Core Working Group Report

Philip Levis (speaking on behalf of the WG)

TTX5
2/22/08

Core WG Charter

- "The TinyOS 2.0 Core Working Group (core) decides what abstractions, services, and interfaces are in the TinyOS 2.0 core system, developing reference implementations when needed. While the core WG focuses on the hardware independent interfaces to hardware-specific abstractions (the basic OS services), it sometimes also addresses basic abstractions that many systems might use, such as data structures."

Core Responsibilities

net2 storage ...

TinyOS Service Interfaces ← Hardware Independent Layers

Core

Hardware
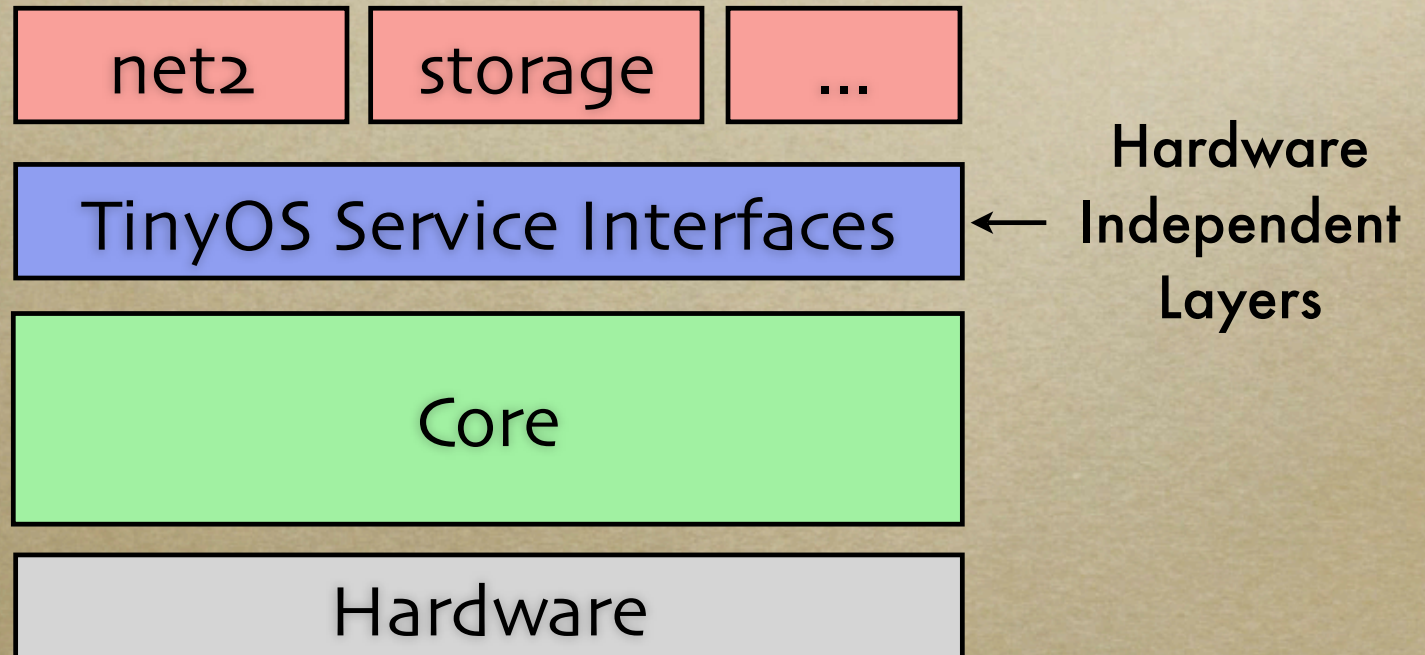
Milestones

- March 20, 2007: All TEPs below 120 sent to SC

- April 29, 2007: TinyOS 2.0.1 release

- July 31, 2007: TinyOS 2.0.2 release

- August, 2007: Move discussion to -devel

- September, 2007: VMWare installations available

- January, 2008: Introduction of Safe TinyOS

- Feb 19, 2008: Port to Crossbow Iris platform

- May, 2008: All TEPs below 120 finalized

- July, 2008: TinyOS 2.1 release

Since TTX4

- Finalized TEPs 1, 102, 115, 116, and 117
- Move to avr-gcc 4.1.2
- IRIS platform
- Low power listening interfaces
- Power locks

