

Tutorial Guide: The Sunflower Tool Suite

Hardware Prototypes and Software Research Platforms for Failure-Prone and Resource-Constrained Embedded Systems

Phillip Stanley-Marbell¹, Diana Marculescu²

¹ Technische Universiteit Eindhoven,
Den Dolech 2, Eindhoven 5612 AZ, The Netherlands.

☎ +31 61-478-2010

² Department of ECE, Carnegie Mellon,
5000 Forbes Ave., Pittsburgh, PA 15213-3890, USA.

Abstract. In computing systems research, software tools (notably, simulators) provide low-cost, flexible, and low turn-around time facilities for investigations, but abstract away many hardware details. Hardware implementations on the other hand, provide the ultimate proofs of concept, but require hardware design expertise, are usually expensive and inflexible, and are not always designed to expose all possible system parameters to researchers. They are also rarely the subject of active evolution over time as research platforms in their own right, as software tools are.

The Sunflower tool suite is a suite of hardware platforms and simulation tools, intended to address these concerns. It comprises a full-system (embedded microarchitecture, networking, power, battery, device failure and analog signal modeling) simulator, a miniature energy-scavenging hardware platform, and a handheld computing device. The suite is intended to provide a set of complementary platforms for research in micro- and system-architectures for embedded systems, with emphases on energy-efficiency and fault-tolerance. This tutorial will provide the audience with a working knowledge of the design, implementation and usage of the components of the Sunflower tool suite.

1 Tutorial Objectives

The objectives of the tutorial are to provide a working knowledge of the use of the Sunflower full-system simulator, and Sunflower hardware platforms, to computing systems researchers. The tutorial covers two main topic areas:

- ➊ **Using and extending the Sunflower full-system simulator;** the tutorial will detail the implementation of, and the facilities provided by, the Sunflower framework for performing full-system simulation (microarchitecture, networking, power dissipation and supply, failure-modeling, and more,) of networks of embedded systems.
- ➋ **Using the Sunflower hardware platforms;** the simulation framework is complemented by open hardware platforms, and the tutorial will outline the process of using the hardware platforms in experimental evaluations and research deployments.

1.1 Intended audience

The tutorial is targeted at several potential audiences:

- ◆ **Microarchitecture researchers**, who are looking for a microarchitectural simulator for embedded systems that provides detailed models of the whole system that surrounds a processor or microcontroller. The tutorial will be of particular interest to researchers investigating systems containing multiple (wired or wirelessly) networked processing elements, those investigating the interaction of computation with input signals such as sensors, and researchers investigating the effects of soft-errors.
- ◆ **Sensor network researchers** interested in investigating the computational aspects of their protocols and systems software.
- ◆ **Systems researchers** looking for a platform to enable them to develop compilers and operating systems for embedded systems, that has greater flexibility, transparency and lower cost than hardware, but is also complemented / calibrated against actual available hardware.

1.2 What you should expect to get out of this tutorial

At the end of this three hour tutorial, it is hoped that you would have gained:

- ◆ The ability to use the Sunflower simulation environment to model a single processor or networked system of embedded processors, given an existing system configuration, and to interpret the behavior of the modeled system.
- ◆ The ability to define new system configurations for simulation, and to compile C-language benchmarks for simulation in Sunflower.
- ◆ A knowledge of the available resources for information on the simulation platform's built-in facilities, as well as information on customizing the simulator via runtime or compile-time configuration, or source-code modification.
- ◆ An understanding of the basic architecture of the Sunflower hardware platforms, and their capabilities.
- ◆ The ability to request new features in the periodic revisions of the Sunflower hardware platforms.

2 Background

There exist an abundance of tools for many aspects of computing systems research, from microarchitectural simulators that are the mainstay of computer architecture research [[August et al.](#); [Burger et al., 1996](#)], to networking simulators and other domain-specific tools. Academic research tools are seldom calibrated against specific hardware platforms during their development and evolution, and retrospective comparisons often yield interesting observations [[Gibson et al., 2000](#); [Langendoen, 2006](#)]. Even when the simulation platforms are indeed calibrated against hardware, there is seldom the opportunity

to evolve the hardware platforms in question. This is due both to the expertise required for implementing hardware designs, as well as the cost of fabrication of hardware prototypes. For high-performance computing systems research, the RAMP platform [Arvind et al., 2005] addresses many of these concerns, providing an open platform for research into multiprocessor architectures. The goal of the Sunflower tool suite is to provide an actively evolving ecosystem of both hardware prototypes and simulation / analysis tools, for low-power embedded systems, with an emphasis on the investigation of issues relating to energy-efficiency, energy acquisition, fault-tolerance, and impact of hardware deployments on the environment.

