

# Interfacing Java-DSP with Sensor Motes

*by*

*H. M. Kwon, V. Berisha and A. Spanias*

*Ira A. Fulton School of Engineering, Department of Electrical Engineering, MIDL Lab*

*Arizona State University, Tempe, AZ 85287-5706, USA*

*[http://jdsp.asu.edu/JDSP\\_sensors/index.html](http://jdsp.asu.edu/JDSP_sensors/index.html)*

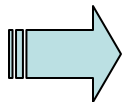


**Sensor, Signal and Information Processing Workshop**  
**April 28<sup>th</sup>, 2005**



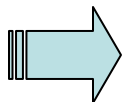
# Overview

J-DSP



- ◆ A Web-based DSP Simulation Tool
- ◆ Universally accessible DSP functions
- ◆ Embeds Interactive Simulations in Web pages
- ◆ Seamlessly Integrates Animated Demos

Wireless  
Sensor  
Motes



- Seamless Integration with J-DSP enables real-time sensor signal analysis
- Java interface natural for remote sensing
- User-friendly GUI for computation/graphics using the J-DSP-Mote interface
- Hardware: *Mica2* from *Crossbow*



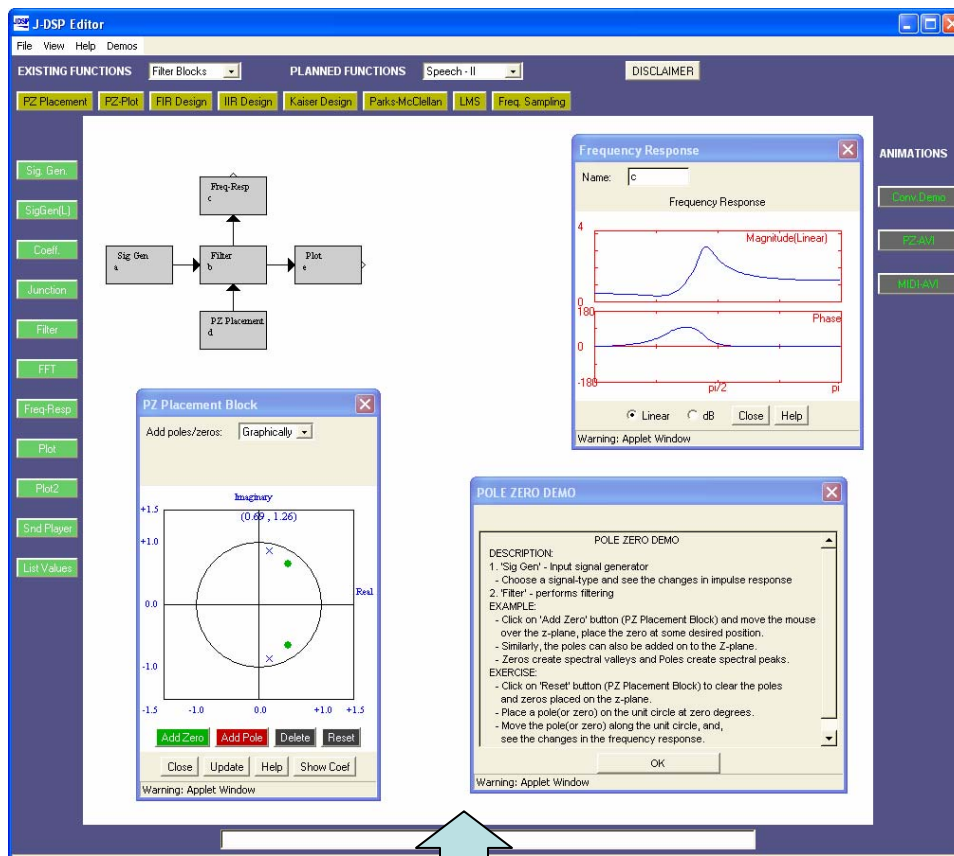
# Motivation

- ◆ Wireless sensor networks have gained popularity in a number of applications
- ◆ Simplify control of *Mica2* platform through the object-oriented, platform independent structure of Java-DSP
- ◆ Connectivity with the signal processing environment of Java-DSP allows for real-time sensor data analysis
- ◆ Remote sensing possibilities
- ◆ Control by Java based handheld devices (i.e. PDAs)



