Sun SPOTs

Bernard Horan
Senior Staff Engineer
Sun Microsystems Laboratories

Sun Small Programmable Object Technology (SPOT)

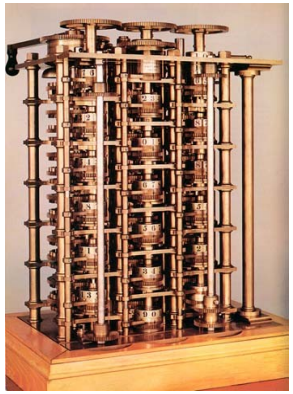
- Sun has licensed Java™ technology to over 1 billion cell phones
- How do we encourage Sun technology in whatever comes next?

Built to Inspire

Wireless Sensor Networks

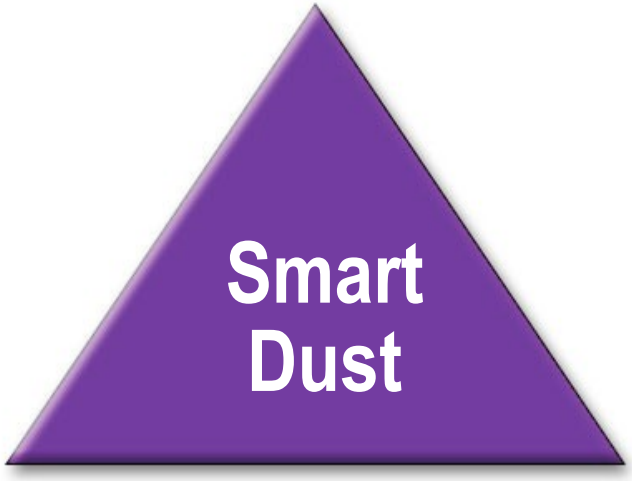
Sensors

Miniaturization, MEMS, lower power, and cost



Computing Power

Results of Mead's law



Communication

Wireless: Wi-Fi, Bluetooth

Better Sensor Networks

- 40% of energy costs in an office building is lighting
- U.S. movie theaters
  - Some have energy costs >$400 per day
  - Can vary by a factor of 10
- Sensor market in 2001 was ~$11 Billion*
  - Wiring installation costs > $100 Billion
- Wireless sensor market in 2010 of $7 Billion†
- 1.5 Billion transducer devices installed by 2010‡

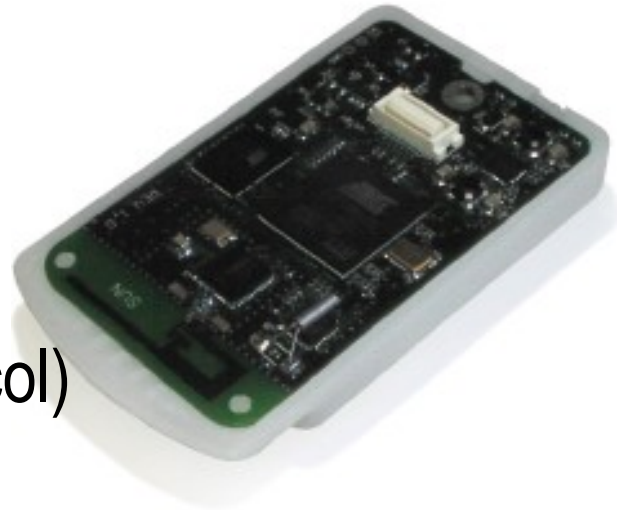
* Freedonia Group report on sensors, 2002 † ON World Report ‡Harbour research report

Sun SPOT Hardware

- Platform for Java wireless sensor networks
- Uses off-the-shelf components
- Small, but not tiny
- Modular architecture
  - > Stackable boards
  - > Hirose connectors
- Up to three boards can be stacked
  - > Plus power supply

SPOT Processor Board

- 180MHz 32-bit ARM 920T CPU
  - 512Kb RAM, 4Mb FLASH
- Chipcon 2420 radio package
  - 2.4GHz frequency
  - IEEE 802.15.4 (Low rate PAN protocol)
- USB interface— mini-b connector
- 3.6V rechargeable 750 mAh Li-Ion battery
- Power consumption 40-100mA
  - Depending on radio/LED/peripheral usage
  - 36 µA deep sleep mode

