



Java is a trademark of Sun Microsystems, Inc.



JavaOneSM

Sun SPOTs: A Great
Solution for Small Device
Development

Claudio M. Horvilleur
Cromasoft, S.A. de C.V.
México

Agenda

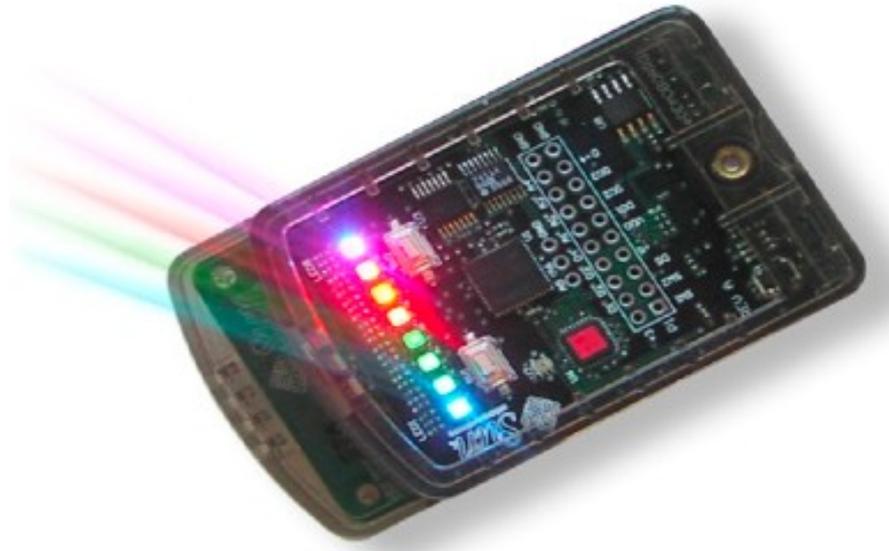
- > What is the Problem?
- > What is a SunSPOT?
- > Adding a New Sensor
- > Hardware Development Issues
- > Adding a New Board
- > Summary

What is the Problem?

- > Device Development issues:
 - Expensive
 - Long Term Projects
 - Requires Specialized Skills
 - Difficult to Debug

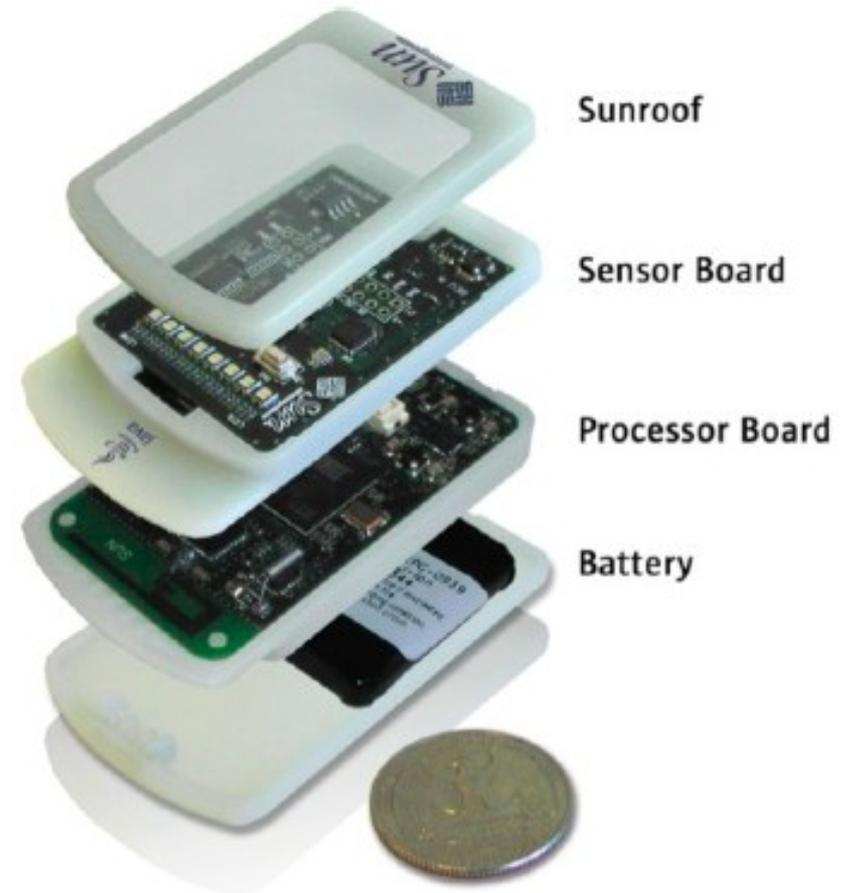
What is a SunSPOT

- > Sun Small Programmable Object Technology
- > Experimental Technology from Sun Labs