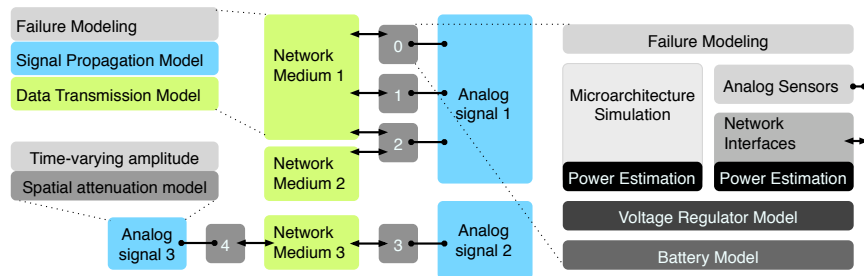


Fig. 1. Illustrative example of the Sunflower full-system simulator’s organization.

On the side of simulation, the *Sunflower full-system simulator* [Stanley-Marbell and Marculescu, 2007b] (Figure 1) enables the evaluation of micro- and system-architectures for networked embedded systems, modeling many aspects of both the hardware platforms and the environments within which they execute.

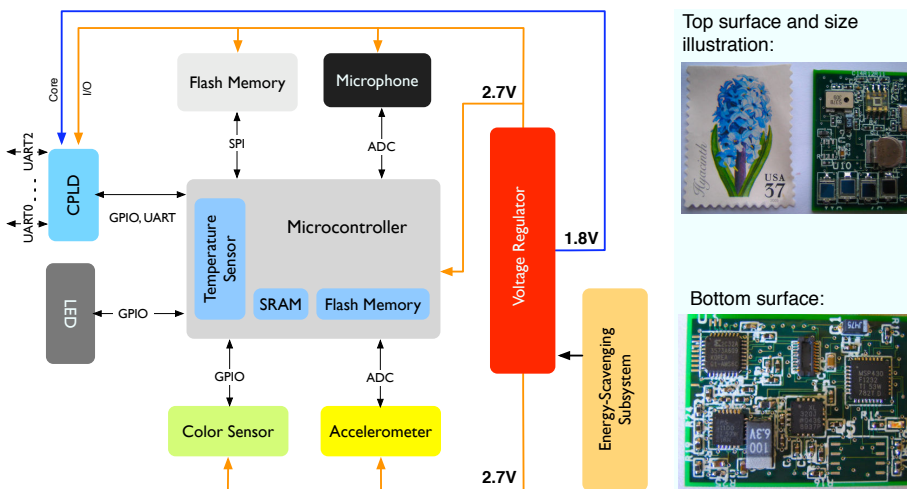


Fig. 2. System architecture of the Sunflower sensor platform (left), and pictures of the current hardware prototype (right).

The *Sunflower sensor platform* [Stanley-Marbell and Marculescu, 2007a] (Figure 2), is one physical realization of components modeled within the Sunflower full-system simulator, enabling the calibration and validation of simulator configurations against real hardware implementations.