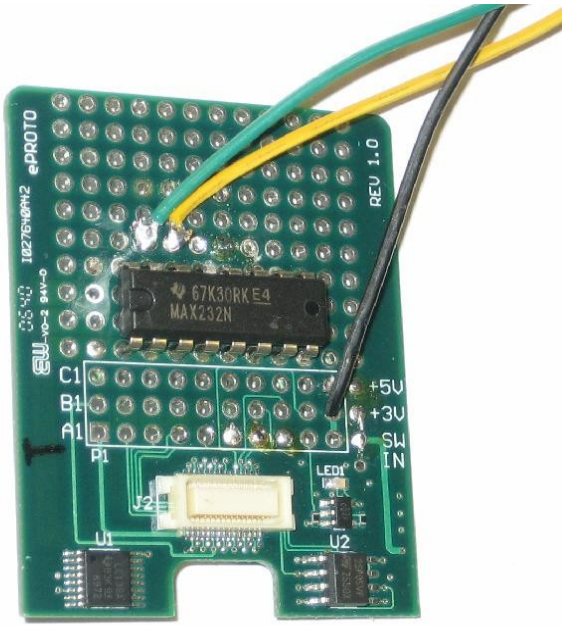


SPOT Demo Sensor Board

- 2G/6G 3-axis Accelerometer
- Temperature sensor
- Light sensor
- 8 RGB 24bit LEDs
- 2 push-button switches
- 6 analogue inputs
- 5 general purpose I/O pins
- 4 High current (100mA) output pins
  - > Easy to interface to servos, speakers and other devices

SPOT Prototype Board

- Direct access to processor board signals
- Serial access via UART and then MAX chip to drive RS-232
- SD Card reader for persistent storage
- Others



SPOT Hardware Open Sourced

- spots-hardware.dev.java.net
- eSpot Processor board
- eDemo sensor board
- eProto/eProtomega board
- eSerial RS-232 interface board
- eFlash SD-Card reader board
- eBones – design guidelines for new boards
- eUSB SPOT as a USB hub (not complete)

Squawk Virtual Machine

- Objective: very portable, small footprint JVM
  - No underlying OS
  - Runs on “bare metal”
- Most of code written in Java
  - Interpreter and low level I/O code written in C
  - Everything else in Java
- Provides Java ME CLDC 1.1 environment
  - Additional libraries for specific functions such as sensors, LEDs, etc
- Isolates

The Sun SPOT SDK: Libraries

- Squawk Java VM: desktop and Sun SPOT
- Libraries
  - Java ME CLDC 1.1 libraries
  - Hardware libraries
    - Demo sensor board library
  - Radio libraries
  - Network libraries
    - 802.15.4 MAC layer written in the Java programming language, uses GCF
  - Desktop libraries (Basestation)

Accelerometer

- SPOT has built in 3-axis accelerometer
  - Uses ST-Micro LIS3L02 component
- Scale can be set to 2G or 6G
  - Good sensitivity
- Acceleration is measured relative to gravity
  - Tilting the SPOT changes the value

```
EDemoBoard db = EDemoBoard.getInstance();
IAccelerometer3D acc = db.getAccelerometer();
acc.setRange(0);  // 2G
IScalarInput xAccel = acc.getX();
int xa = xAccel.getValue();
```

Light/Temperature Sensor

```
EDemoBoard db = EDemoBoard.getInstance();
ILightSensor light = db.getLightSensor();
ITemperatureInput temp =
  db.getTemperatureInput();

int brightness = light.getValue();
int heat = temp.getValue();
```

Switches and LEDs

```
EDemoBoard db = EDemoBoard.getInstance();
ISwitch sw1 = db.bindSwitch(EDemoBoard.SW1);
ISwitch sw2 = db.bindSwitch(EDemoBoard.SW2);
ITriColorLED[] leds = db.getLEDs();

leds[0].setRGB(0, 255, 0);
leds[7].setRGB(0, 0, 255);

while (true) {
  leds[0].setOff();

  if (sw1.isClosed())
    leds[0].setOn();
}
```

Radio Positioning

- SPOT sends 'ping' radio signal
- APIs provide simple radio signal strength access
  - `Radiogram.getRssi()`
- Take signal strength from multiple basestations
  - Inverse square law for distance
  - Triangulate position
  - More basestations means more degrees of accuracy
- Stability of signal strength is not high
  - Resolution of position changes is therefore low
  - Good enough for some situations

Isolates

- SPOT implementation based on JSR-121
- Multiple Java applications on a single VM
  - Isolated from each other
  - Start and stop independently
  - Exception in one isolate does not affect others
- Used for over-the-air application deployment
- Debugging
- Can migrate running isolate to different VM
  - Serialise isolate
- Great for truly distributed applications