What is a SunSPOT

- > Three Basic Elements
 - Battery
 - Processor Board
 - Sensor Board
- > A *base-station* is just a processor board
- > Programmed in Java with standard tools



What is a SunSPOT

Processor Board / Battery

- > 32 bits ARM920T processor @180 MHZ
- > 512 KB of RAM / 4 MB of Flash
- > Radio 802.15.4
- > USB interface
- > Rechargeable Battery (lithium)
- > *Deep Sleep* mode (36uA)



What is a SunSPOT

Sensor Board

- > 3 axes accelerometer 2G/6G
- > Light sensor
- > Temperature sensor
- > Buttons (2)
- > 24 bits RGB LEDs (8)
- > Analog inputs (6)
- > General I/O pins (5)
- > High current output pins(4)



What is a SunSPOT Software

- > Squawk Virtual Machine
 - J2ME CLDC 1.1 IMP Profile
 - Java-on-metal
 - Designed for limited memory devices
 - Mainly written in Java
 - Isolates applications
 - May run on Solaris, Linux, Mac and Windows



What is a SunSPOT Software

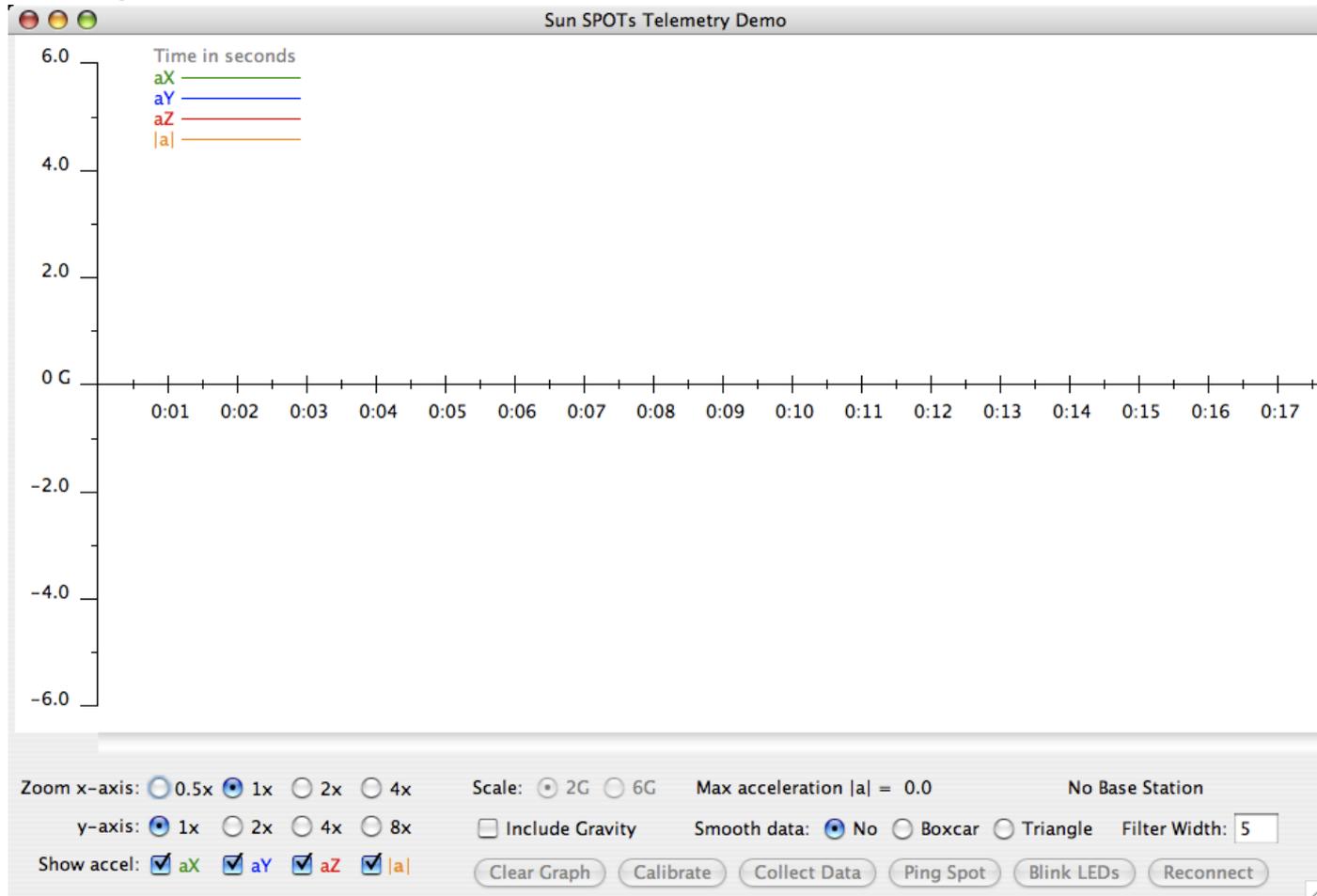
> Transducer library

- Supports all the sensors for the edemoBoard
- Includes energy administration for optimizing the battery life.
- Includes *Mesh Networkng*

> Fully Open Source

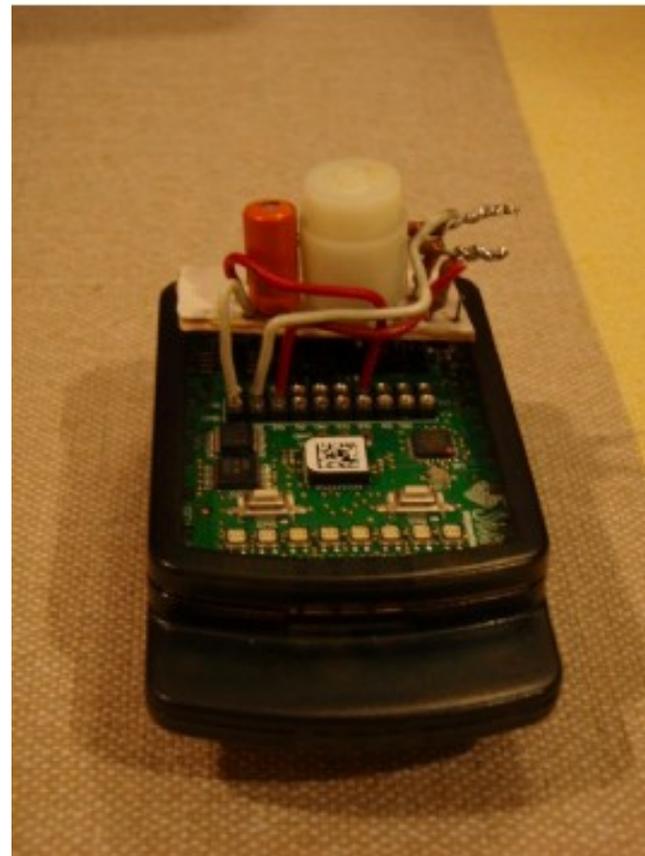
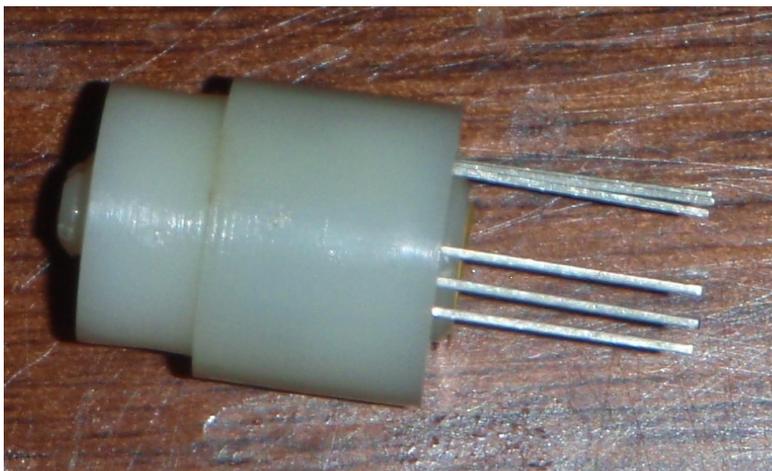
- Squawk (<http://squawk.dev.java.net>)
- SunSpot (<http://spots.dev.java.net>)
-

What is a SunSPOT Telemetry Demo



Adding a New Sensor

An Electronic Compass (Robson R1655)



