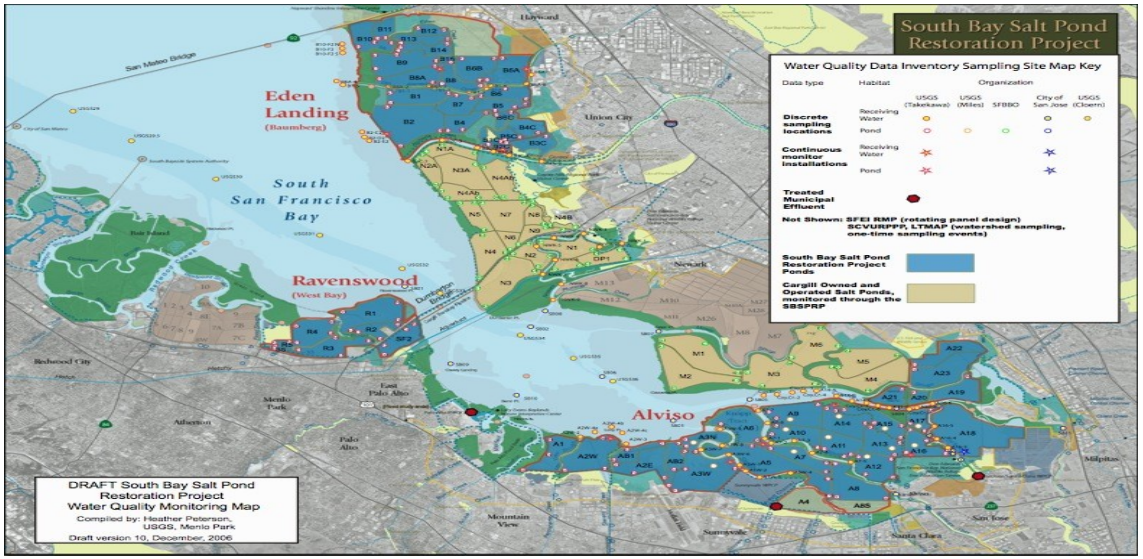
What is Sun SPOT?



- A Java platform for developing applications for wireless networks and small devices
- Applications such as:
  - Robotics
  - Art
  - Toys
  - Personal electronics
  - Commercial Applications
  - Telemetries



Ravenswood 溼地監控



USGS, Menlo Park
南三藩溼地復原項目
目標：監控溼地環境
部署：美國三藩市

世界上最大規模的溼地復原計畫，覆蓋超過 15000 英畝的溼地。需要持續地監測該區域的水質、濕度、溫度等等重要參數。透過傳感器節點進行遠端控制、維護、升級，減少現場勘查的次數。系統採用太陽能發電解決野外供電問題。

http://www.southbayrestoration.org/

輕量級無人駕駛飛行器

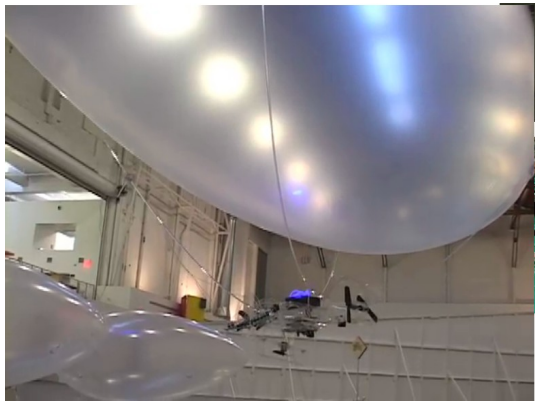
Art Center College of Design

新型武器設計概念

目標：新型飛行器設計

部署：原型展覽

透過 Sun SPOT 監控飛行器的位置、速度以及飛行方向，驅動推進器改變自身的軌跡以改變自身的軌跡以避免與其他飛行器碰撞。用戶可以向 Sun SPOT 發送訊息甚至是打電話來控制飛行器的行為。（起飛、降落起狀態等等），在基站可視覺化地顯示飛行器的位置。 http://www.alavs.com/



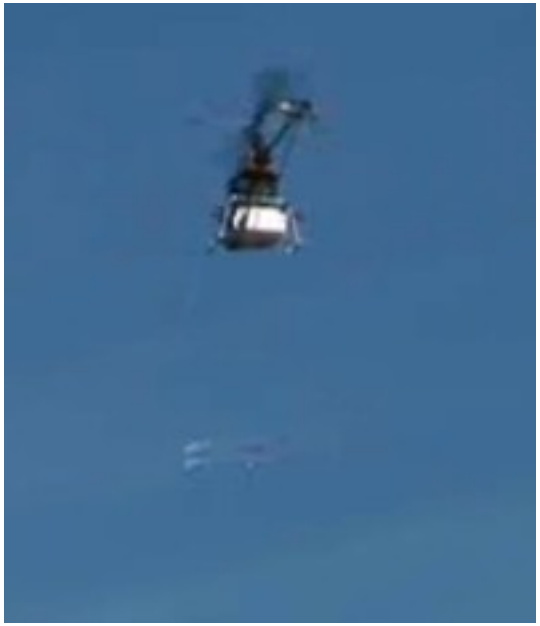
無人駕駛飛行器

Renaissance Computing Institute

災難現場監控

目標：災難反應與管理

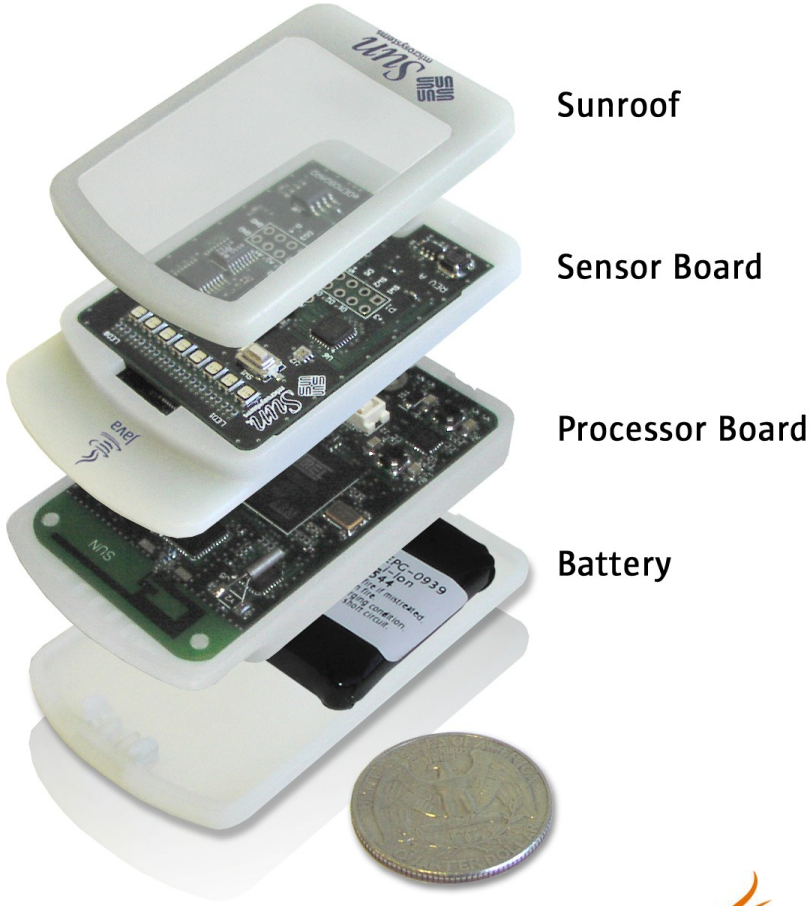
部署：美國三藩市



在災害地區收集現場數據，以幫助災害反應與管理方面的決策。在無人駕駛飛行器上配有 GPS、攝影機以及多種感應器，透過 Sun SPOT 相互通訊並且與指揮部進行聯繫。飛行器可以空投裝備有傳感器的 Sun Spot 到地面，從而透過無線網路收集數據。 http://www.renci.org/news/uav.php

