Power Aware Wireless Sensor Networks

Simulation Framework

Introduction

The PAWiS Project at the Institute of Computer Technology, University of Technology, Vienna, targets on optimized wireless sensor networks (WSN). These require nodes with exceptionally low power consumption, yet have to drive sensors and radio communication.

Common optimization approaches concentrate on modules to make every module as good as possible. Unfortunately this reveals only local optima. To find the best possible solution within certain constraints, the global optimum must be sought. The PAWiS Framework targets on cross layer and cross module optimization.

Methodology

A new development methodology is introduced with the PAWiS Framework. The network and the sensor nodes themselves are simulated and the results evaluated. The nodes are modeled as virtual prototypes, i.e., their

- function,
- timing,
- power consumption and
- failures

are simulated. The design is conducted as true top-down development. Optimization starts at system level and proceeds via cross-layer optimization down to precise module optimization. This ensures the steady approach to the global optimum.

The simulation model reflects the system architecture and is built of modules. Each of them is defined by its type (e.g. MAC Layer, UART, ...). For every type a library of various implementations and varying detailedness is prepared.

The work flow is primarily determined by the refinement cycles and outcome evaluation. The nodes are composed of a set of modules from the library, configured and parametrized. Then the whole network is simulated and the output is evaluated. The acquired information is used to refine the module implementations and adapt the composition and configuration. Then the cycle starts over again. This refinement cycles enhance the development of your WSN system.

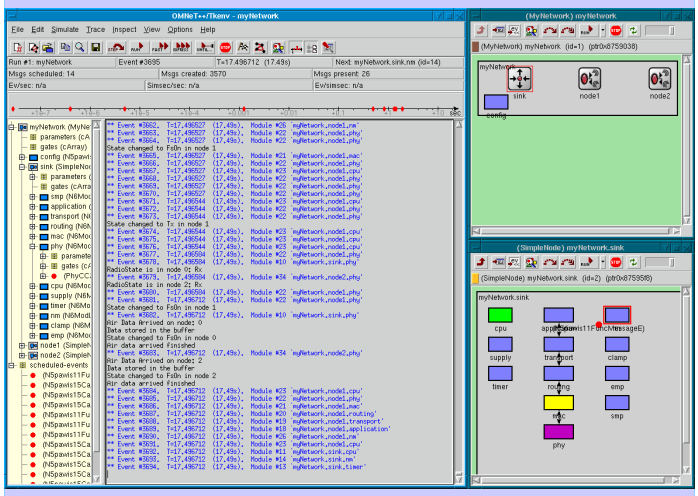
The outcome features the fully functional description, architecture specification, implementation details as well as power consumption figures.

Optimization

For optimization numerous options are utilized to improve functionality and power consumption. The actual module implementation for a certain module type is selected (e.g., the type of an ADC) as well as the scaling of the module (e.g., the resolution of the ADC). This is supplemented by device partitioning between hardware and software, between analog and digital as well as between RF and baseband part. Further more every module can be parameterized (e.g., timing). This large quantity on variations within the design space offers great flexibility and opportunity for optimization.

Simulation Framework

The PAWiS Simulation Framework consist of a library of C++ classes and templates. Every module of the WSN node is implemented as C++ class derived from the PAWiS base classes. The modules are combined to the node with a configuration file. Another configuration file specifies the network layout. The simulator is built by compiling and linking the C++ files. It features a graphical user interface with step by step control of the module interaction. Alternatively a console application for high simulation speed can be built.



Every module is implemented as one ore more state machines interacting with each other. Interaction between modules is implemented as so called Functional Interfaces.

Exactly as in real implementations the modules

can be implemented as software modules (running on a CPU) or as dedicated hardware. Software modules actually delegate their processing time and power consumption to the central CPU module. The simplest implementation of the CPU just reports its power consumption and delays the simulation for the processing time. The actual processing is performed by the module itself. A more sophisticated implementation can interface to a CPU simulator and directly execute the node firmware. The CPU is based on a special class which already prepares features like interrupts and power states.