Adding a New Sensor

Reading the Data

> Initialization

```
EDemoBoard demo = EDemoBoard.getInstance();  
IScalarInput[] analogInputs = demo.getScalarInputs();
```

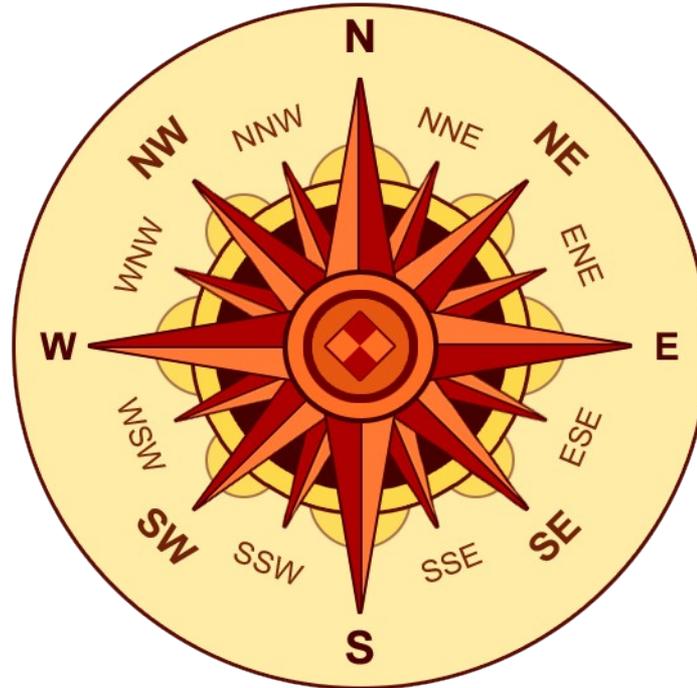
> Getting the data

```
Sender.send(  
    analogInputs[EDemoBoard.A2].getValue(),  
    analogInputs[EDemoBoard.A3].getValue());
```

Adding a New Sensor

Compass Demo

Where is the North?



Hardware Development Issues

Standard Process

- > Design and implementation of a board
 - Have to start with a micro-processor
- > Firmware programming
 - Signal management
 - Data I/O
 - DMA
 - Multithreading / Multitasking
 - Functionality

Hardware Development Issues

Standard Process

- > Software development in:
 - Assembler
 - “C” with a cross-compiler
- > EEPROM programming
- > Debugging with a signal analyzer

Hardware Development Issues

Follow the Trail - “Por las Piedritas”



Hardware Development Issues

With a SunSPOT

- > Design and implementation of a board
 - Start with a base platform
- > Firmware programming
 - Included in the SunSPOT
 - Signal management
 - Data I/O
 - DMA
 - Multithreading / Multitasking
 - Functionality

Hardware Development Issues

With a SunSPOT

- > Software development in Java
- > Download to the SunSPOT with a simple “deploy”
- > Debugging from the development environment