## BASIC FUNCTIONALITY IN J-DSP

- ◆ Fundamental DSP functions (FFT, IFFT, Windowing, etc.)
- ◆ Arithmetic Functionality
- ◆ Digital Filtering
- ◆ FIR/IIR Filter Design
- ◆ Spectral Estimation
- ◆ Multi-rate DSP
- ◆ Visualization Blocks
- ◆ Pole-Zero Demo
- ◆ Frequency Response
- ◆ Sensor Networks



The screenshot shows the J-DSP Editor interface with several key components:

- Main Window:** Displays a block diagram with blocks labeled 'Sig Gen' (a), 'Filter' (b), 'PZ Placement' (d), 'Freq Resp' (c), and 'Plot' (e).
- Frequency Response Window:** Shows two plots: 'Magnitude(Linear)' and 'Phase'. The magnitude plot shows a peak, and the phase plot shows a corresponding phase shift.
- PZ Placement Block Window:** Features a complex plane plot with a unit circle. A zero is marked with a green dot at (0.89, 1.26) and a pole with a red 'x' at (0.89, -1.26). Buttons for 'Add Zero', 'Add Pole', 'Delete', and 'Reset' are visible.
- POLE ZERO DEMO Window:** Contains a 'DESCRIPTION' section with instructions on how to use the PZ Placement Block and a 'POLE ZERO DEMO' section with an 'EXERCISE' section.

At the bottom of the screenshot, two physical DSP boards are shown with a double-headed arrow between them, indicating hardware connectivity.



# Hardware Platform

Temperature: Panasonic ERT-J1VR103J

Microphone / Tone Detector

Sounder: Ario (centered at 4.5 kHz)

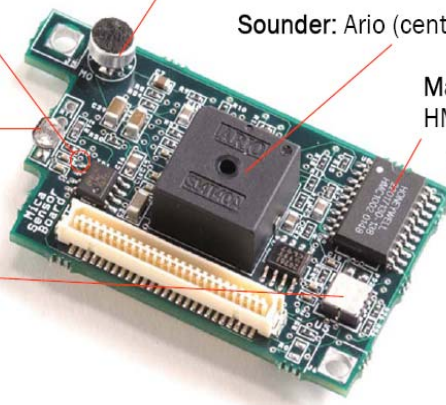
Light (Photoresistor):  
Clairex CL94L

Magnetometer: Honeywell  
HMC1002 (MTS310CA only)

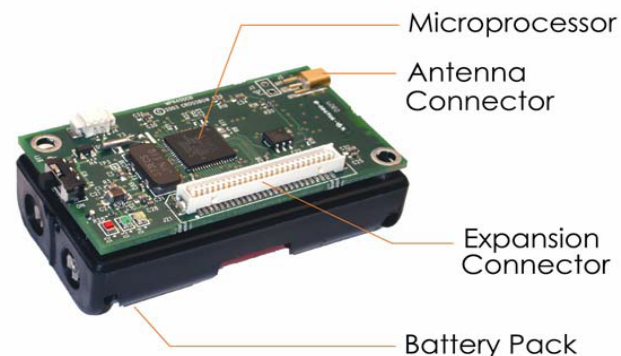
- Resolution: 134 mGauss

Accelerometer: ADI ADXL202  
(MTS310CA only)