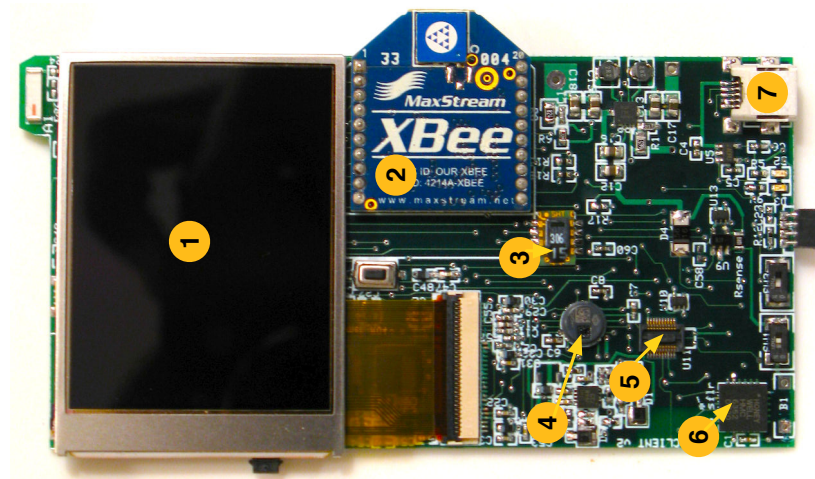


Fig. 3. The *Sunflower mobile client* platform has a 320×240 pixel color display (1), humidity/temperature (3), and pressure sensors (4), and a digital compass (6). It includes a dedicated expansion connector (5), USB (7), and an 802.15.4 radio interface (2), as well as a microSD slot for flash memory or peripheral cards. The primary source of computing power is a 32-bit ARM7 implementation (AT91SAM7S256) with 64 KB of on-chip RAM and 256 KB of on-chip flash memory (on the rear side of the device), and the system is powered by a thin 2000 mAh rechargeable lithium polymer battery.

The *Sunflower mobile client platform* (Figure 3, with its system architecture shown in Figure 4) is another member of the suite of hardware tools, and is intended to be used, for example, to study the hardware and software aspects of building low-power mobile computing platforms containing multiple processing elements.

Additional hardware platforms with complementary hardware capabilities (e.g., wireless communication interfaces and graphical displays) are planned, and it is intended to employ these platforms as a framework for the implementation of ideas by a community of researchers who may not necessarily have interests or expertise in hardware design, but might require specific hardware facilities to enable the investigation of novel software algorithms.

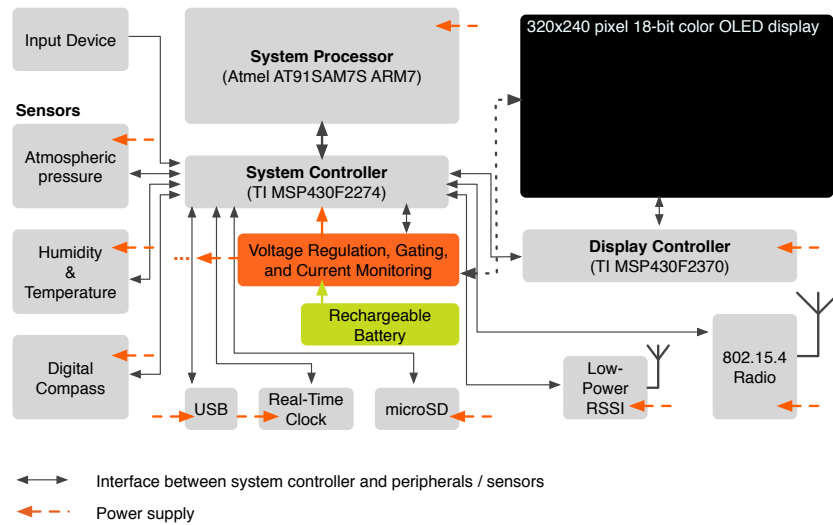


Fig. 4. System architecture; The *system controller* implements the low-level software interfaces to peripherals and sensors, and applications run over the *system processor*, an ARM processor running FreeRTOS.

3 Tutorial Outline

The planned outline of topics to be discussed in the tutorial is as shown below:

Topic	Duration
Motivation for the suite of hardware and simulation tools	5 min
Sunflower simulator overview	5 min
Hands-on simple example and setup: single-processor embedded system	5 min
Microarchitecture simulation and its implementation	5 min
Power estimation, power supply, and battery modeling implementation	10 min
Modeling analog signals external to processor	10 min
Communication interconnect modeling: wired and wireless networks	10 min
Larger example: modeling a network of processors	10 min
Support for distributions of random variables and constants in simulation	5 min
Running MiBench, ALPBench, SPEC and other benchmarks on the simulator	5 min
Break	(30 min)
Sunflower hardware platforms brief overview	10 min
Sunflower sensor node architecture	10 min
Sunflower handheld architecture	10 min
Hardware tools	5 min
Compilation tools	5 min
Using the Sunflower simulator to emulate the Sunflower hardware	10 min
Submitting requests for hardware features in the periodic HW revisions	10 min
Summary, and pointers to further available resources	5 min
Q & A	10 min

4 Contributor Profiles

Phillip Stanley-Marbell is a post-doctoral researcher at the Technische Universiteit Eindhoven. He received the PhD in computer engineering from Carnegie Mellon University in 2007, and is the principal architect and implementor of the Sunflower simulation framework and hardware platforms. Prior to, and during his PhD, he held industrial positions at Bell-Labs (Lucent Microelectronics), Philips Consumer Communications, and NEC research labs. His research interests include energy-resource constrained and failure-prone systems.