Sun SPOT Device

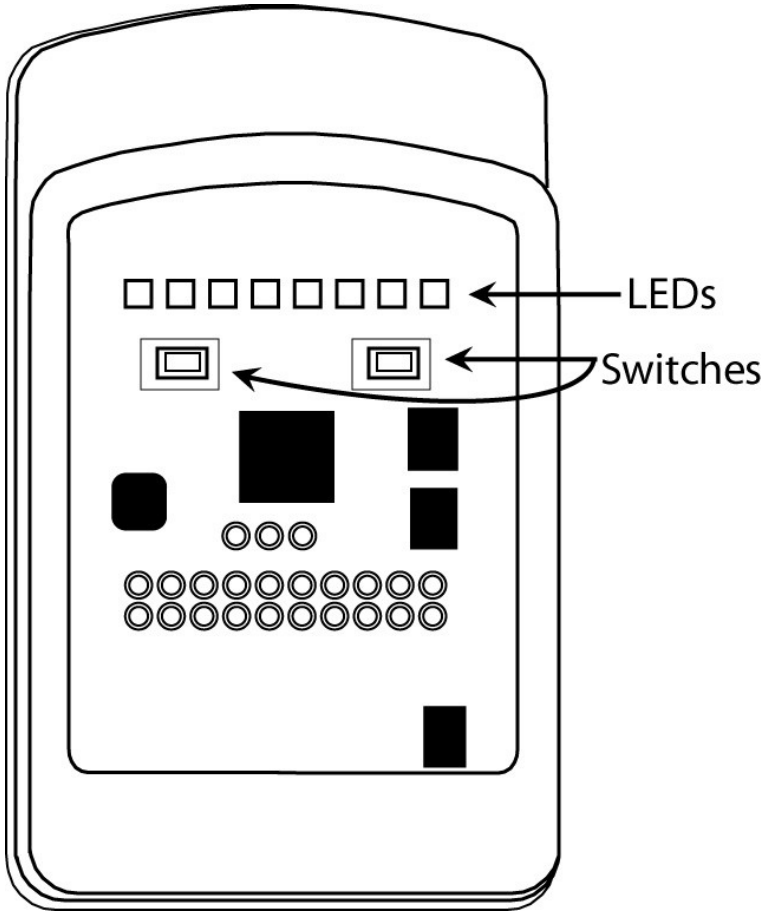
- Basic device has three layers
  - > Battery
  - > Processor Board with Wireless Communication (radio)
  - > Sensor Board(more later)
    - > Stacked on top of processor board
- Processor Board alone acts as a base-station
- User programs the device entirely in Java and standard Java tools (Netbeans)

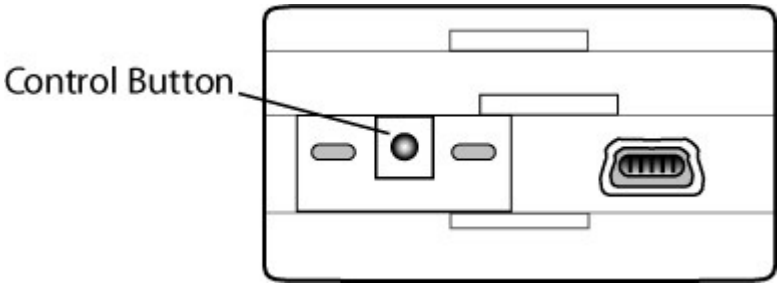


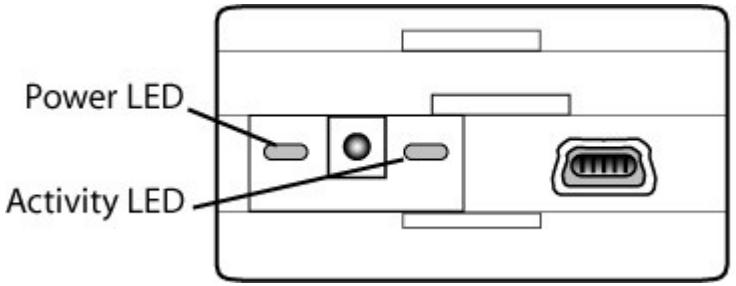
Processor Board

- 180 Mhz 32-bit ARM920T core
  - 512K RAM/4M ROM
- ChipCon 2420 radio
  - 2.4 GHz IEEE 802.15.4 (ZigBee)
- USB interface
- 3.7V rechargeable 750 mAh prismatic lithium ion battery
- 40 uA deep sleep mode, 40 mA to 100+ mA
- 64 mm x 38 mm
- Double sided connector for stackable boards

Sun Spot







Sun Spot

開機：在關機狀態下按一下控制按鈕

關機：在開機狀態下按下控制按鈕 5 秒

重置：在開機狀態下按一下控制按钮

電源指示燈		活動指示燈	
閃爍三次红色	电量不足	快速閃爍綠色	正搜索與主工作站 USB 連接
慢速交替閃爍亮綠和暗綠色	USB 正在充电，CPU 活動狀態	閃爍紅色	正從主工作站收到封包
缓慢閃爍暗綠色	USB 正在充电，CPU 睡眠狀態	閃爍紅色	正向主工作站傳送封包
穩定顯示暗綠色	與 USB 相连，電池充滿，CPU 活動狀態	每 12 秒閃爍兩次綠色	功能正常
穩定顯示暗紅色	電池電量低		
快速閃爍一次綠色	正在重置，重新啟動程序		
一次亮綠色脈衝顯示，快速顯現缓慢消失	電量上升		

Sun Spot

基站： 0014.4F01.0000.1301
移動： 0014.4F01.0000.0FC9
移動： 0014.4F01.0000.0FDD

IEEE Extended Mac Address (IEEE 延伸 Mac 位址)

由四組十六進位數字組成（ nnnn.nnnn.nnnn.nnnn ）
所有 Sun SPOT 的位址前面兩組數字為 0014.4F01
使用 ant slots 命令察看連現在電腦上的 Sun SPOT 的位址

http://en.wikipedia.org/wiki/MAC_address

Sensor Board

- SPOT kits comes with some sensors
- “Out of the box” sensors
  - 2G/6G 3-axis Accelerometer
  - Temperature sensor
  - Light sensor
  - 8 3-colour LEDs
  - 2 push switches
  - A/D input pins
  - 5 General Purpose I/O pins
  - High current (100mA) output pins

Sun SPOT developer's kit

- Two Full Sun SPOTs with eDemoSensor boards and batteries
- One base-station Sun SPOT
- Software
  - > Squawk VM
  - > Java SDK
  - > Netbeans
- USB cable
- $499 introductory price
  - > http://www.sunspotworld.com

Sun SPOT SDK

- Squawk Java VM: Desktop and Sun SPOT
- Libraries
  - Java ME CLDC 1.1 libraries
  - Hardware libraries
    - SPI, AIC, TC, PIO drivers all written in the Java programming language
    - Demo sensor board library
  - Wireless layer libraries
  - Network layer libraries
    - 802.15.4 MAC layer written in Java
  - Desktop libraries

Why Sun SPOT?

- Need for higher level programming language
  - With good development tools and IDEs. eg. Netbeans.
- higher level proramming language
  - Java vs. C
- Separate developer from low level hardware
  - Focus on application functionality and features.

Why Sun SPOT?

- Abstraction of low level details through VM
- Protection of hardware by VM
  - Memory protection from bad code
  - Security protection from malicious code
- 802.15.4 wireless communication
  - upto 250Kbps, ZigBee support
  - adhoc, mesh, cluster tee and star topologies.
- A simple device with sensors and wireless operates at low power

Why Sun SPOT?

- OpenSource !
  (that's why we are here!!)
  - > https://spots.dev.java.net/

Welcome to the SPOTs project

Sun SPOTs are small, Java-based, wireless devices developed at Sun Labs. The SPOTs project is the place on java.net for development of open source code relating to Sun SPOTs. This includes system code, application frameworks, demonstrations, and applications.

It is also the place for educators interested in using Sun SPOTs in their classes. Here they can develop and share instructional materials.

Please visit the various SPOTs subprojects and join in.

- SPOT Applications - demonstration and applications for Sun SPOTs
- SPOT Curriculum - instructional materials using Sun SPOTs
- SPOT Hardware - open source development of the hardware used by Sun SPOTs
- SPOT Libraries - open source development of the libraries used by SPOT applications
- Squawk VM - open source development of the Java virtual machine used by SPOTs

To find out more about Sun SPOTs or to purchase a development kit, see the Sun SPOT World home page.