- 2 axis
- Resolution:  $\pm 2\text{mG}$



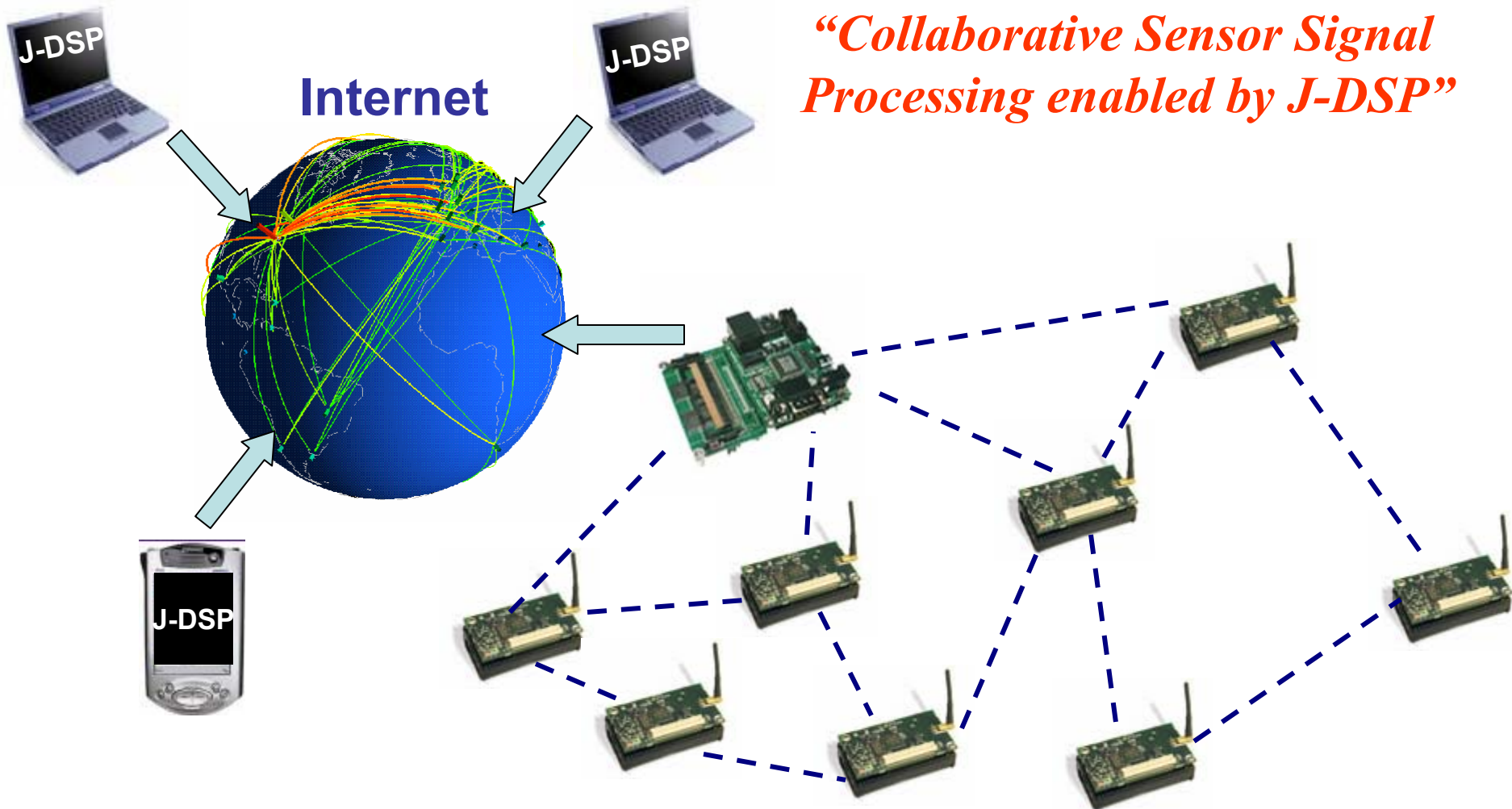
**MIB(510) Gateway**  
: serial port programmer



- **Targeted Applications:** Environmental Monitoring, Security, Source Localization, Tracking, Biological Applications



