



# Project Sun™ Small Programmable Object Technology Sun SPOTs

**David G. Simmons**

Staff Engineer, Project Sun SPOTs  
Sun Microsystems Laboratories

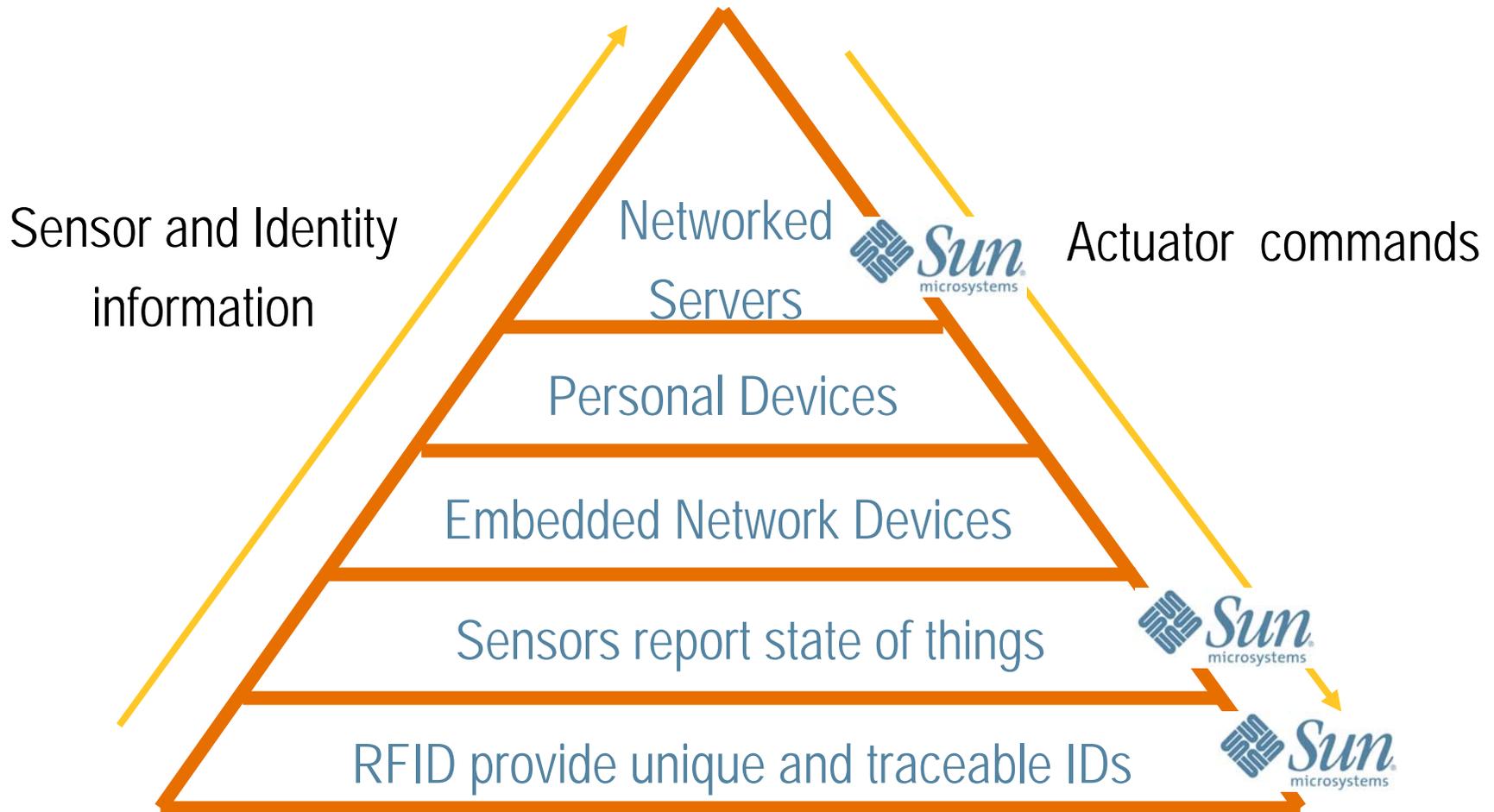


# What is a Sun SPOT?

- A Sun Labs research project to investigate small wireless sensor technology
  - > Powerful 32-bit processor (180 MHz ARM9)
  - > 2.4 GHz 802.15.4 radio with antenna
  - > Sensors
    - > 3-axis 2G/6G Accelerometer
    - > Light sensor
    - > Temperature sensor
    - > 8 tri-color LEDs for display
    - > 5 GPIO pins for external I/O control
      - 4 high-current I/O pins
    - > 2 Switches
  - > 6 analogue inputs
  - > USB interface



# The new internet of things



# A platform for inspiration

- Think back to the internet in 1990
  - > What would it be used for?
  - > Where would it take us?
- Innovation happens elsewhere
  - > Basements
  - > Garages
  - > Classrooms and education/research labs
- Community is the catalyst for inspiration
  - > Students and Educational Institutions support major community development efforts

# Program the World

## The Internet of Things

- Provide a platform to allow programming of the next emerging class of devices
  - > Java on 1 Billion cell phones
  - > The sensor market dwarfs the cell phone market
- Provide the ease of Java™ for development, deployment and debugging
  - > Stable, robust and widely-accepted language and tools
- Provide powerful tools for rapid prototyping, development, deployment and debugging of Sensor Applications