Hardware Development Issues

Follow the Highway



Adding a New Board Specs

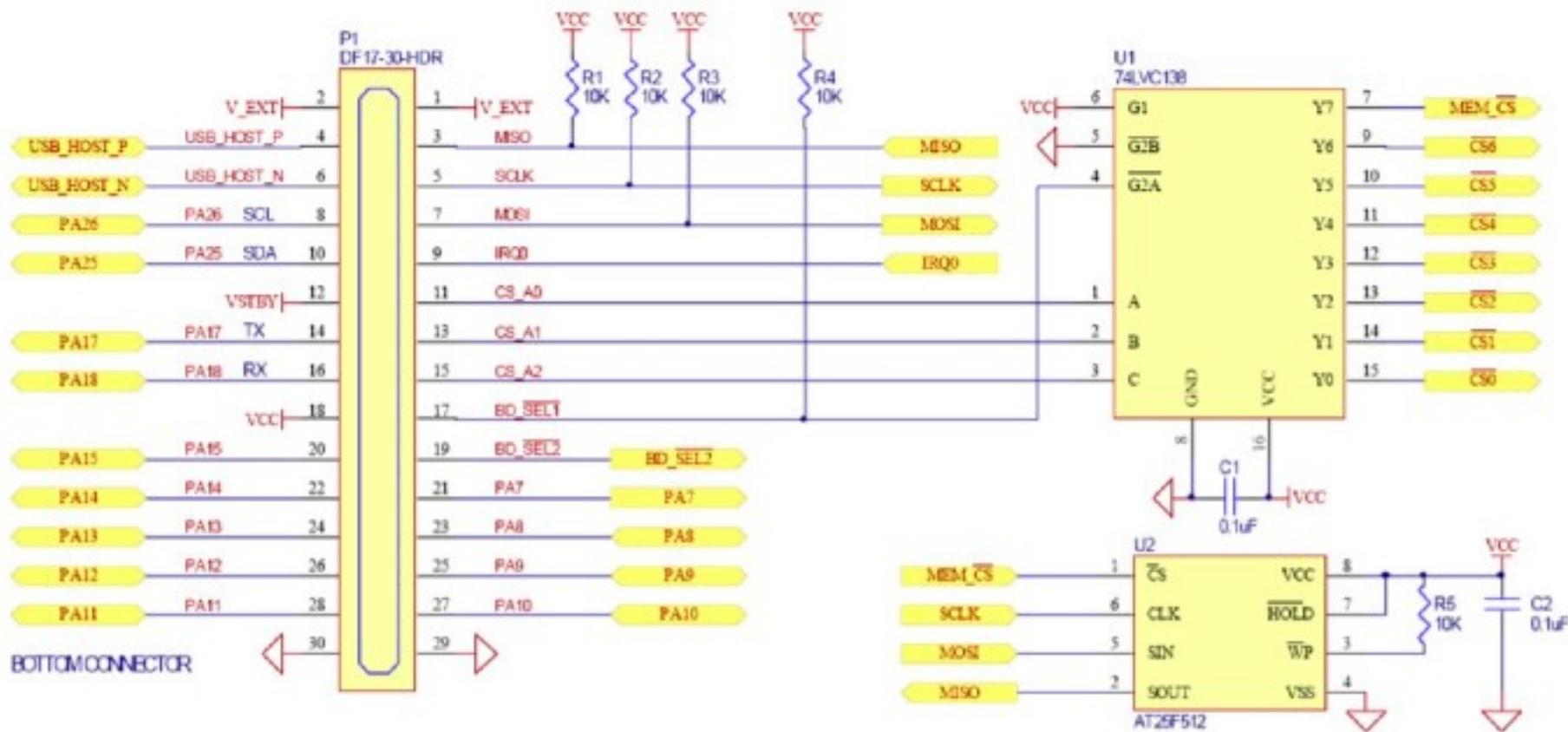
- > Board with as many signals and interfaces
 - RF-ID reader from *SkyeTek* (SkyModule M1-Mini with SPI or Serial interface)
 - One-Wire interface from *Dallas Semiconductor* *Maxim* (serial)
 - SD card reader
 - USB-HOST interface