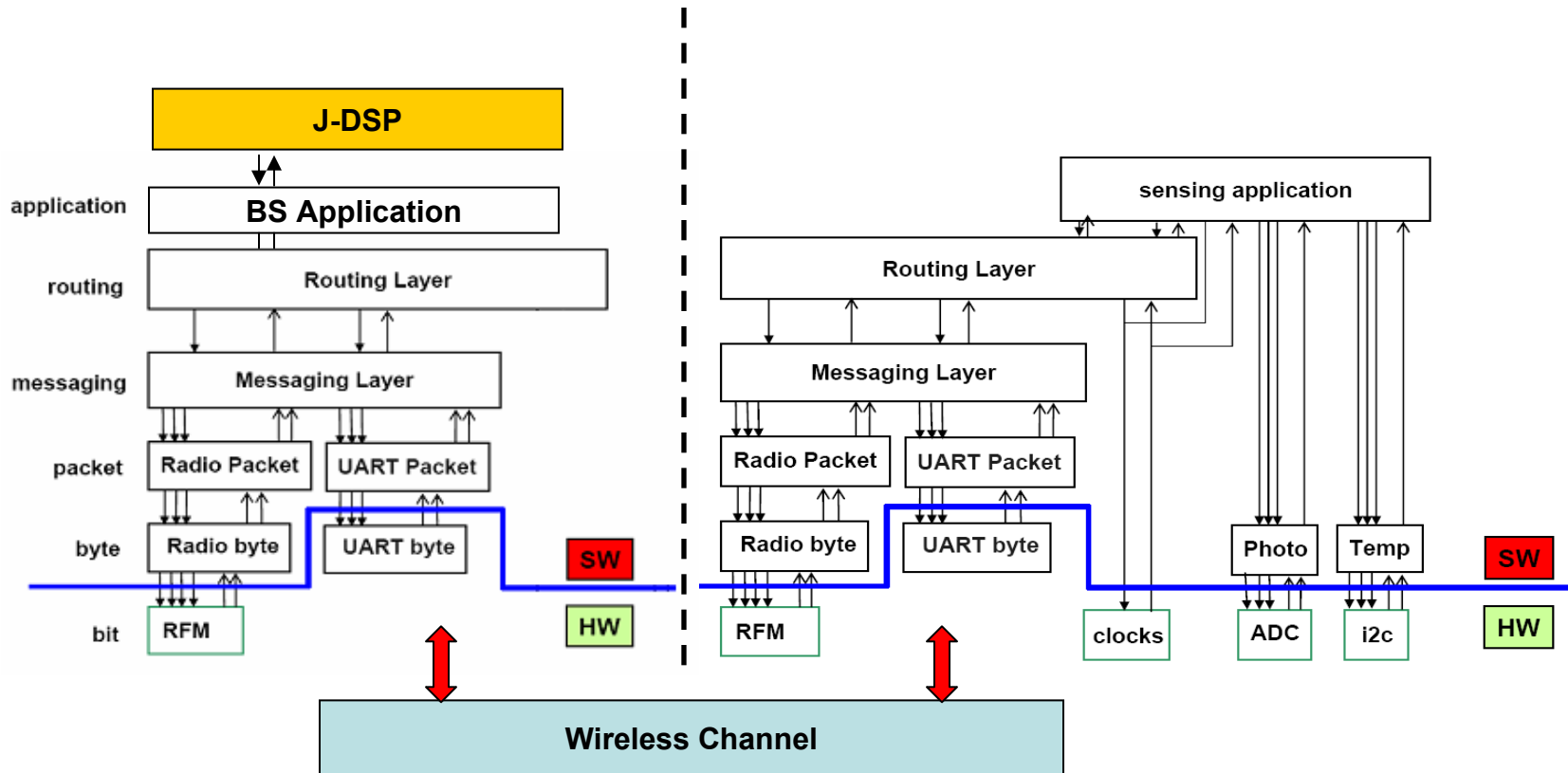
# Java-DSP and the Motes



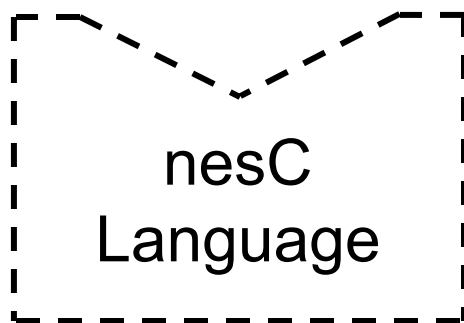
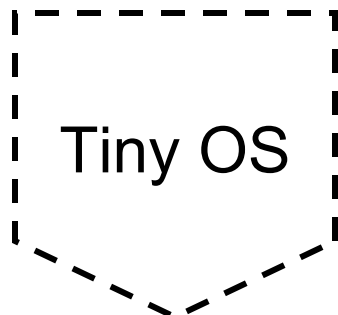


# Tiny OS and Java-DSP

- ◆ Java-DSP acts as an additional layer at the base station
- ◆ Lower layer processing is seamless to the user