# What sorts of applications?

- Prototype gadgets
- Wireless mesh/adhoc networking
- Robotics
- Monitoring and reporting
  - > Industrial applications
    - > Manufacturing
    - > Environmental monitoring
- Monitoring and effecting
  - > Autonomous environmental control
  - > Pollution monitoring and reporting
- Military and security applications

# Make it a software problem

- Proper software/hardware environment can do for small devices what the Apple II did for computers
  - > Before Apple II, computer hobbyists needed to own a soldering iron
  - > After Apple II there were software hobbyists
  - > Abstracted the hardware from the solution
- With the proper hardware environment
  - > Focus on the solution, not the process
  - > Quickly implement a prototype
  - > Abstract the hardware from the solution
  - > Non EE students can build operational devices
- With the proper software tools
  - > Quickly get to working models

# What is Squawk?

- A Sun Labs software research project into Java on small devices
  - > Fully capable J2ME CLDC 1.1 Java OS
  - > Executes directly out of on-board flash memory
  - > Complete set of native Java device drivers
  - > Automatic power management
    - > Brings to power and battery management what Java brought to memory management
  - > Bring the ease of Java development to the world of sensors
    - > Compatible with IDEs for easy development and deployment
  - > Rapid prototyping, development and deployment of sensor applications using Java

# Advanced tools

- Works with Java 1.4.2 and 1.5
- Builds and deploys using Ant
- Fully integrated with NetBeans Integrated Development Environment (IDE)
  - > Build applications
  - > Deploy applications
  - > Debug applications
- No barriers to other IDEs

# Sample Translator Application

- How many know Morse code?
  - > We need a translator
- Morse code to Semaphore
  - > Not very useful, but the point is ...
- Simple application, built by software engineers
  - > No soldering
  - > Off the shelf parts
  - > Extremely fast development time

# Anatomy of an application

## Setting up radio communications

```
public void setConnection() {
    String url = "radiogram://broadcast:44";
    LowPanPacketDispatcher.getInstance().setChannelNumber(26);
    LowPanPacketDispatcher.getInstance().setPanId(panID);
    try{
        conn =(DatagramConnection)Connector.open(url);
    } catch(IOException e){
        e.printStackTrace();
    }
}
```

# Anatomy of an application

## Setting up the sensor board

```
public void initSensorBoard() {  
    IRangeInput  accelerator;  
    IRangeInput  lightSense;  
    IRangeInput  tempSense;  
    IColoredLed[] leds;  
    ISwitch sw1, sw2;  
  
    accelerator=DemoSensorBoard.getInstance().getAccelerometer();  
    lightSense=DemoSensorBoard.getInstance().getLightSensor();  
    tempSense=DemoSensorBoard.getInstance().getTemperatureSensor();  
    leds=DemoSensorBoard.getInstance().getColoredLedArray();  
    sw1 = DemoSensorBoard.getInstance().getSwitch1();  
    sw2 = DemoSensorBoard.getInstance().getSwitch2();  
}
```

# Anatomy of an application

## Getting input from the sensor board

```
public void getSensorValues() throws IOException {
    int accelValue;
    int lightValue;
    int tempValue;
    boolean switch1, switch2;

    try{
        accelvalue = accelerator.getValue();
        lightValue = lightSense.getValue();
        tempValue = tempSense.getValue();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
    switch1 = sw1.isOpen();
    switch2 = sw.isClosed();
    sw1.waitForChange(); // blocks waiting for the switch
}
```

# Anatomy of an application

## Showing output on the LEDs

```
public void setLEDs() {  
  
    for(int x = 0; x < leds.length; x++){  
        leds[x].setOn();  
        leds[x].setRGB(100,0,0);  
    }  
}
```

# Anatomy of an application

## Getting external input/output

```
public void getGPIOInput() {
    ISingleBitInput[] inputPins = new ISingleBitInput[4];
    ISingleBitOutput[] outputPins= new ISingleBitOutput[4];
    for(int x = 0; x < inputPins.length; x++){
        inputPins[x]=DemoSensorBoard.getInstance().getInputD(x);
        outputPins[x]=DemoSensorBoard.getInstance().getOutputDO(x);
    }
    // this will generate a 100ms pulse on pin 0
    for(int z = 0; z < 100; z++){
        outputPins[0].setHigh();
        Thread.sleep(100);
        outputPins[0].setLow();
    }
    if(inputPin[0].isHigh())
        System.out.println("Got a signal on pin 0");
}
```

# That's all there is!

- All Java
- Simple interfaces
- Easy, rapid development
- Library source code included
  - > Modify as needed
  - > Write your own

# Questions?

- More information
  - > <http://www.SunSpotWorld.com/>
- Questions and support
  - > <http://www.SunSpotWorld.com/forums>





# Sun SPOTs Wireless Sensors

David G. Simmons  
davidgs@sun.com





# Sun Small Programmable Object Technology Sun SPOTs

<http://www.SunSpotWorld.com/>

Government Contacts: Jean Edwards  
[jean.edwards@sun.com](mailto:jean.edwards@sun.com)

Dennis McLain:  
[dennis.mclain@sun.com](mailto:dennis.mclain@sun.com)

David G. Simmons  
Staff Engineer, Sun Laboratories  
[davidgs@sun.com](mailto:davidgs@sun.com)