Adding a New Board

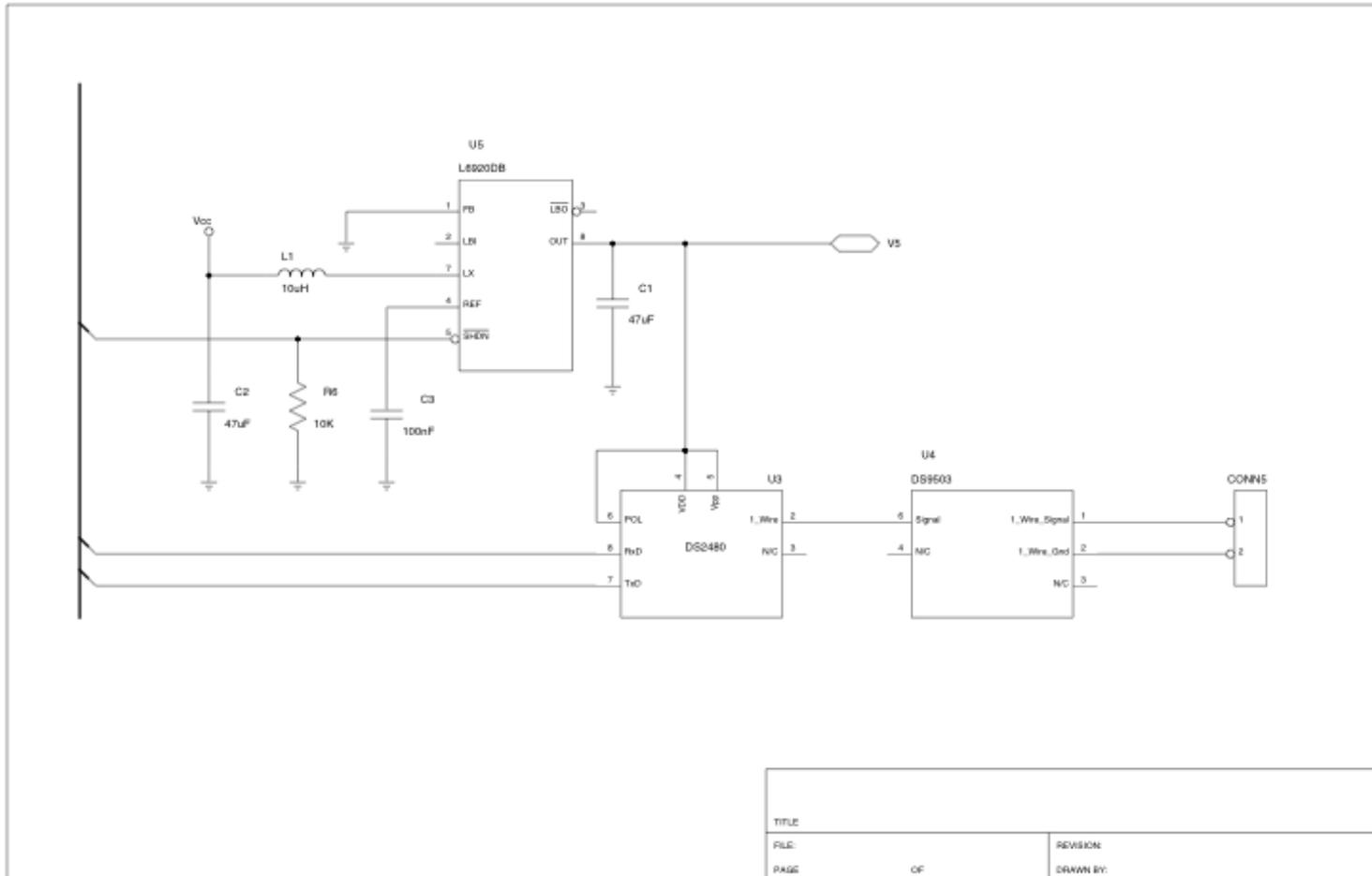
The board

- > Design Software
 - gEDA (<http://www.gpleda.org/>)
- > Start with the “eBones”
- > Add your devices
- > Build the board
 - Print the circuit board (PCBexpress <http://pcbexpress.com>)
 - Solder the elements