Sun SPOT Developer's Kit

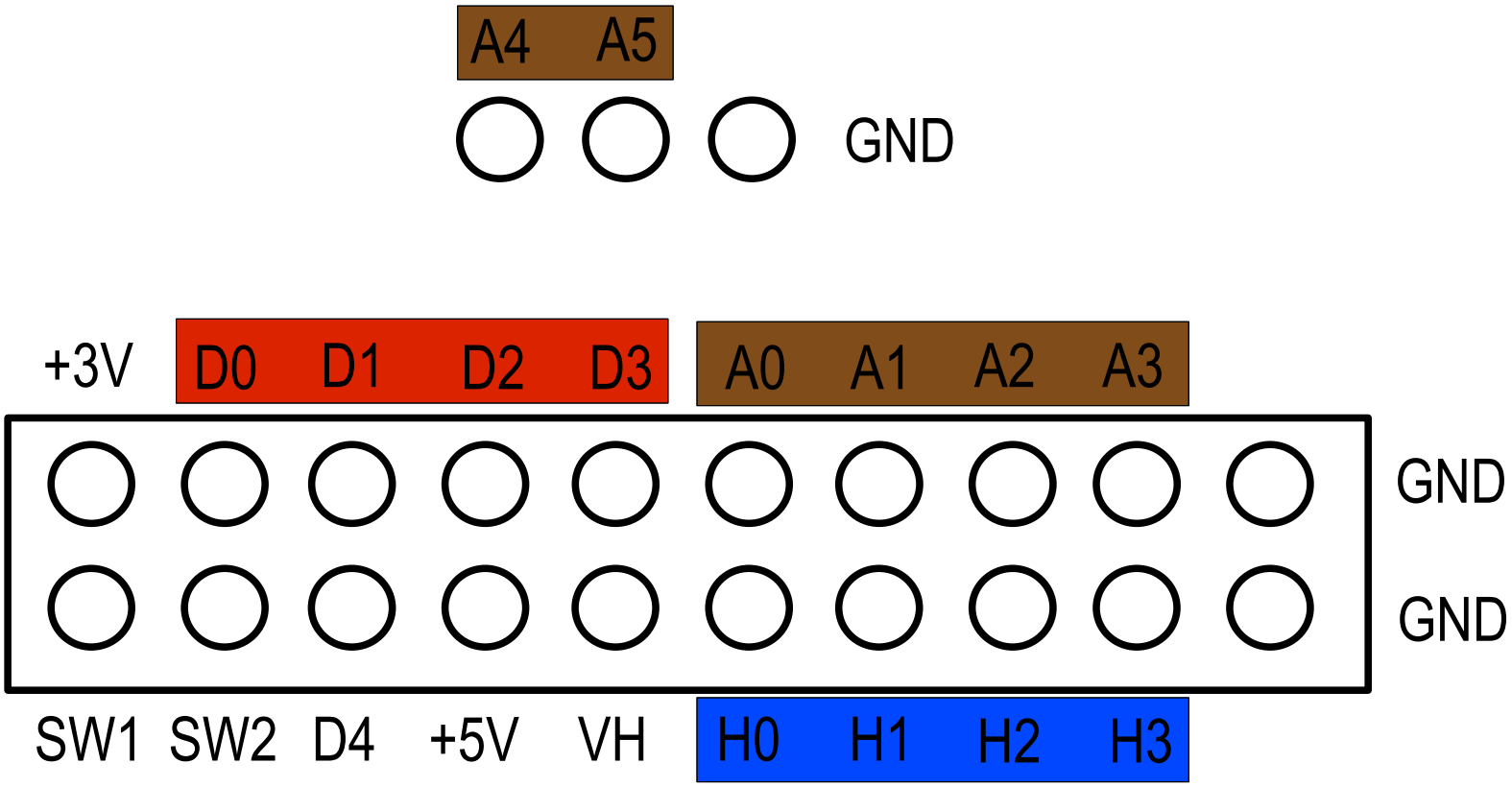
- Two full Sun SPOTs with eDemoSensor boards and batteries
- One base-station Sun SPOT
- Software
  - Squawk VM
  - Java SDK
  - Netbeans
- USB cable
- Mounting clips

Sun's Focus

- Strengths
  - Operating Environment—*Squawk VM/Java*
  - Development/Deployment Tools—*Net Beans, SPOTWorld*
  - Security/DRM—*Sizzle, OpenMediaCommons.org*
  - Scalability/Back-end support
- Other Differentiators
  - Platform for experimentation/inspiration—*don't optimize prematurely, design for flexibility*
  - Significant Local Processing—*32-bit processing*
  - Encourage Actuation/Control—*robotics, toys, etc.*
    - Very easy to control servos etc.