Completed

- Finalize TEPs 2, 109, 112, 113, 114
- 802.15.4 MAC layer
- Time synchronization
- Safe TinyOS

In Progress

TEPs

- TEP 1: TEP Format and Keywords
- TEP 102: Timers
- TEP 115: Power Management of Non-Virtualised Devices
- TEP 116: Packet Protocols
- TEP 117: Low-level I/O

- TEP 2: Hardware Abstraction Architecture
- TEP 109: Sensorboards
- TEP 112: Microcontroller Power Management
- TEP 113: Serial Communication
- TEP 114: SIDs: Source and Sink Independent Drivers

avr-gcc 4.1.2

- TinyOS had been using 3.4.3 for 6 years
  - ✳ Does not support more recent MCU, such as atm1281
  - ✳ Reluctance to shift due to inevitable incompatibilities
- Began discussion with avr-gcc developers in October
  - ✳ Settled on 4.1.2 as a recent, stable version
- Established interoperability
  - ✳ Explored best set of optimization flags
  - ✳ Adjusted thresholds, etc. (4.1.2 has more aggressive inlining)
  - ✳ John Regehr's group independently validated results
- 4.1.2 is now the standard in the Debian packages
  - ✳ 2.1 will require it

Crossbow Iris Platform

- T2 port by Janos Sallai and Miklos Maroti

- Atmel ATMega1281 MCU
  - 8kB RAM
  - 8mA active, 8uA sleep
- Atmel RF230 radio (802.15.4 compliant)
  - >300m range (outdoors, line-of-sight)
  - TX: 17mA @ 3dBm, RX: 16mA (-101 dBm receiver sensitivity)
- MCU
  - Timers/Counters, ADC, SPI, I2C, USART, SPI and I2C fully supported
- RF230 Radio stack
  - Interoperable with the CC2420
  - Preliminary timestamping/time synchronization support
  - No LPL support yet (work in progress)
- Deluge/T2 is supported
- Instructions at: http://www.isis.vanderbilt.edu/iris

Low Power Listening

- TEP 105: Low Power Listening

- "BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking." Technical Report SING-08-00.

```
interface LowPowerListening {
  command void setLocalSleepInterval(uint16_t sleepIntervalMs);
  command uint16_t getLocalSleepInterval();
  command void setLocalDutyCycle(uint16_t dutyCycle);
  command uint16_t getLocalDutyCycle();
  command void setRxSleepInterval(message_t *msg, uint16_t sleepIntervalMs);
  command uint16_t getRxSleepInterval(message_t *msg);
  command void setRxDutyCycle(message_t *msg, uint16_t dutyCycle);
  command uint16_t getRxDutyCycle(message_t *msg);
  command uint16_t dutyCycleToSleepInterval(uint16_t dutyCycle);
  command uint16_t sleepIntervalToDutyCycle(uint16_t sleepInterval);
}
```

Power Locks

- TEP 108: Resource Arbitration

- "Integrating Concurrency Control and Energy Management in Device Drivers." In Proceedings of the 21st ACM Symposium on Operating System Principles (SOSP), 2007.

