IBM SYMPOSIUM
7. OKTOBER 2010

Congress Center Wien
1020 Wien, Messeplatz 1
IBM Mote Runner: Drahtlose Sensornetze für Smarter Planet

Dr. Thorsten Kramp
IBM Zurich Research Laboratory
Wien, im Oktober 2010
A wireless sensor network (WSN) is a wireless network of spatially distributed autonomous devices using sensors and actuators to cooperatively monitor and react to physical or environmental conditions.

- Heterogeneous set of autonomous devices (motes).
  - Processors, radio controllers, sensors / actuators
- Potentially very limited resources.
  - Computing power, memory, energy
- Application development, network management, and data visualization is very difficult.
  - Even for computer scientists
WSN Application Scenarios

• Agronomy management
  – Measure various soil parameters
e.g., moisture, salinity, temperature
  – Optimize the ability to grow, flourish, proliferate
  control parameter to stay in “the zone”
  – Avoid diseases, inconsistency, waste of water / fertilizer
  identify problems early, not when the plant declines
  – Examples: green houses, vineyards, agriculture, golf courses

• Environmental monitoring
  – Environmental research
e.g., long-term biocomplexity mapping and habitat sensing
  – Early warning systems
e.g., pollution, floods, fires, landslides, earthquakes
WSN Application Scenarios

- **Building and facility management**
  - Convenience, safety, and security  
    e.g., lighting, air handlers, fire warning, surveillance  
  - Smart metering and energy monitoring  
    e.g., min-energy buildings, remote access for utility companies

- **Industrial applications**
  - Safety and regulatory compliance  
    e.g., emission control, collateral effects monitoring  
  - Logistics  
    e.g., goods tracking, traffic data / road conditions

- **Health care and sports medicine**
  - Examples: elderly people at home, training optimization
IBM Mote Runner: What is it?
An open, dynamic run-time platform and development environment for WSNs

- **Operating System**
  - 8/16 bit CPUs (also runs on 32 bit CPUs)
  - minimum requirements: 8 KB RAM, 64 KB Flash
  - power management, device access, scheduling

- **Virtual Machine**
  - portable binary applications
  - managed memory (garbage collection)
  - controlled access to all objects

- **Programming Platform & Management Framework**
  - support of IDEs for development (Eclipse, ...)
  - state-of-the-art tool chain (Java compiler, converter, optimizer, debugger, simulation)
  - manage applications on a network of motes (load, delete, update)
And what not?

• Hardware Development
  – use of existing SOCs and mote hardware (e.g., Crossbow Iris)
  – no development of sensors and actuators

• Network Protocols
  – no design of new network protocols
  – use of existing standards (e.g., ZigBee, 6LowPan, WirelessHart)

• Application Development
  – only to demo MoteRunner
  – occasional application to attract domain specialists
Key Characteristics

• **Efficient:** Make “best” use of the available resources (especially power).

• **Scalable:** Can be deployed on a wide range of motes (especially small ones).

• **Portable:** Virtual machine shields applications, generates “virtual homogeneity.”

• **Dynamic:** Can be dynamically (re-)configured in the field without physical access.

• **Tool Chain:** Multi-language (Java, …), compiler, optimizer, debugger, simulation.

• **Remote Management:** Multi-protocol edge server provides web-based management plus integration with different management backends.

• **Accessible:** Can be programmed in a state-of-the-art way by computer scientists; can be deployed and used by domain specialists.
A Bird’s Eye View

physical network

simulated network

backend applications

edge server

web applications

development environment
Simulation Environment

• **Mote Hardware**
  – memory image, power usage, CPU performance, sensor feeds

• **Networks**
  – network of motes run in a single process
    (individual radio messages with timing, location and signal quality).

• **Comfortable Testing and Inspection**
  – source-level debugging of applications;
  – logging and tracing of system events;
  – simulation can stop / step network.
IBM Mote Runner: Summary

• Runs efficiently on rather limited resources.
  8 bit processor, 8 KB RAM, 64 KB Flash

• Supports broad range of standard hardware.
  configurable for standard motes hardware with varying resource configurations

• Application portability by means of virtual machine.
  shields application developers from hardware peculiarities

• Allows dynamic reconfiguration after deployment.
  applications can be loaded and removed at any point in time

• Comfortable to program and manage.
  object-oriented programming languages (Java, C#), state-of-the-art tools (source-level debugging), simulation environment, web-based configuration / management / visualization
Available now @ alphaworks.
http://www.alphaworks.ibm.com/tech/moterunner

Free for academic use, 90 days commercial evaluation.
Your feedback is important to us: moterunner@zurich.ibm.com

Ready for selected customer projects.
Get in touch with us: moterunner@zurich.ibm.com