SPOTs related links

- Sun SPOT World Home Page
- SPOTs documentation
- SPOTs Discussion Forums

Open source Software

squawk - the JVM

spots-core-libraries - Ongoing development of the core libraries for the Sun SPOT

spots-network-library - Ongoing library development for network stack of the Sun SPOT

spots-security - Ongoing library development for security on the Sun SPOTs

spots-sensorboard-library - Ongoing library development for different SPOT sensorboards

solarium - an application for managing Sun SPOTs

spots-sdk-demos - Demo code distributed with the Sun SPOT SDK (AirText, BounceDemo, BuiltInSensorDemo, CalibrateAccelerometer, CodeSamples, DatabaseDemo, EmulatorDemo, HTTPDemo, RadioStrengthDemo, SpotCheck, TelemetryDemo)

spots-midi - Interfacing MIDI to Sunspots

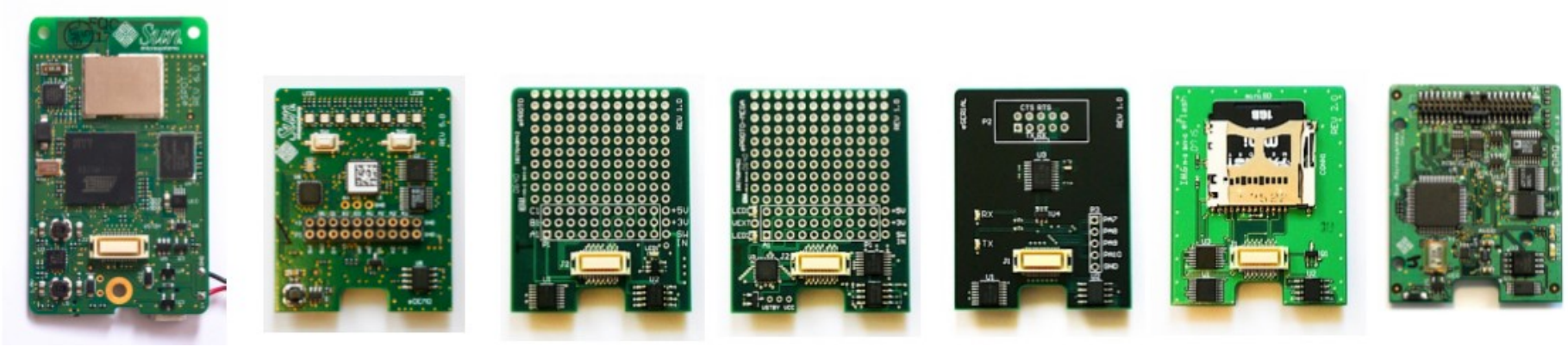
yggdrasil - Reference implementation of a data collection framework

java.net The Source for Java Technology Collaboration

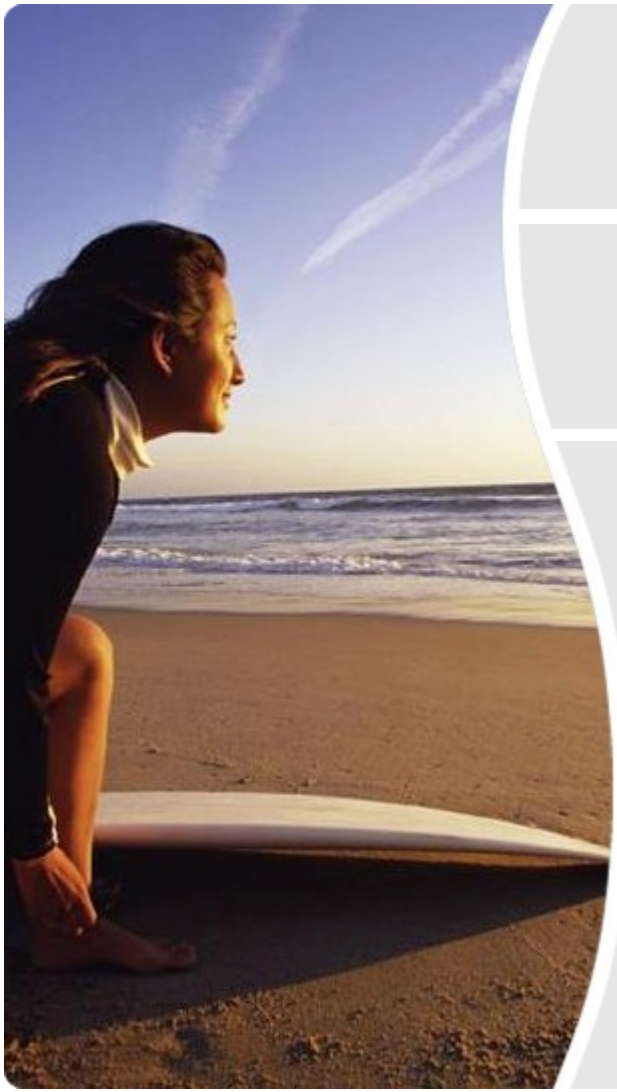
Open source hardware

spot-espot - the Sun SPOT main board
spot-edemo – Demonstration add-on board for Sun SPOT eSPOT

spot-ebones – eBones: information to create add-on boards for the eSPOT
spot-eprtoto – Prototype board with simple interface to SunSPOT eSPOT
spot-eprotomega – eSPOT Prototyping board with Atmega88 processor
spot-eflash – MiniSD card reader/writer add on board for Sun SPOT eSPOT
spot-eserial – RS232 Serial interface board for the SunSPOT eSPOT
spot-eusbhost – USB host interface to Sun SPOT eSPOT
spot-edaq High-resolution Autonomous Data Acquisition board for SunSPOTs



Agenda



Have Fun With Sun SPOT

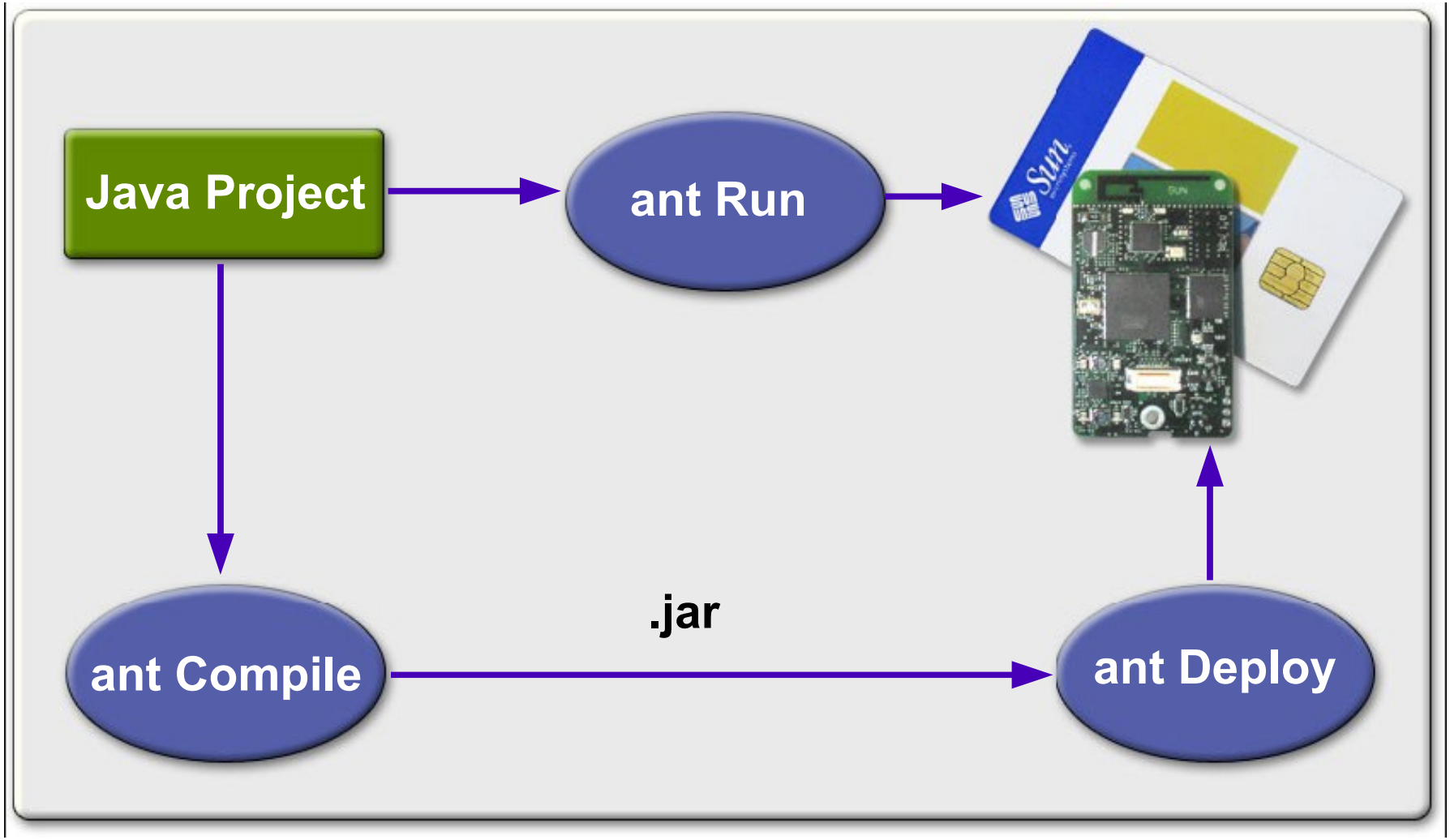
What is Sun SPOT?