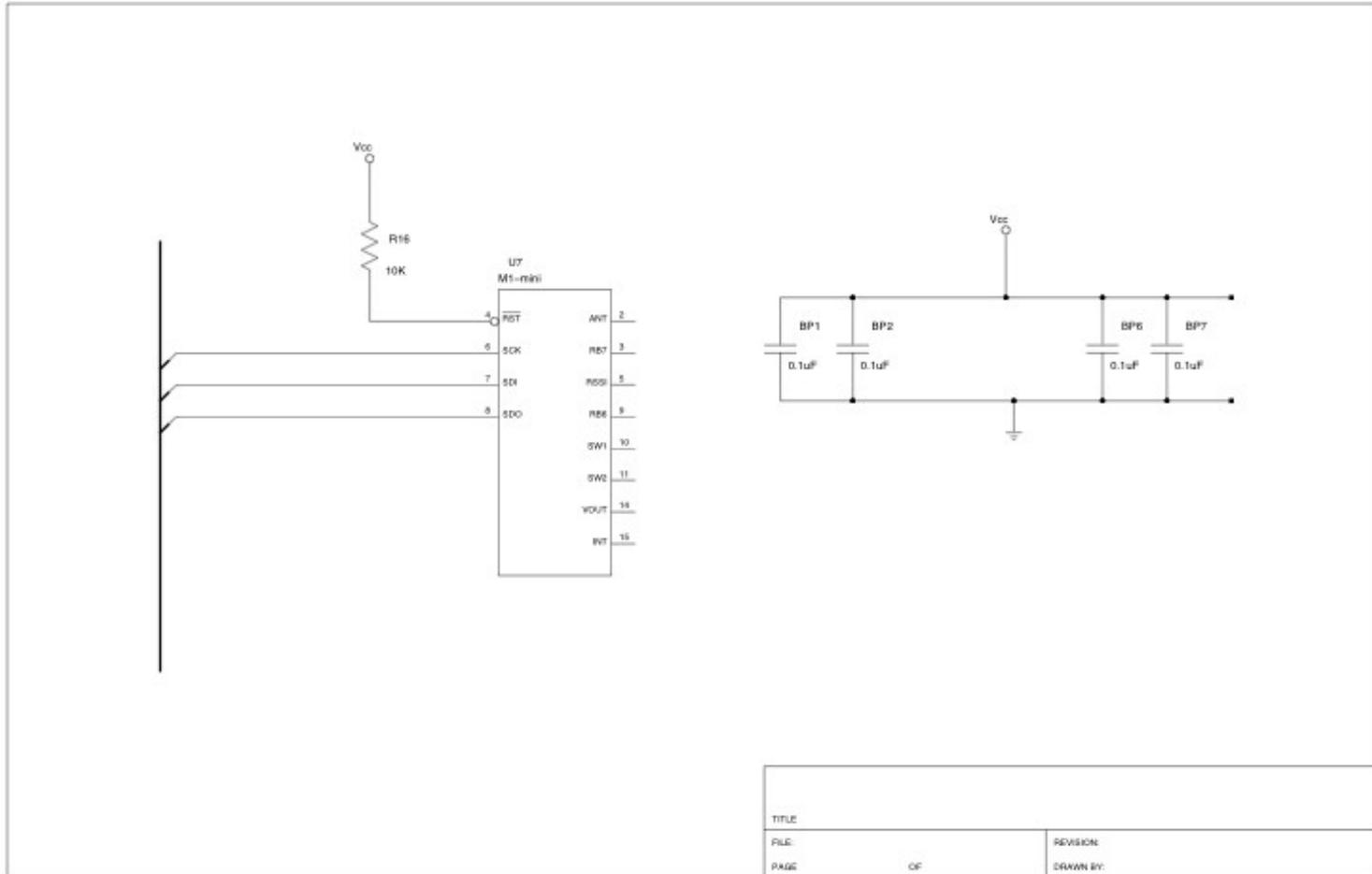
Adding a New Board eBones



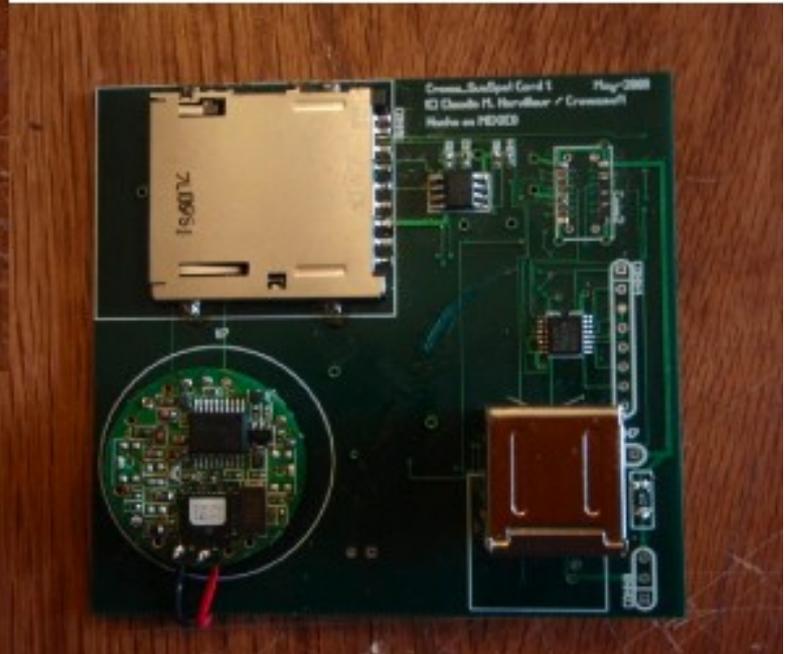
Adding a New Board Serial Device



Adding a New Board SPI device



Adding a New Board Board



Adding a New Board Software

- > Start with the device's vendor software
- > Adapt the I/O specific classes
 - Port Setup
 - Data I/O

Adding a New Board

Serial I/O Programming

> Device Setup

```
serialOutput = Connector.openOutputStream(  
    "serial://usart?baudrate=9600");  
serialInput = Connector.openInputStream(  
    "serial://usart");
```