# Tiny OS & nesC



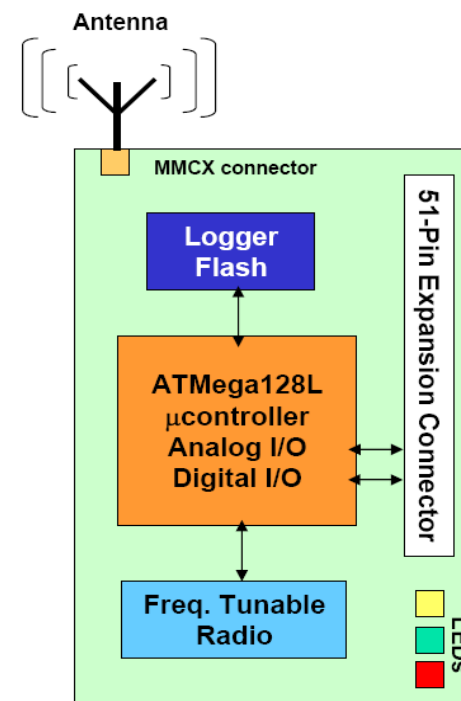
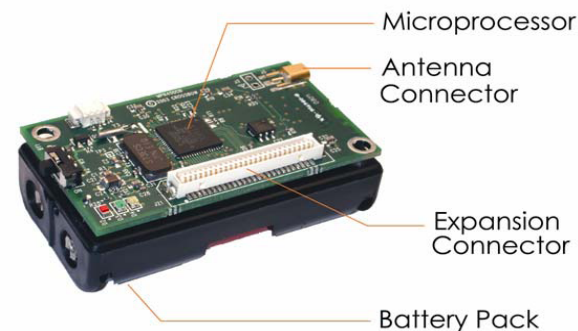
- Simple and powerful OS for low power
  - Re-use of component
  - “Hurry up and sleep”
  - Scheduling based on events and tasks
  - FIFO structure
- 
- TinyOS syntax and structure
  - Dialect of C language
  - A pre-processor
    - Converts wiring of high level modules into efficient code
    - nesC output is a c program file that is compiled and linked using gnu-gcc tools for a specific Mote





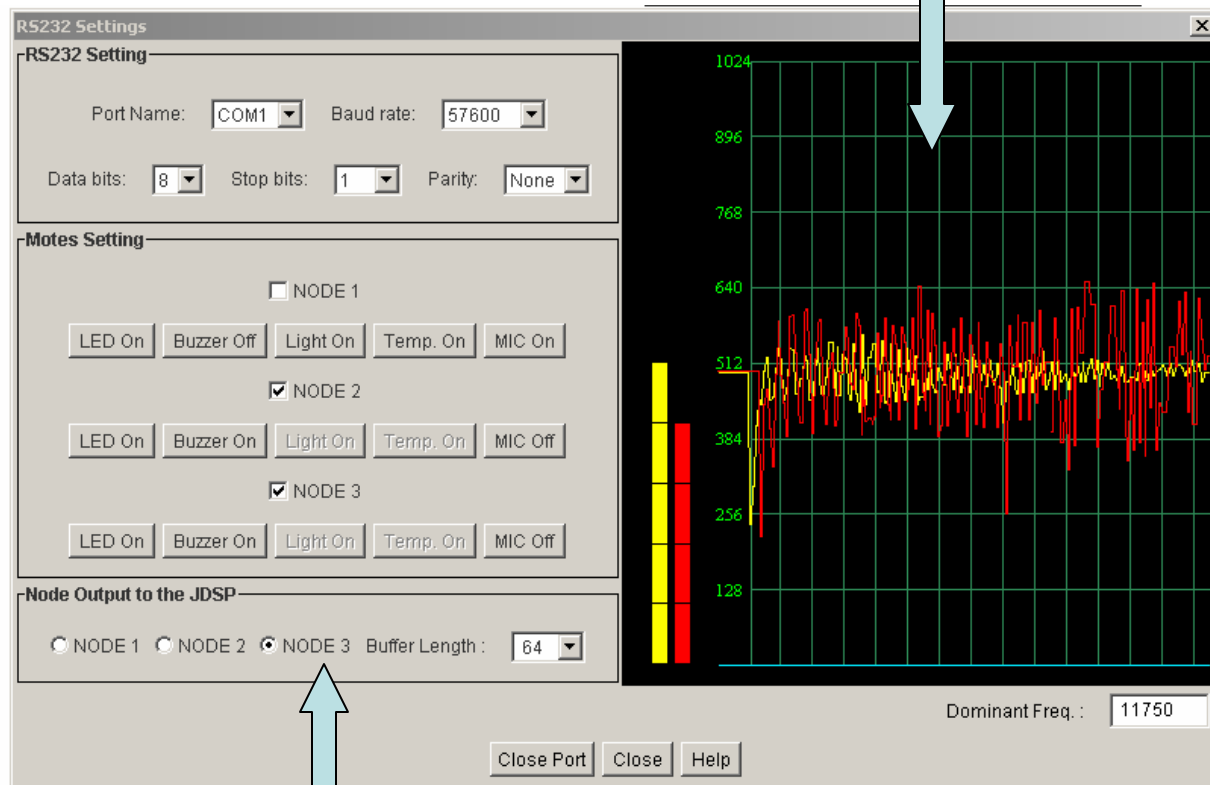
# The Motes (MICA2 Platform)