The Squawk Java VM

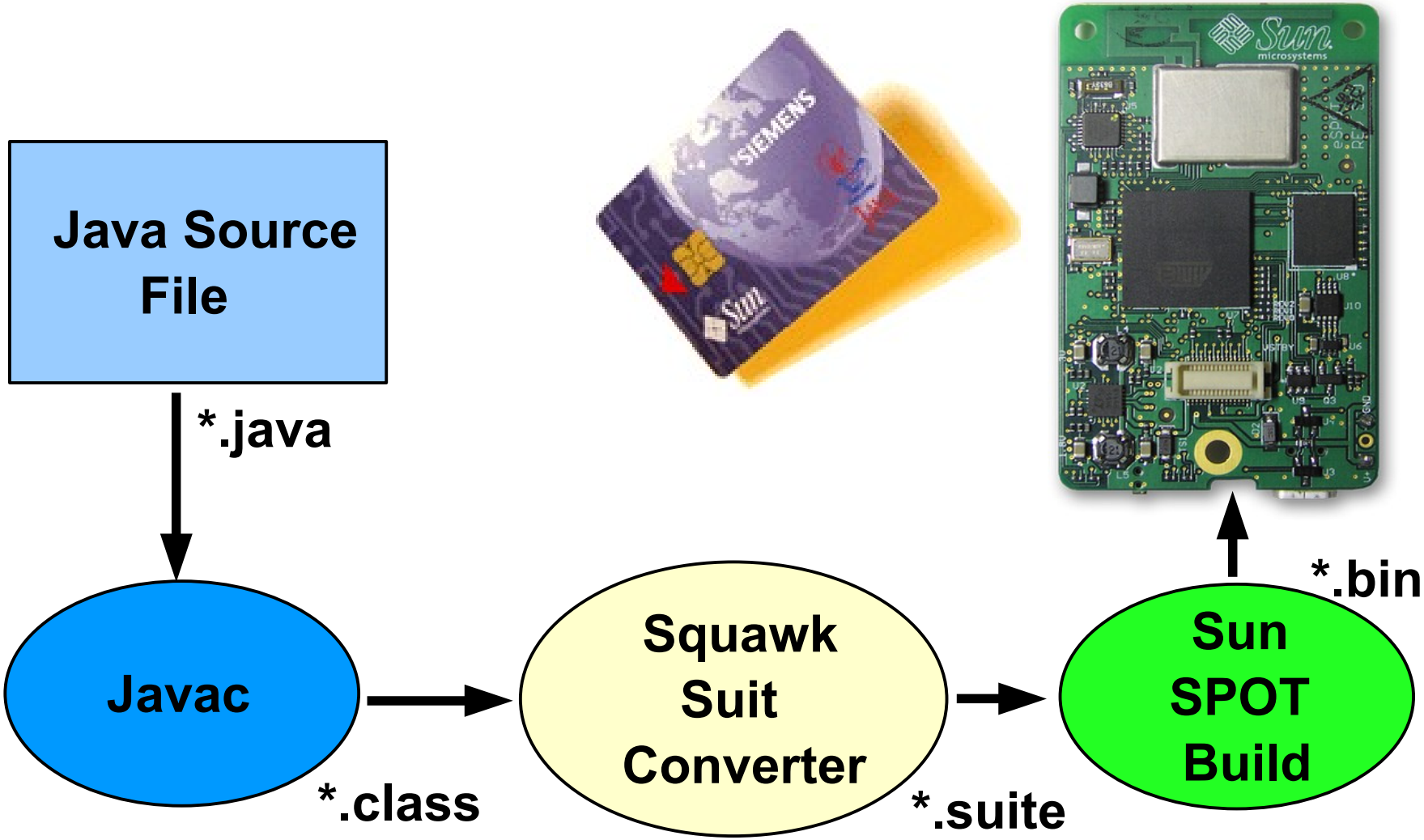
Java on Sun SPOT

What To Do/Where To Go

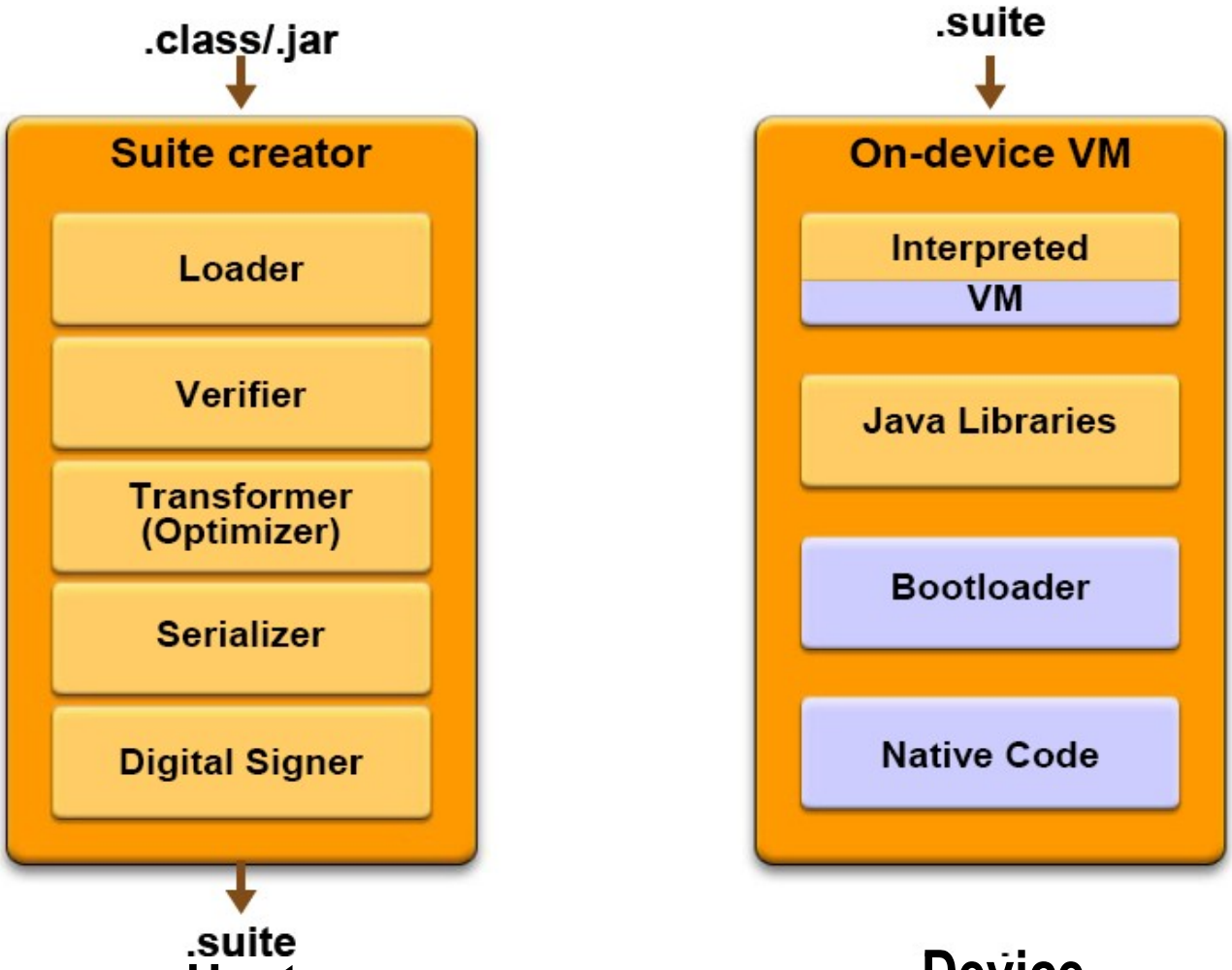
SPOT Development Cycle



Sun SPOT Build and Deploy Flow



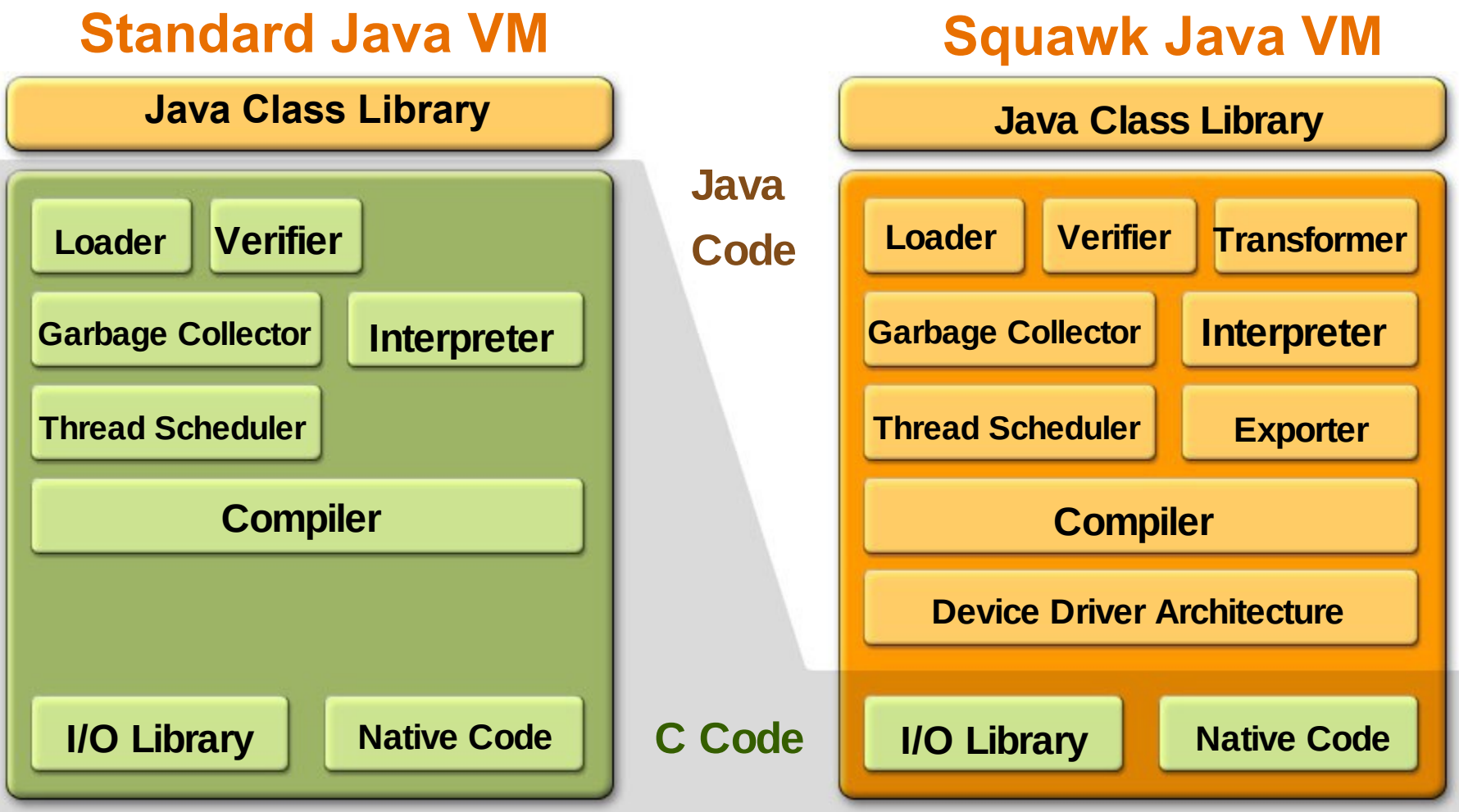
The Squawk Java VM



The Squawk Java Virtual Machine

- Small footprint, highly portable Java VM
  - Runs on the metal
- Most of the JVM is written in Java programming language
  - Interpreter and low level I/O is written in C
  - Everything else is in Java
- Provides a CLDC 1.1 platform
  - Connected Limited Device Configuration
  - Same platform as your mobile phone
  - Additional libraries for sensors, LEDs, bootstrapping, etc.