Adding a New Board

Serial I/O Programming

- > Data output

```
serialOutput.write(data);  
serialOutput.flush();
```

- > Data output

```
if (serialInput.available() == 0) { ... }  
byte start = serialInput.read();  
int actualBytes =  
    serialInput.read(respBuffer, 0, respLen);
```

Adding a New Board

SPI Programming

> Device setup

```
spi = newBoardDeviceSPI(2,  
    (ISpiMaster.CSR_MODE0 |  
    ISpiMaster.CSR_BITS_8 |  
    ISpiMaster.CSR_SCBR_250K |  
    ISpiMaster.CSR_DLYBCT_200));
```

Adding a New Board

SPI Programming

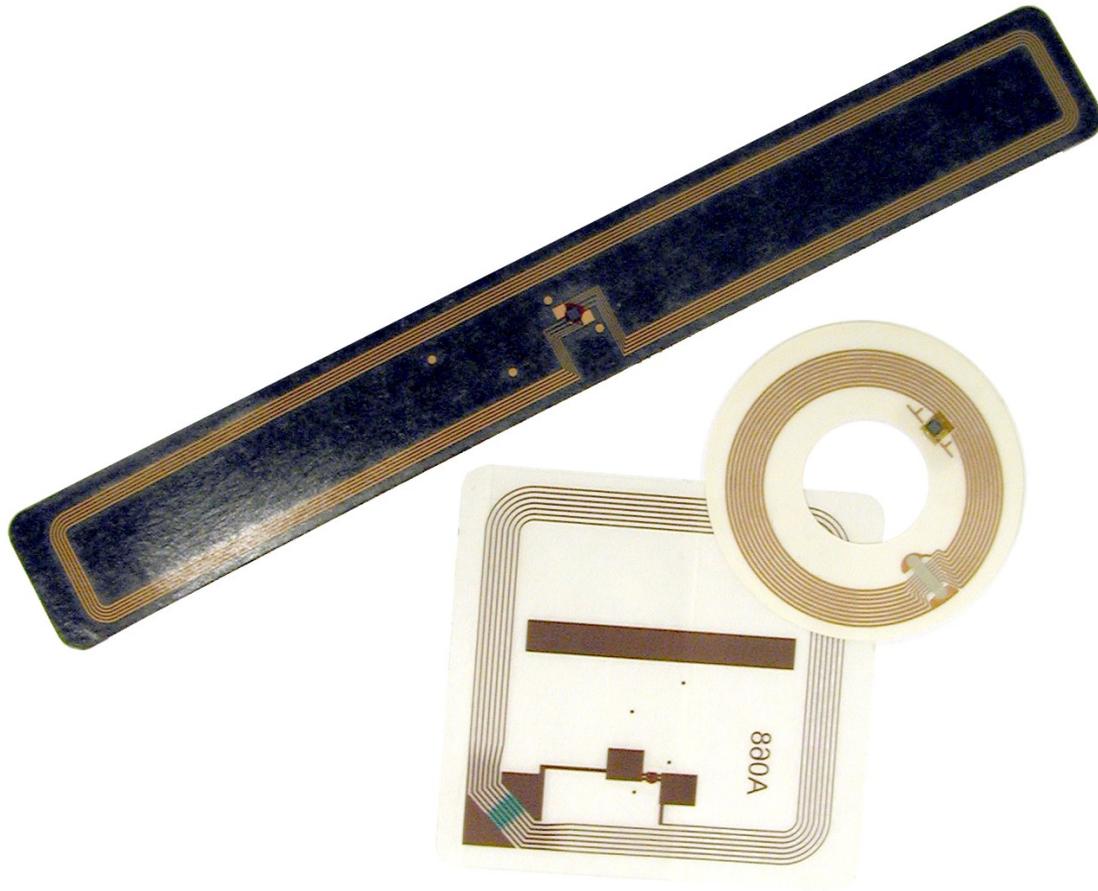
- > Data output

```
spi.sendSPICommand(req, req.length, null, 0);
```

- > Data input

```
spi.sendSPICommand(nul, 0, resp, resp.length);
```

Adding a New Board RF-ID Demo



Summary

- > The SunSPOT is a very usefull platform for small devices development.
- > Device development does not need to be expensive
- > It can be a short term project
- > If you know Java you can develop firmware
- > You can debug it as any other software

Summary

From a Forest



Summary ... to a Rose Garden



Summary Q&A





JavaOneSM

Thank You

Claudio M. Horvilleur
claudio@croma.com.mx