Power Locks

- TEP 108: Resource Arbitration

- Many fine tunings since TTX4

- Detailed paper in SOSP 2007

- Power trace from sample low-rate sensing application with no explicit power management:
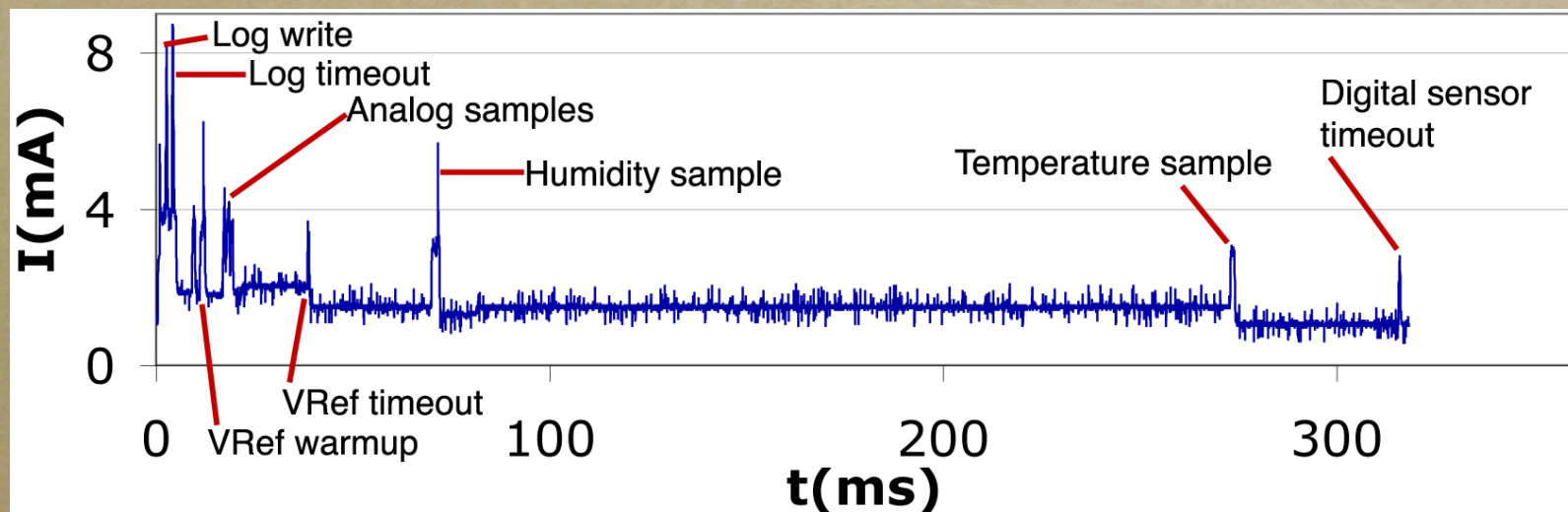
Work in Progress

802.15.4 MAC

- Jan-Hinrich Hauer

- Full implementation of 15.4 MAC
  - ✳ Current stacks only use link layer, not MAC layer
  - ✳ Allows interoperability with other 15.4 networks (e.g., ZigBee)

- Most major components are implemented and tested

- Planned for 2.1 release

Timestamping, Time Synch

- Core WG, effort led by Miklos Maroti
- Establishing interfaces and abstractions time stamping and time synchronization
    - ✳ Packet-level
    - ✳ System-level
- Wrapping up discussion of a packet interface proposal
    - ✳ Should be a TEP soon
- Next will be system interfaces

Safe TinyOS

- John Regehr, David Gay
- Provide optional run-time memory safety checks
  - ✴ No cost if you don't use them
- John will give technical details later
- Great deal of discussion on making uninvasive
  - ✴ Led to a few interface tweaks for better checkability
- Part of 2.1

WG Members

10 Universities and 6 companies:

David Gay, Vlado Handziski, David Moss, Philip Levis, Kevin Klues, Jan Heinrich-Hauer, Andres Koepke, Phillip Huppertz, David Culler, Jonathan Hui, Gilman Tolle, Philip Buonadonna, Cory Sharp, Maxime Muller, Joe Polastre, Martin Turon, Benjamin Greenstein, Prabal Dutta, Lama Nachman, Jan Beutel, Robert Szewczyk, Martin Leopold, John Regehr, Miklos Maroti, Janos Sallai, and more...

Questions