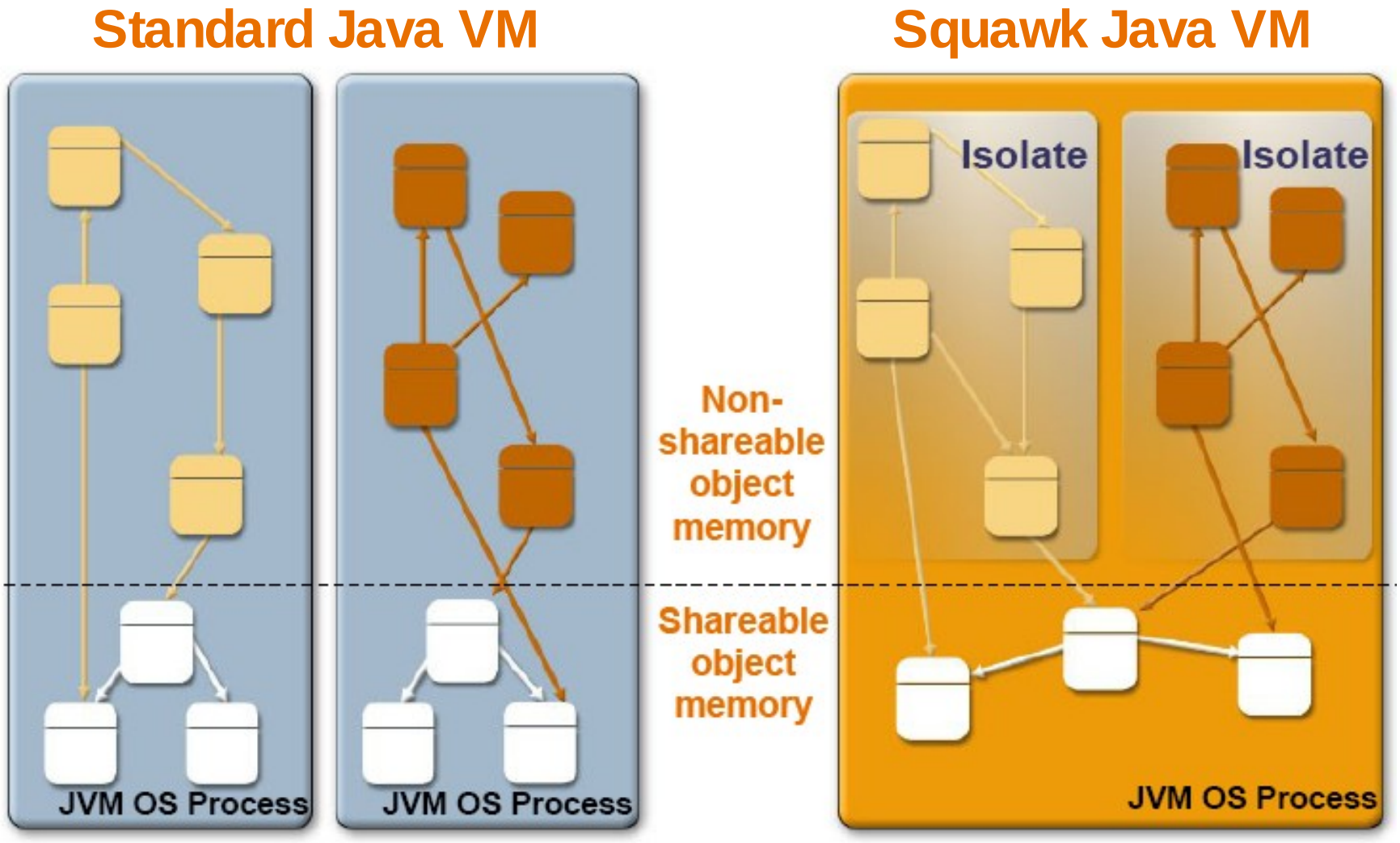
"Regular" vs. Squawk Virtual Machine



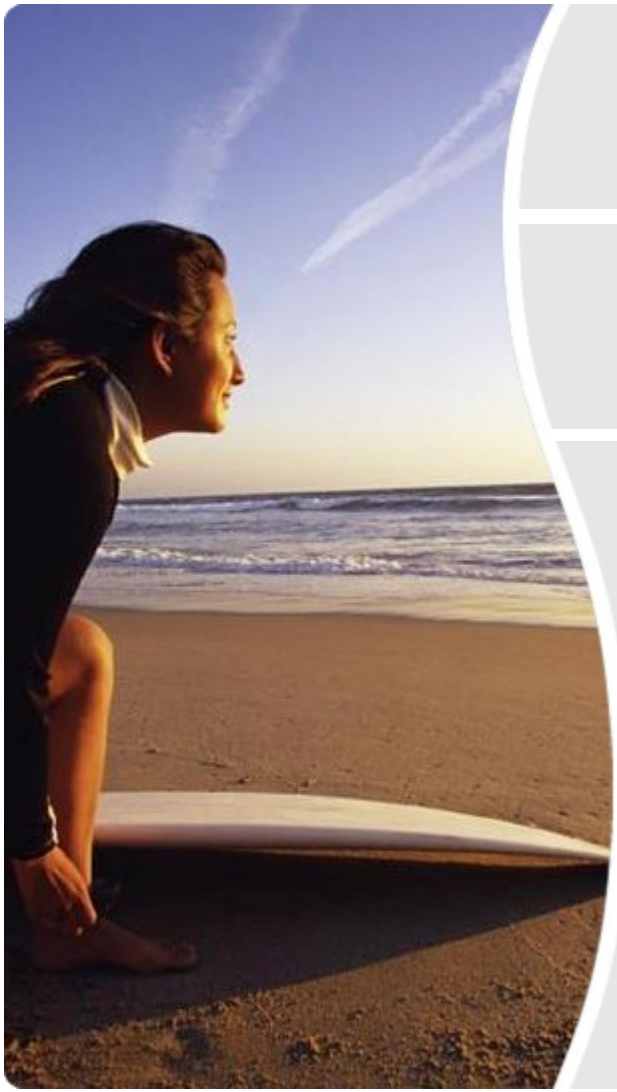
Isolate Application Model

- JSR 121 – Application Isolation API Specification
- Allows running multiple Java applications inside a single Java VM
- Every isolate has its own state for all static variables
- Inter-isolate communication
  - Provides lower level asynchronous message delivery
- Can migrate from one device to another
  - Not a standard Isolate feature

"Regular" vs. Isolates



Agenda



What is Sun SPOT?

The Squawk Java VM

Java on Sun SPOT

Sun SPOT Demo

What To Do/Where To Go

Accessing the Sensors

- EDemoBoard class represents all the sensors on the “out-of-the-box” sensor board
- Singleton – EDemoBoard.getInstance()
- Access to sensors
  - getADCTemperature() - temperature
  - getLEDs() - LEDs
  - getLightSensors() - light sensors
  - getAccelerometer() - accelerometer
  - getSwitches() - push switches

SPOT Programming Example

```
protected void startApp()
    throws MIDletStateChangeException {