- **Microprocessor: Atmel ATmega 128L**
  - 7.3728 MHz clock
  - 128 kB of Flash for program memory
  - 4 kB of SRAM for data and variables
  - 2 UARTs
  - Serial Port Interface (SPI) bus
  - Inter IC (I2C) bus
- **Radio: Chipcon's CC1000**
- **External serial flash memory: 512 kB**
- **51-pin expansion connector**
  - Eight 10-bit analog I/O
  - 21 general purpose digital I/O
- **User interface: 3 LEDs**
- **JTAG port**
- **Powered by two AA batteries**
  - 1850 mAh capacity



# The MOTE Block

- ◆ GUI for the motes
- ◆ Control panel is used to control the individual motes and the RS232 settings
- ◆ MOTE block in J-DSP allows users to control individual motes
- ◆ Real-time graph plots data as it comes

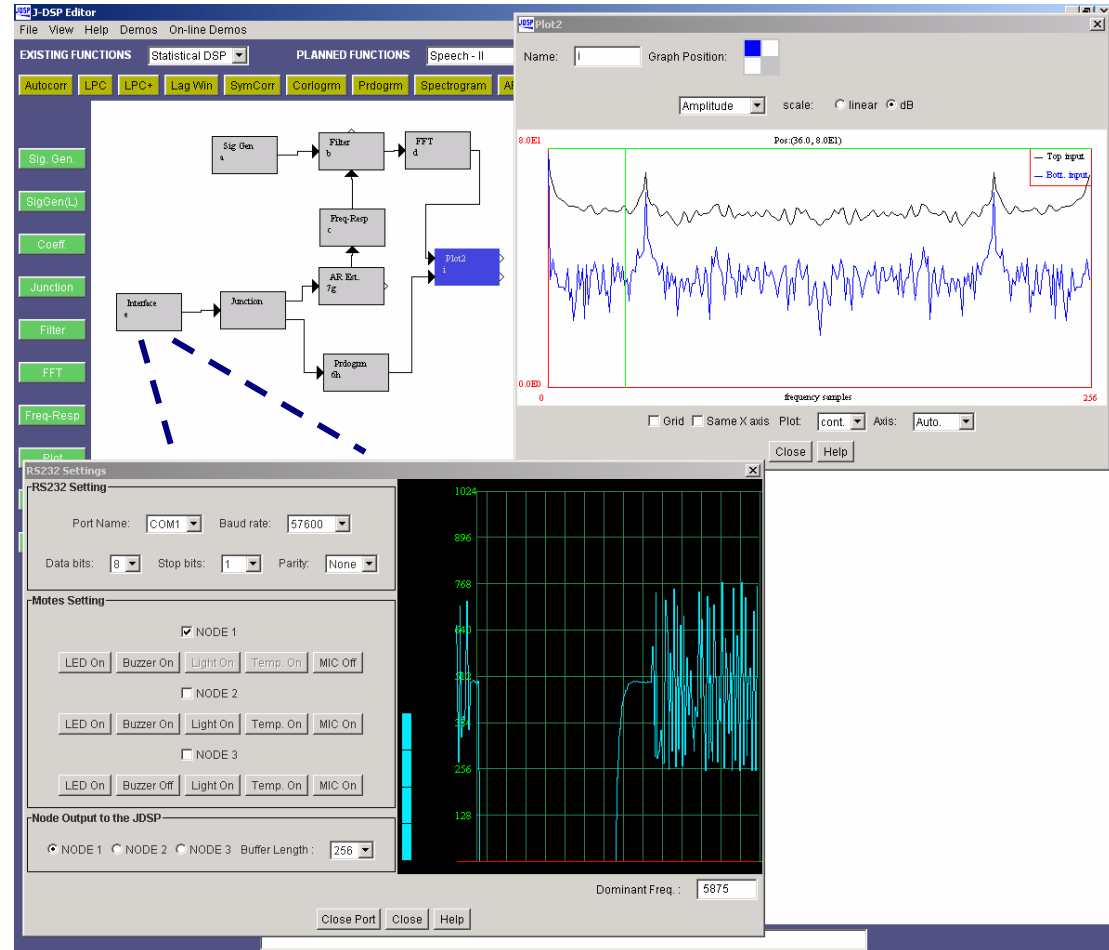


Control Panel

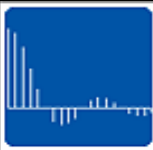


# Sensor Network Signal Processing with J-DSP

- ◆ A number of advanced signal processing features available in J-DSP
- ◆ You can connect the incoming data to existing blocks to create DSP systems
- ◆ Example: Fitting incoming data to an auto-regressive model



# Remote Sensing with J-DSP



- ◆ Preliminary example shows possibilities for sensing and security applications
- ◆ Display panel shows which sensors are active
- ◆ Active Sensors:
  - ◆ Light
  - ◆ Sound
  - ◆ Temperature
  - ◆ Accelerometer

Security Demo

Port Setting

Port Name: COM1

```

Node ID: 21
BAT: 398 TEP: 515 LIT: 779
MIC: 493 ACX: 520 ACY: 470
Node ID: 24
BAT: 434 TEP: 515 LIT: 512
MIC: 501 ACX: 494 ACY: 511
Node ID: 21
BAT: 399 TEP: 515 LIT: 784
MIC: 489 ACX: 521 ACY: 470
Node ID: 22
BAT: 400 TEP: 498 LIT: 937
MIC: 488 ACX: 481 ACY: 501
Node ID: 21
BAT: 398 TEP: 515 LIT: 778
MIC: 478 ACX: 520 ACY: 470
Node ID: 22
BAT: 400 TEP: 498 LIT: 936
MIC: 537 ACX: 481 ACY: 500
Node ID: 24
BAT: 434 TEP: 515 LIT: 523
MIC: 525 ACX: 495 ACY: 512