SPOT Interfacing

External Connections



Digital Lines

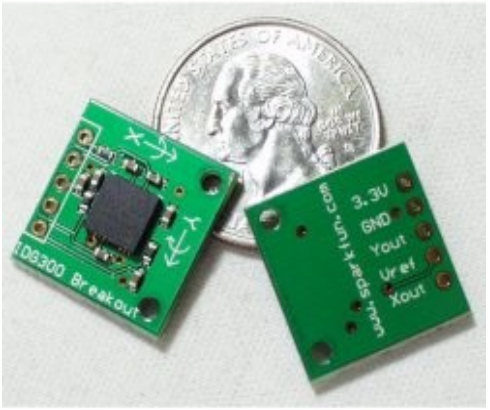
- Can be used for either input or output
- Input
  - isHigh()/isLow()
  - getPulse()
- Output (can be used to drive servo motors)
  - IServoController interface
  - bindServo()

Analog To Digital Converters

- SPOT has 6 ADC lines accessible via external header pins
- Apply input that is in range 0-3V
- Read value with 10-bit resolution via **`IScalarInput`** class

```
EDemoBoard db = EDemoBoard.getInstance();
IScalarInput analog =
    db.bindScalarInput(EDemoBoard.A0);
int analogValue = analog.getValue();
```

Solid State Gyroscope



- IDG-300 dual-axis gyroscope
- 3V supply can be taken from SPOT
- X and Y lines connect to ADC pins on SPOT
- Use 2 mounted orthogonally for full 3D data
- Provides rotational velocity
  - > Can be used to calculate change in orientation of SPOT
  - > Change is 2mV/degree/second
  - > Some drift creeps in – needs to be accounted for

The P5 Data Glove

- Designed for gaming applications
- Uses proprietary hardware and software
- Required modification to work with SPOT
  - Very fiddly soldering to surface mounted connector
- Two gyros glued inside
- SPOT mounted on top using velcro

Game Pad Thumb Joystick

- Remove from cheap game pad
- Left-right and back-forward wired to ADC lines
  - Implemented as potentiometers
- Push-button wired to digital input
  - Switch pulls pin from 0 to 3V

Compass Sensor

- Useful for determining orientation
  - Z-axis of accelerometer not really suitable
- CMPS03 magnetic compass module
  - Uses 2 Philips KMZ10A sensors
  - 0.1 degree resolution, 3-4 degrees accuracy
- Separate head-mounted SPOT
- I2C or PWM data connection
  - SPOT can do I2C via “bit-banging”
  - SPOT API has **`getPulse()`** method

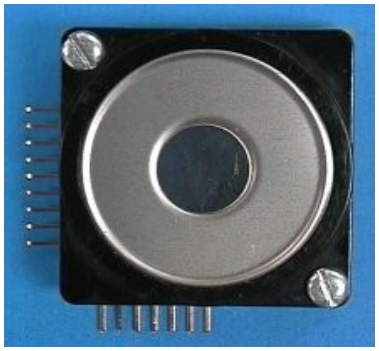
Servo Motors

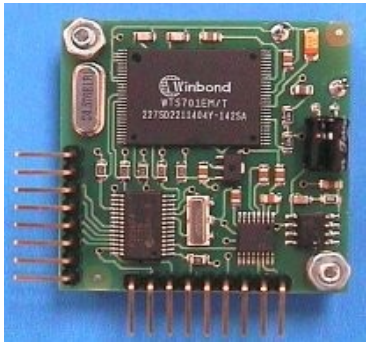
- Simple interface
- Digital lines (D0-D4) or high current lines (H0-H3)
  - Using H lines requires V+ from servo power supply
- Instantiate new **`Servo`** object with chosen pin
- Set bounds
- Set servo to desired position
- Can be made continuous rotation
  - Minor hardware modification to servo

RoboSapien RS-Media Robot

- Runs embedded Linux
  - PhoneME open source Java ported to robot
- Serial console port
  - Accessible internally
- Connect SPOT to serial port for bi-directional comms
  - Tricky
- Wireless control!

Feedback





- Talking SPOT
- RS-232 interface SP03 board
  - 30 pre-recorded phrases
  - Text to speech capable
- SPOT can drive pins as UART
- Use MAX3232 as line driver
  - Convert TTL voltages to RS-232

For More Information

- Squawk (Now open sourced)
  - http://squawk.dev.java.net
  - http://research.sun.com/projects/squawk
- Sun SPOT (also open sourced)
  - http://www.sunspotworld.com
- Papers
  - “Java™ on the Bare Metal of Wireless Sensor Devices—The Squawk Java Virtual Machine”, VEE, June 2006
  - “The Squawk Virtual Machine: Java™ on the Bare Metal”, Extended Abstract, OOPSLA, Oct 2005

Demos