    ITriColorLED[] leds =
        EdemoBoard.getInstance().getLEDs();

    leds[0].setColor(LEDColor.GREEN);
    leds[0].setOn(true);

    //Keep the program from exiting
    while (true) {};
}
```

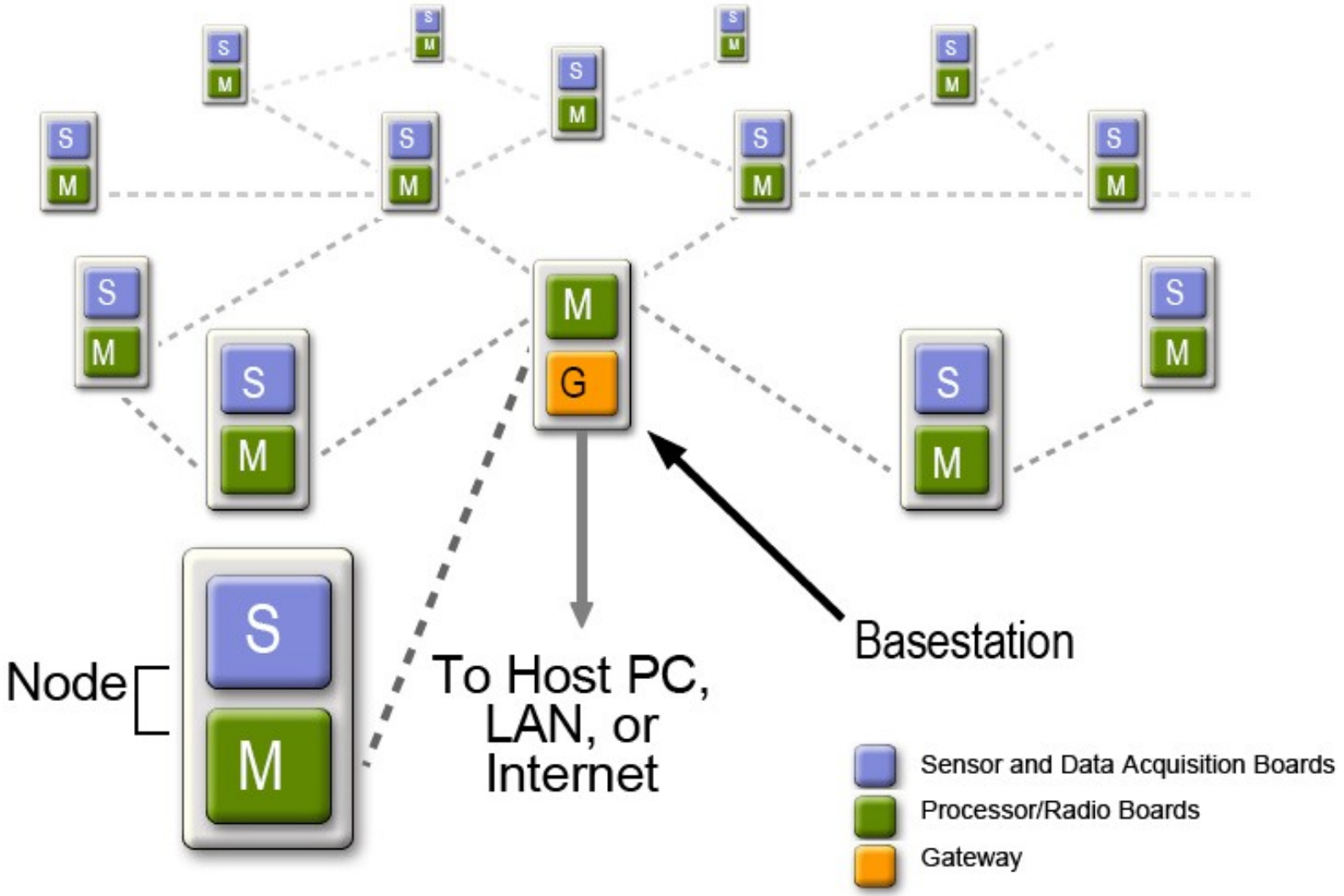
Sun SPOT Sensor Code Snippet

```
RangeInput light = SensorBoard.getLightSensor();
RangeInput temp = SensorBoard.getTemperatureSensor();
ISwitch switch1 = SensorBoard.getSwitch1();
Accelerometer3D accel = SensorBoard.getAccelerometer();
// Set accelerometer to 6G scalling
( (LIS3L02AQAccelerometer) accel).set6GSScale();

SensorBoardColouredLED led1 =
        SensorBoardColouredLED.getLed1();
led1.setOn();
// Change LED colour as SPOT is tilted
int xAccel = accel.getX().getValue();

if(xAccel > 0)
        led1.setRGB(xAccel, 0, 0);
else
        led1.setRGB(0, 0, xAccel);
```

Wireless Sensor Networks



Using the Base Station

- Base station can communicate with SPOT through the radio network
- Two modes of communcation
  - Broadcast and point-to-point
- Each SPOT has a 64 bit IEEE address
- Uses CLDC's GCF to open connection