Node ID: 23
BAT: 400 TEP: 504 LIT: 944
MIC: 453 ACX: 506 ACY: 487
Node ID: 21
BAT: 398 TEP: 515 LIT: 778
MIC: 521 ACX: 521 ACY: 470
Node ID: 22
BAT: 400 TEP: 498 LIT: 937
MIC: 426 ACX: 481 ACY: 501
Node ID: 21
BAT: 398 TEP: 515 LIT: 777
MIC: 453 ACX: 520 ACY: 470
Node ID: 24
BAT: 433 TEP: 515 LIT: 523
MIC: 480 ACX: 494 ACY: 512
Node ID: 22
BAT: 400 TEP: 499 LIT: 937
MIC: 587 ACX: 481 ACY: 500
Node ID: 21
BAT: 398 TEP: 515 LIT: 788
MIC: 479 ACX: 520 ACY: 470
Node ID: 24
BAT: 433 TEP: 515 LIT: 525
MIC: 484 ACX: 494 ACY: 512
Node ID: 21
BAT: 398 TEP: 515 LIT: 779
MIC: 466 ACX: 520 ACY: 470
    
```

AVERAGE

Temperature : 508

Light : 796

MIC. : 497

Accel\_X : 500

Accel\_Y : 492

Close Port Close Help



## ◆ J-DSP and Motes for Research

- ◆ Source localization using the Motes
- ◆ Target tracking
- ◆ Interfacing with advanced J-DSP features (i.e. HMM)
- ◆ Collaborative remote sensing using J-DSP
- ◆ Implement sensor networks using J-DSP/Motes for smart home and security applications

## ◆ J-DSP and Motes for Education

- ◆ Train UG and grad. students the basics of working with wireless Motes using the J-DSP GUI
- ◆ Train engineers and practitioners in real-time analysis of sensor data
- ◆ Use hands-on hardware/software approach to create a workforce trained in using sensors for security and other applications



# Summary



- ◆ Simulation modules and blocks in J-DSP have been developed to control the *Crossbow* Motes
- ◆ Object-oriented structure of J-DSP allows for easy manipulation of the Motes
- ◆ Please visit <http://jdsp.asu.edu> for more information on J-DSP
- ◆ J-DSP also supports: Statistical DSP simulations, Communications, Speech analysis-synthesis, 2D and Image processing, Spectrogram/time-frequency experiments, and Controls simulations

*Some figures taken from <http://www.xbow.com>*

