TUnit

From TinyOS Wiki

Tunit.png

TUnit is a unit testing (http://en.wikipedia.org/wiki/Unit_testing) framework geared for TinyOS and sensor networks. The philosophy surrounding the design of TUnit stems from years of accepted industry standard testing in other languages (http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks) . TUnit is used daily in TinyOS Test Driven Development (http://en.wikipedia.org/wiki/Test-driven_development) , decreasing the time spent manually testing and debugging software by literally hundreds of man-hours.

Contents

Open Source Testing

The TUnit unit testing framework is available open source (http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/tunit/) to be used both publicly and privately.

Unit tests can be contributed by the public to the public automated unit testing system in tinyos-2.x-contrib/tests.

Privately, any group can run TUnit standalone or as an automated process in their own team development environment.

Requirements

- TUnit Requirements

Tutorials

- How TUnit Works
  - TUnit Philosophy
  - TUnit Test Flow
  - TUnit Assertions
- Setting up TUnit
  - suite.properties
  - tunit.xml
- Single-Node Unit Testing
  - Your First Test
  - Assertion Playground

  - State Interface Test
  - Characterization Testing with Statistics
- Multi-Node Unit Testing

Case Studies

- TestRxFifo (http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/tunit/tests/tinyos-2.x-contrib/blaze/tos/chips/ccxx00/integration/cc1100/receive/TestRxFifoReceive/)
  - Isolation test testing the receive functionality of a radio stack from the rest of the system, allowing us to test edge cases observed by receiving various types of data in the software implemented RX FIFO.

- AckBehavior (http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/tunit/tests/tinyos-2.x-contrib/blaze/tos/chips/ccxx00/isolation/AckBehavior/)
  - Isolation test of the acknowledgements layer of the CC1100/CC2500 dual radio stack.

- CsmaBehavior (http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/tunit/tests/tinyos-2.x-contrib/blaze/tos/chips/ccxx00/isolation/CsmaBehavior/)
  - Isolation test of the CSMA layer of the CC2500 radio stack.

- TestMac (http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/tunit/tests/tinyos-2.x/tos/chips/general_radio_tests/TestMac/)
  - Multi-Node Testing and Characterization: Multiple statistics logging

- TestRxThroughputWithNoCca (http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/tunit/tests/tinyos-2.x/tos/chips/cc2420/TestRxThroughputWithNoCca/)
  - Multi-Node Test: Upon receiving the first message from the Driving Node, the Supporting Node receiver starts a timer and accumulates received messages. When the timer fires, it reports performance and verifies enough packets got through. The performance is characterized with statistics logging.

- TestSwAcks (http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/tunit/tests/tinyos-2.x/tos/chips/cc2420/TestSwAcks/)
  - Multi-Node Test: An idle Supporting Node simply sends back an acknowledgment to the Driving Node.

- TestLplDefaultContinuousDelivery (http://tinyos.cvs.sourceforge.net/tinyos/tinyos-2.x-contrib/tunit/tests/tinyos-2.x/tos/chips/cc2420/TestLplDefaultContinuousDelivery/)
  - Multi-Node Test: A Test Suite where 2 to 19 Supporting Nodes deliver LPL messages to a single listener, the Driving Node. At the end of a period of time, a dynamic threshold is calculated at +/-10% and assertions are made against each of the nodes in the network to determine if the channel is being shared fairly amongst the multiple transmitters.

Progress

- TUnit Feature Requests
- TUnit Known Issues
- TUnit Recent Updates

External Links

- TinyOS 2.x Public Automated Unit Testing (http://www.lavalampmotemasters.com) (http://www.lavalampmotemasters.com)

- Three Rules of Test-Driven Development (http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd)
- Six Rules of Unit Testing (http://radio.weblogs.com/0100190/stories/2002/07/25/sixRulesOfUnitTesting.html)
- Big Visible Charts (http://www.xprogramming.com/xpmag/BigVisibleCharts.htm)
- Testing, fun? Really? (http://www.ibm.com/developerworks/library/j-test.html)
- Introduction to Test-Driven Design (http://www.agiledata.org/essays/tdd.html)

Retrieved from "http://tinyos.stanford.edu/tinyos-wiki/index.php?title=TUnit&oldid=2998"
Category: TUnit

- This page was last modified on 26 February 2010, at 03:25.
- This page has been accessed 45,841 times.