WIKIPEDIA
The Free Encyclopedia

TinyOS

TinyOS is an embedded, component-based operating system and platform for low-power wireless devices, such as those used in wireless sensor networks (WSNs), smartdust, ubiquitous computing, personal area networks, building automation, and smart meters. It is written in the programming language nesC, as a set of cooperating tasks and processes. It began as a collaboration between the University of California, Berkeley, Intel Research, and Crossbow Technology, was released as free and open-source software under a BSD license, and has since grown into an international consortium, the TinyOS Alliance.

TinyOS has been used in space, being implemented in ESTCube-1.

TinyOS

Developer	TinyOS Alliance
Written in	nesC
OS family	Embedded operating systems
Working state	Current
Source model	Open source
Initial release	2000
Latest release	2.1.2 / August 20, 2012
Repository	github.com/tinyos/tinyos-main (https://github.com/tinyos/tinyos-main)
Marketing target	Wireless sensor networks
Available in	English
License	BSD
Official website	tinyos.net (http://tinyos.net)

Implementation

TinyOS applications are written in the programming language nesC, a dialect of the C language optimized for the memory limits of sensor networks. Its supplementary tools are mainly in the form of Java and shell script front-ends. Associated libraries and tools, such as the nesC compiler and Atmel AVR binutils toolchains, are mostly written in C.

TinyOS programs are built of software components, some of which present hardware abstractions. Components are connected to each other using interfaces. TinyOS provides interfaces and components for common abstractions such as packet communication, routing, sensing, actuation and storage.

TinyOS is fully non-blocking: it has one call stack. Thus, all input/output (I/O) operations that last longer than a few hundred microseconds are asynchronous and have a callback. To enable the native compiler to better optimize across call boundaries, TinyOS uses nesC's features to link these callbacks, called events, statically. While being non-blocking enables TinyOS to maintain high concurrency with one stack, it forces programmers to write complex logic by stitching together many small event handlers. To support larger computations, TinyOS provides tasks, which are similar to a Deferred Procedure Call and interrupt handler bottom halves. A TinyOS component can post a task, which the OS will schedule to run later. Tasks are non-preemptive and run in first in, first out order. This simple concurrency model is typically sufficient for I/O centric applications, but its difficulty with CPU-heavy applications has led to developing a thread library for the OS, named TOSThreads. TOSThreads are unmaintained and have been deprecated.[1]

TinyOS code is statically linked with program code and is compiled into a small binary, using a custom GNU toolchain. Associated utilities are provided to complete a development platform for working with TinyOS.

History

TinyOS began as a project at UC Berkeley as part of the DARPA NEST program. It has since grown to involve thousands of academic and commercial developers and users worldwide. (list in reverse chronological order)

- August 2012: TinyOS 2.1.2 released
- April 2010: TinyOS 2.1.1 released.
- August 2008: TinyOS 2.1.0 released.
- July 2007: TinyOS 2.0.2 released. Work on TinyOS 2.1, which involves slight changes to a few interfaces, begins.
- April 2007: TinyOS 2.0.1 released at the 4th TinyOS Technology Exchange in Cambridge, MA.
- November 2006: TinyOS 2.0 released at the SenSys conference in Boulder, CO.
- July 2006: TinyOS 2.0 beta2 released.
- February 2006: TinyOS 2.0 beta1 released at the 3rd TinyOS Technology Exchange in Stanford, CA.
- December 2005: TinyOS 1.1.15, the last 1.1 version, is released.
- July 2005: NEST project concludes.
- June 2004: Working group forms on next steps for TinyOS, based on experiences porting to new platforms. Group agrees to start work on 2.0.
- September 2003 – December 2005: TinyOS begins a periodic minor release process.
- August 2003: TinyOS version 1.1 is released, which includes new nesC features including data race detection.
- September 2002: TinyOS version 1.0, implemented in nesC, is released.

Slyde Analytics Exhibit No. 2006, IPR2024-00833
Page 001

- April 2002: Work on the nesC programming language begins as a collaboration between Intel Research and UC Berkeley.
- February 2002: Berkeley distributes 1000 mica nodes to other participants in the NEST project.
- 2001: Berkeley develops the mica platform and releases TinyOS version 0.6.
- 2000: Berkeley designs the rene platform and partners with Crossbow, Inc., who mass-produces the hardware. TinyOS version 0.43 is made available to the public via SourceForge. Pre-1.0 versions of TinyOS are a mix of C and Perl scripts.
- 1999: First TinyOS platform (WeC) and OS implementations are developed at Berkeley.

Integrated development environments

As of 2010, three integrated development environments (IDEs) are available for TinyOS, as plug-ins for Eclipse:

- *YETI 2*, ETH Zürich (development stopped)
- *XPairtise* by Rasmus Pedersen, Copenhagen Business School
- *TinyDT*, Vanderbilt University

References

1. "TinyOS development repository TOSThreads deprecated" (https://github.com/tp-freeforall/prod/tree/2308921ea2f2a1d48abb24c115a7691e1e400140/deprecated). *TinyOS current development repository*. Retrieved 3 December 2018.

External links

- TinyOS on GitHub (https://github.com/tinyos/tinyos-main)
- TinyOS (https://curlie.org/Computers/Software/Operating_Systems/Network/TinyOS) at Curlie

Retrieved from "https://en.wikipedia.org/w/index.php?title=TinyOS&oldid=1185085129"

Slyde Analytics Exhibit No. 2006, IPR2024-00833
Page 002