SPOT Communication Protocol

- Radio Protocol:
  - Socket-like peer-to-peer protocol that provides reliable, buffered streambased IO between two devices

```
RadioConnection conn=
        (RadioConnection)Connector.open
        ("radio://<destinationaddr>:<portNo>");
```

- Radiogram Protocol:
  - Client-server protocol that provides reliable, buffered datagram-based IO between two devices
  - Server:

```
RadiogramConnection conn =
  (RadiogramConnection)
  Connector.open("radiogram://:<portNo>”);
```

SPOT Communication Protocol

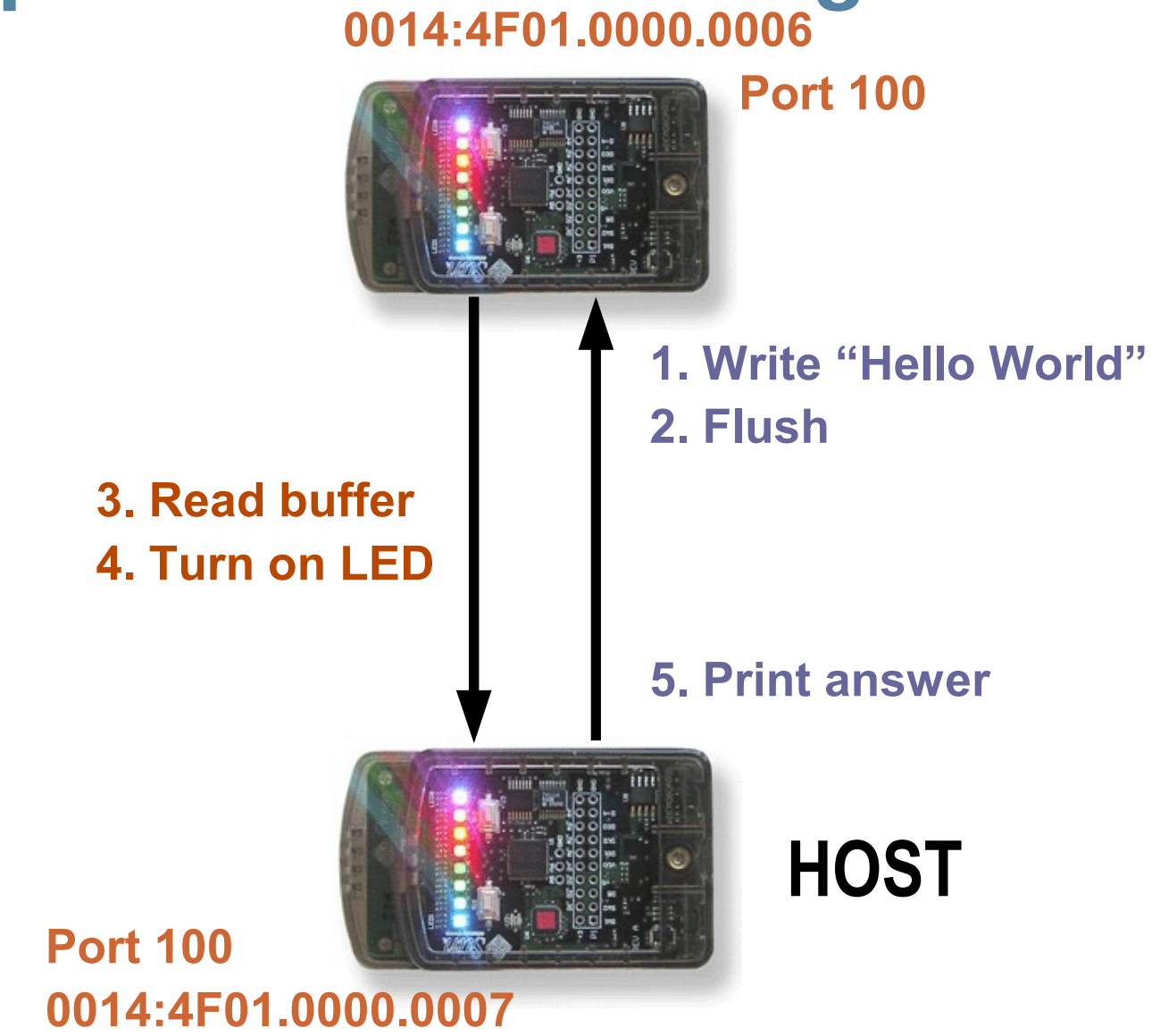
- Radiogram Protocol:
  - Client:

```
RadiogramConnection conn =
  (RadiogramConnection) Connector.open
  (“radiogram://<serveraddr>:<portNo>” ) ;
```

  - Broadcasting:

```
RadiogramConnection conn =
  (RadiogramConnection) Connector.open
  (“radiogram://broadcast:<portNo>” ) ;
```

Example: Welcome Message...



Radio Connection – Host

```
RadioConnection conn = (RadioConnection)
    Connector.open("radio:/ / 0014:4F01.0000.0006:100");

DataOutputStream dos = conn.openDataOutputStream();

dos.writeUTF("Hello World");
dis.flush();

dos.close();
conn.close();
```

Radio Connection – SPOT

```
RadioConnection conn = (RadioConnection)
    Connector.open("radio:/ / 0014:4F01.0000.0007:100");

DataInputStream dos = conn.openDataInputStream();

String msg = dis.readUTF();
//Turn on LED

dis.close();
conn.close();
```

Radiogram Connection – Host

```
DatagramConnection conn = (DatagramConnection)
    Connector.open("radiogram:/ / :100");

//Creates a datagram with the maximum size
Datagram dgram = conn.newDatagram(conn.getMaximumLength());

dgram.writeUTF("Hello World");
conn.send(dgram);

conn.close();
```

Radio Connection – SPOT

```
DatagramConnection conn = (DatagramConnection)
    Connector.open("radiogram:/ / 0014:4F01.0000.0007:100");

Datagram dgram = conn.newDatagram(conn.getMaximumLength());

conn.receive(dgram);
String msg = dgram.readUTF();

conn.close();
```

Broadcasting

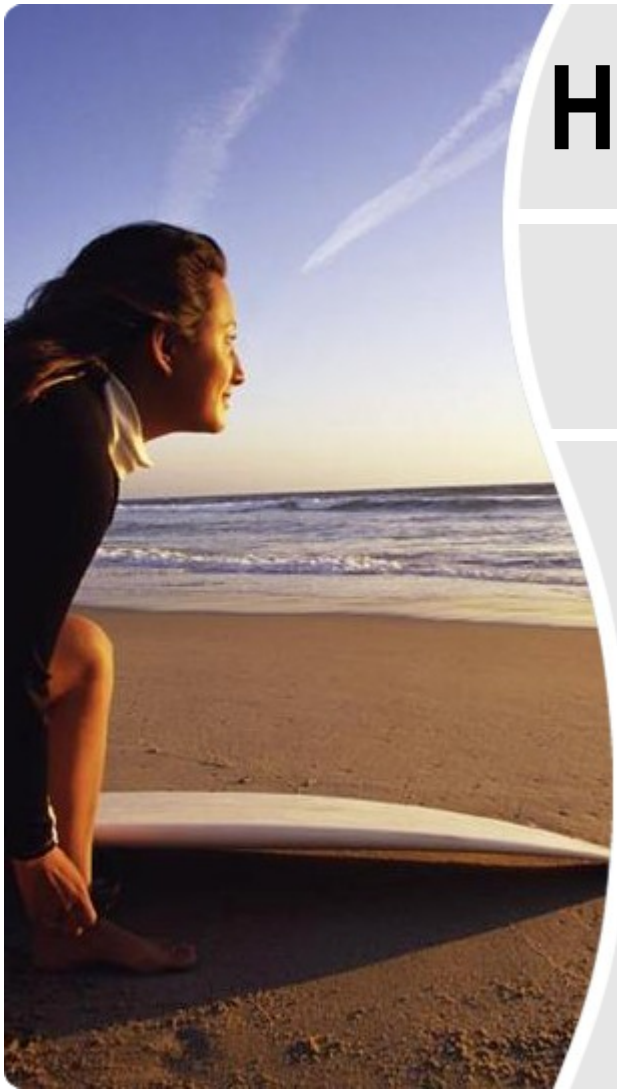
Host

```
DatagramConnection conn = (DatagramConnection)
    Connector.open("radiogram:/ / broadcast:100");
//As sending datagram
```

SPOT

```
DatagramConnection conn = (DatagramConnection)
    Connector.open("radiogram:/ / :100");
//As receiving datagram
```

Agenda



Have Fun With Sun SPOT

What is Sun SPOT?