One of the most important features of the PAWiS Framework is the simulation of power consumption. Therefore every module reports its power consumption to the central power monitor. A power supply management is responsible for supply hierarchy and the power distribution.

The PAWiS Simulation Framework is equipped with a scripting interface for a dynamic environment. This is typically used to setup complex node arrangements, implement mobile nodes and to simulate node failures.

Environment

The Environment holds the three dimensional scenario of all nodes and obstacles. It is responsible for the outer world of all nodes and provides the physical quantities (e.g., temperature, humidity) for all sensors. The environment model for the various units is open to the implementer.

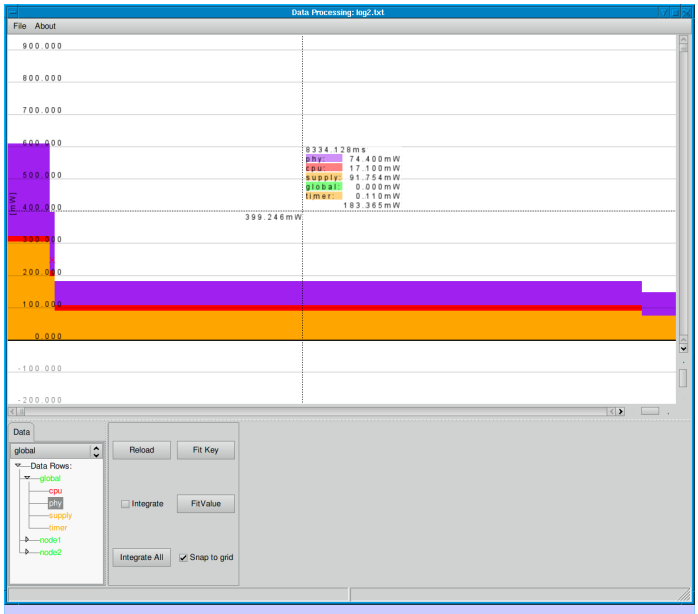
The radio transmission between the nodes is implemented in the so called "Air", which itself is part of the Environment. The transmission is performed for complete packets. It consideres transmission effects and features like

- attenuation due to distance,
- additional attenuation,
- data packet for "wired-OR" addressing schemes (e.g., BitMAC),
- frequency channels (e.g., 83 channels with 1MHz each at the 2.4GHz ISM band) and
- frequency bands (2.4GHz ISM band, GSM 1800 band) and
- code division multiple access (CDMA) scheme.

The receiving nodes are notified of transmissions. From the received signal power the SNR and the BER are calculated which results in packet errors.

Evaluation

Every power report during simulation is stored to a database. The data is visualized by the Data Processing tool. It offers power analysis within



modules and across module boundaries.

Extenstions

The PAWiS Framework is built with extensibility in mind. The following extensions are possible and partly planned.

SystemC: Extend the PAWiS Simulator with your existing SystemC models and perform a PAWiS/SystemC co-simulation.

RF Channel: Simplification for model programmer through optimized RSSI calculation. Continuous interferers, burst interferers and more complex RF interferers.

3D Graphical User Interface: Presentation of and interaction with the environment in a 3D scene.

Availability

The PAWiS Framework is published under the terms of the GNU LGPL (Lesser General Public License). That means that you can download, use and redistribute it free of charge while your source code does not need to be licensed under the LGPL. The Data Processing Tool is released under the Terms of the GNU GPL (General Public License).

Resources

PAWiS Homepage:

http://pawis.sourceforge.net/

PAWiS Wiki:

http://wiki.pawis.sourceforge.net/

System Requirements: The PAWiS Framework is implemented platform independently and supports Unix systems (e.g. Linux) as well as the Windows operating system.

The PAWiS Framework is based on the OMNeT++ 3.3 Discrete Event Simulation System.

Build tools:
Unix: GNU GCC 4.0+; Autoconf 2.60a, Automake 1.9.6
Windows: Microsoft Visual C++ Ver. 7
LUA 5.2, Doxygen 1.4.7
Recommended system parameters: 100 MB free HDD space, 256 MB RAM, 1024x768 screen resolution, CPU at 1200 MHz