

General Project Description

Project title: TinyOS component verification

Contractor: ISP RAS (primary developer Nickolay Pakoulin)

Partner: Luxoft

Project start date: January 1, 2004.

Project end date: February 29, 2004.

Goals of the project

The goals of the project were: (1) to develop patterns of applying UniTesK to testing TinyOS applications; (2) to prove feasibility of UniTesK approach to verification of sensor networks.

Project Input

Introduction to Tiny OS

Tiny OS is a component-based platform for development applications for tiny devices (4-16 KBytes of RAM) with sensors and short-range radios. The primary development language for TinyOS applications is **nesC** – an extension of the C programming language. Several frameworks are already built atop TinyOS, for example TinyDB. TinyDB provides access to readings of sensors on devices as reading from virtual database. More information on TinyOS is available from their site <http://webs.cs.berkeley.edu/tos/>.

Object under test

The object under test is a component **Attr** from TinyDB that manages access to sensors readings. We tested implementation of six methods on two interfaces of the component.

Functional requirements were elicited from documentation and source code. We had to use source code because the documentation was incomplete. About 40 functional requirements were elicited.

Process Used

Method: CTesK toolkit. The project features that distinguish it from other UniTesK applications are (1) asynchronous operations in interface; (2) processing an operation on an interface might involve operations on other interfaces.

Development: Development was done on Linux using CTesK toolkit, TinyOS development tools, and common Unix development tools (Emacs, make).

Test harness and tools: The component under test was deployed in TinyOS simulator on a PC under Linux. The testing was performed remotely. Specialized components for TinyOS were developed that provide connection between the test system and the component under test.

Project Effort:

Project duration: 2 months

effort estimate: 2 man-month

Project Results

Tested 6 interface entries. Test suite component sizes:

Test Suite Component	Size, in lines of code
Specification	550
Test Scenarios	200
Mediators	500

Total number of test scenarios: 5.

Total number of test cases: about 50.

Number of defects revealed: 1.

The project has achieved its goals. CTesK applicability to testing embedded systems was evaluated. Patterns for applying CTesK to testing embedded systems with asynchronous operations were suggested.

The project demonstrated that UniTesK is well suited for functional testing of embedded applications.