The Squawk Java VM

Java on Sun SPOT

What To Do/Where To Go

Project Blackbox

Monitoring a Mobile Data Center

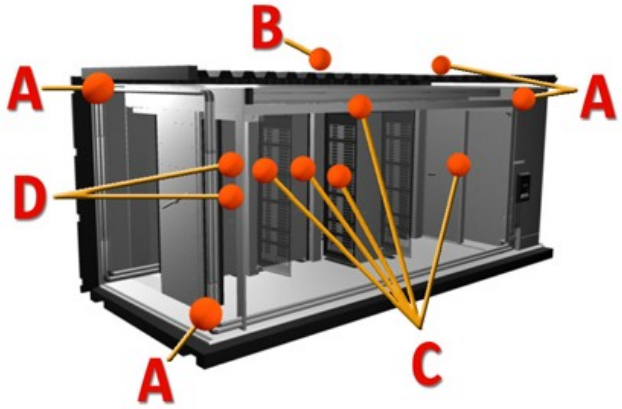
Sun SPOTs on Blackbox

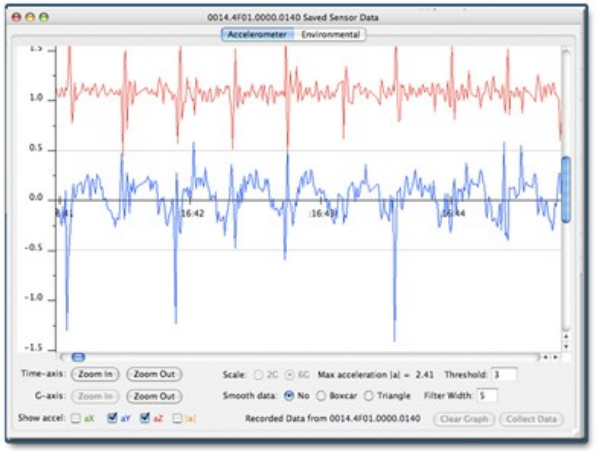
A 3D Accelerometer. Rigid body simulation: records movement of the Blackbox container in transit.

B GPS. Tracks Blackbox location.

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

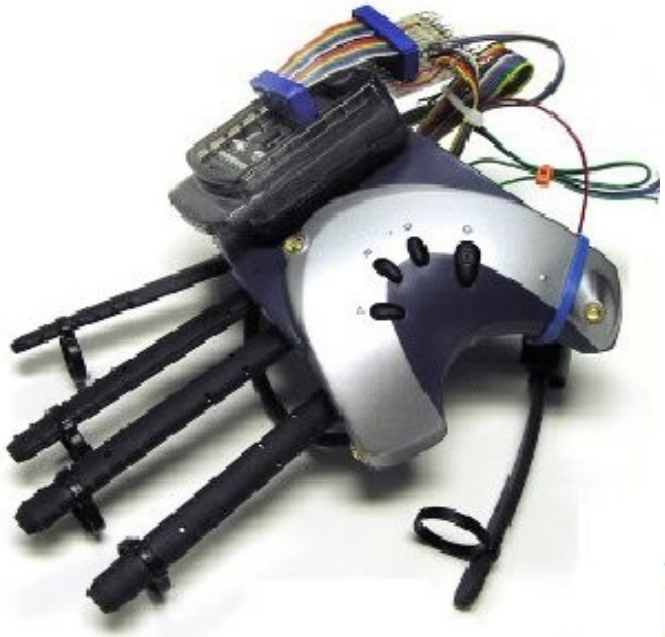
D Data loggers. Redundant data collection.





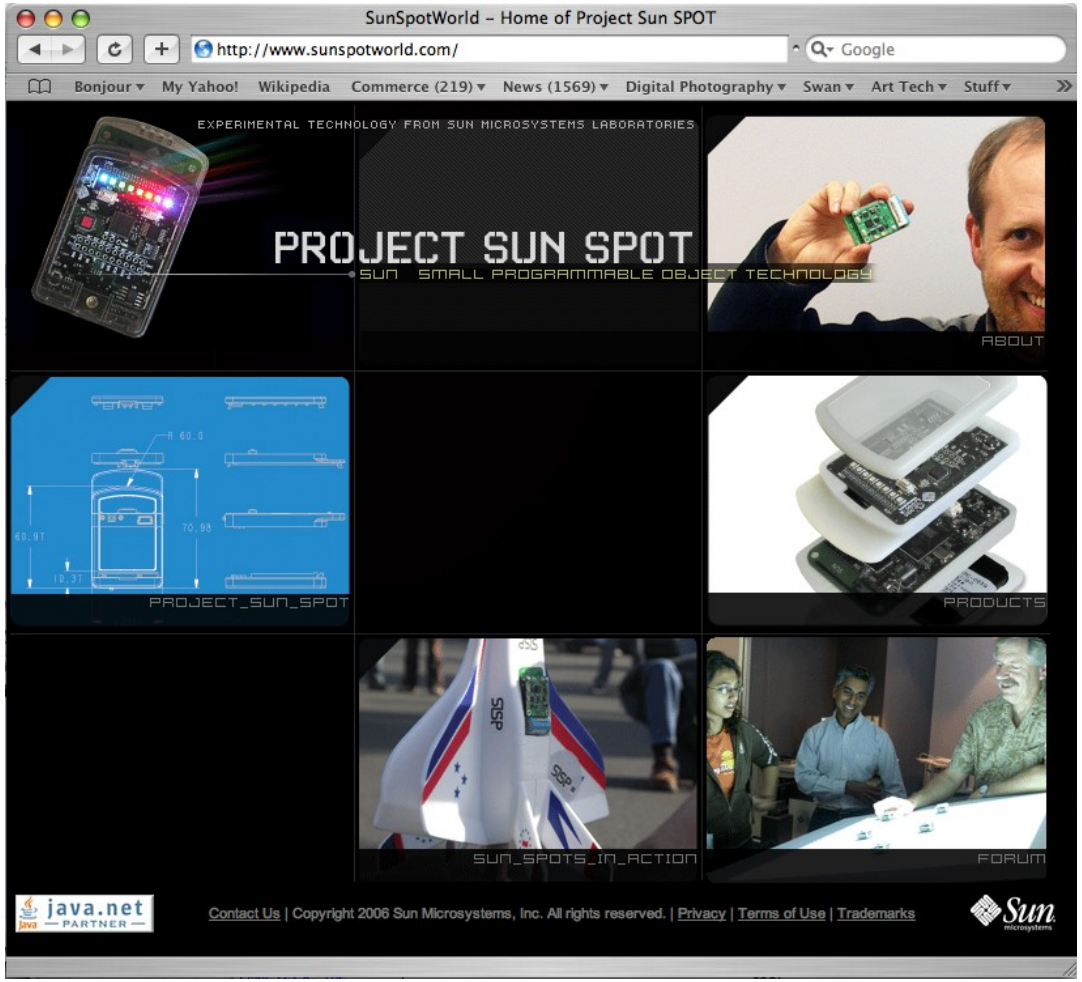
Visit http://www.sunspotworld.com/

- Submit a proposal for classroom curriculum using Sun SPOTS
- Obtain a Sun SPOT development kit
- Watch some cool Sun SPOT video demos
- Participate in Sun SPOT forum
- Create something cool!



Web Site

http://www.sunspotworld.com



- More Information
- Forums
  - for questions
- Mailing list
  - for latest information

Sun SPOT Resources

- Project Sun SPOT http://www.sunspotworld.com/
  - Sun SPOT documents and applications notes
  - Short Video demos
  - Classroom Curriculum
  - Discussion Forum
- David's Blog: http://blogs.sun.com/davidgs/
  - Latest news, other informal information on Sun SPOT
- The Squawk Project http://research.sun.com/projects/squawk/
  - Sun Microsystems official Squawk research project page
- NetBeans IDE
  - Free download software development tool for Sun SPOT.

Project Sun SPOT

http://www.sunspotworld.com

THANK YOU!

cookys (林秉毅)

CA of Feng Chia University

cookys@gmail.com