Diana Marculescu received the Dipl. Eng. degree in computer science from University Politehnica of Bucharest, Romania, in 1991, and the Ph.D. degree in computer engineering from the University of Southern California, Los Angeles, in 1998. She is currently an Associate Professor of Electrical and Computer Engineering at Carnegie Mellon University, Pittsburgh, PA. Her research interests include energy-aware computing, CAD tools for low-power systems, and emerging technologies (such as electronic textiles or ambient intelligent systems).

Dr. Marculescu is the recipient of a National Science Foundation Faculty Career Award (2000-2004), an ACM-SIGDA Technical Leadership Award (2003), and the Carnegie Institute of Technology George Tallman Ladd Research Award (2004). She was an IEEE Circuits and Systems Society Distinguished Lecturer (2004-2005) and is the Chair of the ACM Special Interest Group on Design Automation (SIGDA).

5 Participant Notes

This section is intended to help the participant organize any notes they may have under the different topic headings covered during the tutorial.

5.1 Motivation for the suite of hardware and simulation tools

5.2 Sunflower Simulator Overview

5.3 Hands-on simple example and setup: single-processor embedded system

5.4 Microarchitecture simulation and its implementation

5.5 Power estimation, power supply, and battery modeling implementation

5.6 Modeling analog signals external to processor

5.7 Communication interconnect modeling: wired and wireless networks

5.8 Large example: modeling a network of processors

8 P. Stanley-Marbell, D. Marculescu

5.9 Support for distributions of random variables and constants in simulation

5.10 Running MiBench, ALPBench, SPEC and other benchmarks on the simulator

5.11 Break

5.12 Sunflower hardware platforms brief overview

5.13 Sunflower sensor node architecture

5.14 Sunflower handheld architecture

5.15 Hardware tools

5.16 Compilation tools

5.17 Using the Sunflower simulator to emulate the Sunflower hardware

5.18 Submitting requests for hardware features in the periodic hardware revisions

5.19 Summary, and pointers to further available resources

A Participant Questionnaire

A.1 Do you have prior experience with similar platforms or tools?

A.2 What did you expect to get out of the tutorial?

A.3 Was the tutorial of sufficient technical depth, given the 3-hour duration?

A.4 Would you have liked to see more discussion of the simulator usage, or of its implementation?

A.5 Would you have liked to see more discussion of the hardware platform usage, or of their implementations?

A.6 Were the materials provided to participants of the tutorial helpful?

A.7 Do you think there is sufficient documentation of the *simulator* to enable you to probe further?

A.8 Do you think there is sufficient documentation of the *hardware platforms* to enable you to probe further?

A.9 What are the things you *did not* like about the tutorial?

A.10 What are the things you *did* like about the tutorial?

Any other comments:

Bibliography

- Arvind, K. Asanovic, D. Chiou, J. C. Hoe, C. Kozyrakis, S.-L. Lu, M. Oskin, D. Patterson, J. Rabaey, and J. Wawrzynek. Ramp: Research accelerator for multiple processors — a community vision for a shared experimental parallel hw/sw platform. Technical Report UCB/CSD-05-1412, EECS Department, University of California, Berkeley, 2005.
- D. August, S. Girbal, D. Gracia-Perez, G. Mouchard, A. Cristal, and O. Temam. UNISIM: UNited SIMulation environment.
- D. Burger, T. Austin, and S. Bennett. Evaluating Future Microprocessors: The SimpleScalar ToolSet. Technical Report CS-TR-1308, Computer Sciences Department, University of Wisconsin-Madison, 1996.
- J. Gibson, R. Kunz, D. Ofelt, M. Horowitz, J. Hennessy, and M. Heinrich. Flash vs. (simulated) flash: closing the simulation loop. *SIGPLAN Not.*, 35(11):49–58, 2000. ISSN 0362-1340.
- K. Langendoen. Apples, oranges, and testbeds. In *2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 387 – 396, October 2006.
- P. Stanley-Marbell and D. Marculescu. An 0.9 x 1.2", low power, energy-harvesting system with custom multi-channel communication interface. In *DATE '07: Design Automation and Test in Europe*, pages 15 – 20, 2007a.
- P. Stanley-Marbell and D. Marculescu. Sunflower: Full-System, Embedded Microarchitecture Evaluation. *2nd European conference on High Performance Embedded Architectures and Computers (HiPEAC 2007) / Lecture Notes on Computer Science*, 4367:168–182, 2007b.