Integrated Graphical Development and Simulation Environment for TinyOS-based Wireless Sensor Networks

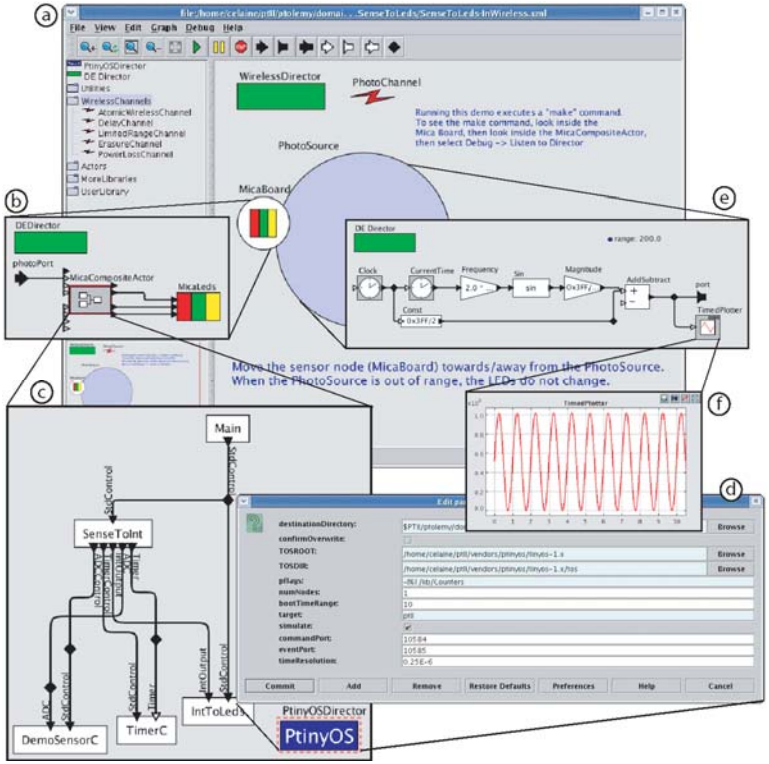
http://ptolemy.eecs.berkeley.edu/viptos

The Ptolemy Project at UC Berkeley is pleased to announce the **Viptos 1.0.beta release**. Viptos (Visual Ptolemy and TinyOS) is an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks.

Highlighted new features in Viptos 1.0.beta:

- Compatible with both Linux and Cygwin.
- Includes multi-hop routing demo using the TinyOS Surge routing protocol application demo.
- Based on the latest release of Ptolemy II (version 6.0).
- Improved user interface.

Note that we are using a new version numbering system to more accurately reflect the development level of Viptos.



Viptos allows networked embedded systems developers to construct block and arrow diagrams to create TinyOS programs from any standard library of TinyOS components written in nesC, a C-based programming language. Viptos automatically transforms the diagram into a nesC program that can be compiled and downloaded from within the graphical environment onto any TinyOS-supported target platform.

Full abstract:

Viptos (Visual Ptolemy and TinyOS) is an integrated graphical development and simulation environment for TinyOS-based wireless sensor networks. TinyOS is a component-based, event-driven runtime environment designed for wireless sensor networks.

Viptos is built on Ptolemy II, a modeling and simulation environment for embedded systems, and TOSSIM, an interrupt-level discrete event simulator for homogeneous TinyOS networks. In particular, Viptos includes the full capabilities of VisualSense, a Ptolemy II environment that can model communication channels, networks, and non-TinyOS nodes. Viptos extends the capabilities of TOSSIM to allow simulation of heterogeneous networks.

Viptos provides a bridge between VisualSense and TOSSIM by providing interrupt-level simulation of actual TinyOS programs, with packet-level simulation of the network, while allowing the developer to use other models of computation available in Ptolemy II for modeling the physical environment and other parts of the system. This framework allows application developers to easily transition between high-level simulation of algorithms to low-level implementation and simulation.

Contact information:

Elaine Cheong (celaine@eecs.berkeley.edu), Ptolemy Project, University of California, Berkeley