Survey of Embedded Real Time Systems

James Marshall
April 2nd 2013

Embedded, Reactive OS

Embedded: constrained hardware

Reactive: Not HardRT, maybe Soft.

OS: need to provide the programmer with useful abstractions

Not mentioned: many distributed, loosely coupled nodes

Sensor Network

Hundreds of "motes"

- Cheap

Long deployment

Radio communication

Smart Home

Tens of components

- Still Cheap

Heterogeneous sensors and controllers

Communication and power?

Other Applications

"Internet of Things" - ?????

Inventory tracking

Security systems

TinyOS

Targeted at Sensor Networks

Component based

Split-phase execution model

Compile time checks

TinyOS - Execution Model

Reactive system

Events (interrupts, hw & sw)

Tasks (unit of computation)

Commands (communication)

command -> task -> event

TinyOS - Execution Model cont.

Tasks

- Queued
- Run to completion
- Atomic with respect to other tasks

Events / Commands

- Interrupt
- Must be short
- May "post" a task

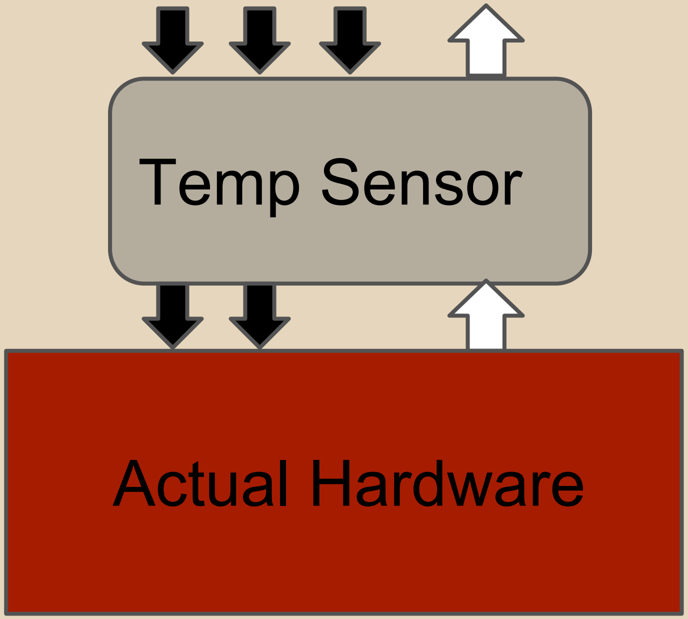
TinyOS - Component Based

Commands ⬇

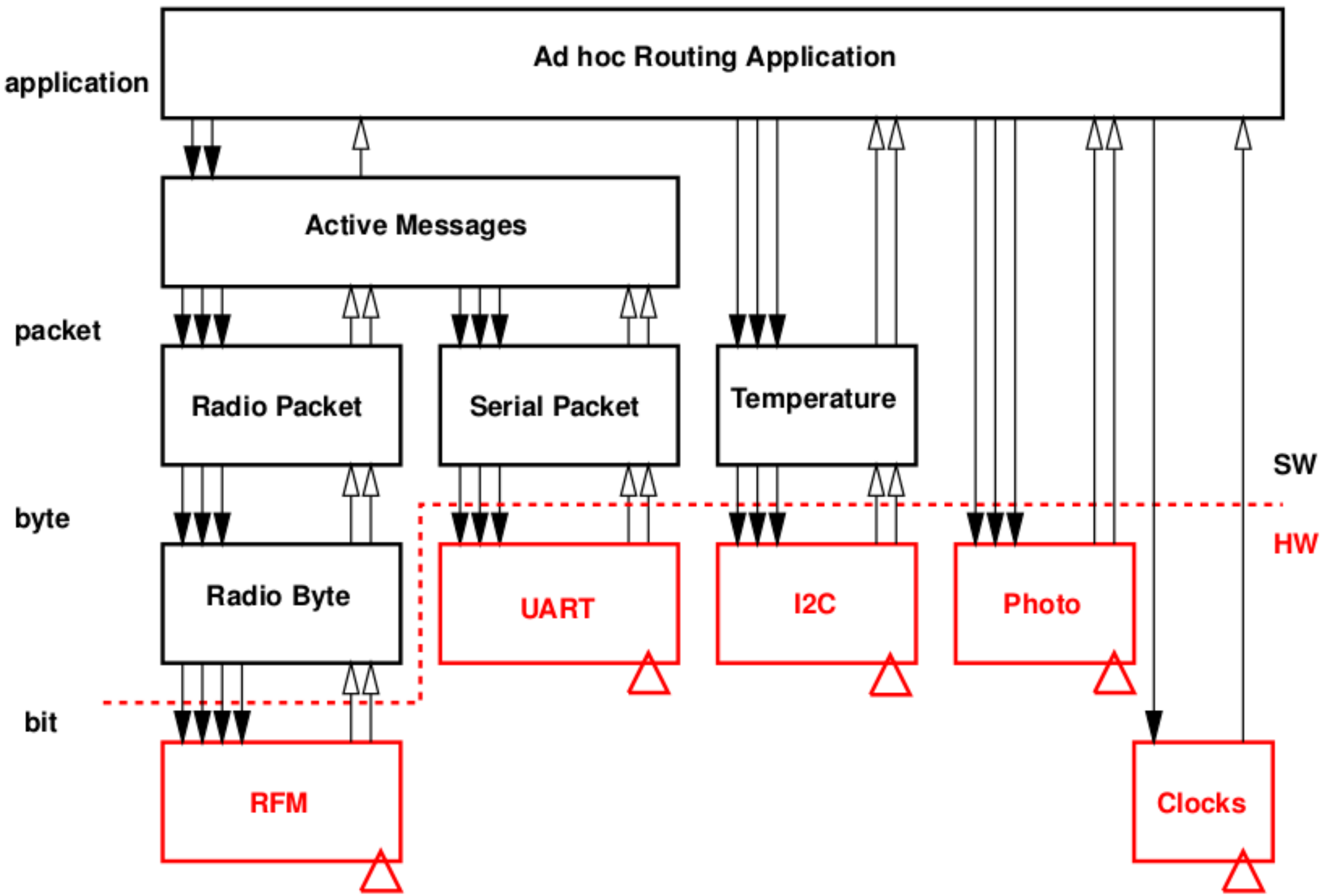
Events ⇧

Component ▭

Tasks?



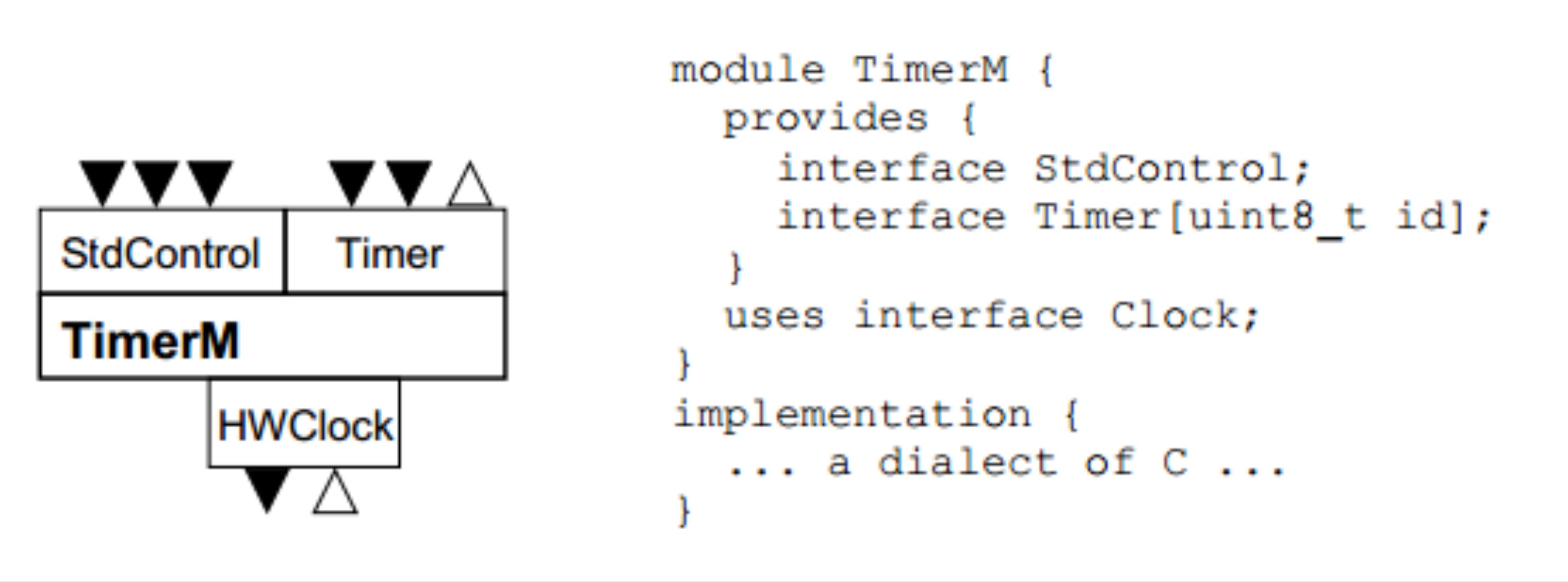
TinyOS - example



TinyOS-nesC

Restricted version of C

nesC -> verifier / opt. -> C -> gcc -> binary



```
module TimerM {
  provides {
    interface StdControl;
    interface Timer[uint8_t id];
  }
  uses interface Clock;
}
implementation {
  ... a dialect of C ...
}
```

TinyOS - nesC - Checking

Detects *most* data race conditions...

Can mark atomic sections of tasks

Can mark variables as "norace"

But how?

No function pointers.

No dynamic memory.

Only worried about data reachable from at least one event.

TinyOS - nesC - Optimizations

Compiled into a single program.

Address space is shared - everything is a function call.

Other optimizations.

EMERALDS

Microkernel

Multi-threaded processes

Provides semaphores

Everyone plays nice

EMERALDS - Play nice?

Processes are cooperative

ex. programmers are trusted to access a device driver only through the correct interface, but not enforced

But may still be faulty

ex. separate address spaces are provided...

EMERALDS - Kernel Address Space

Functionality is pushed to user level (device drivers)

Kernel is always mapped to the same space

No context switch, just traps

System calls can use the stack

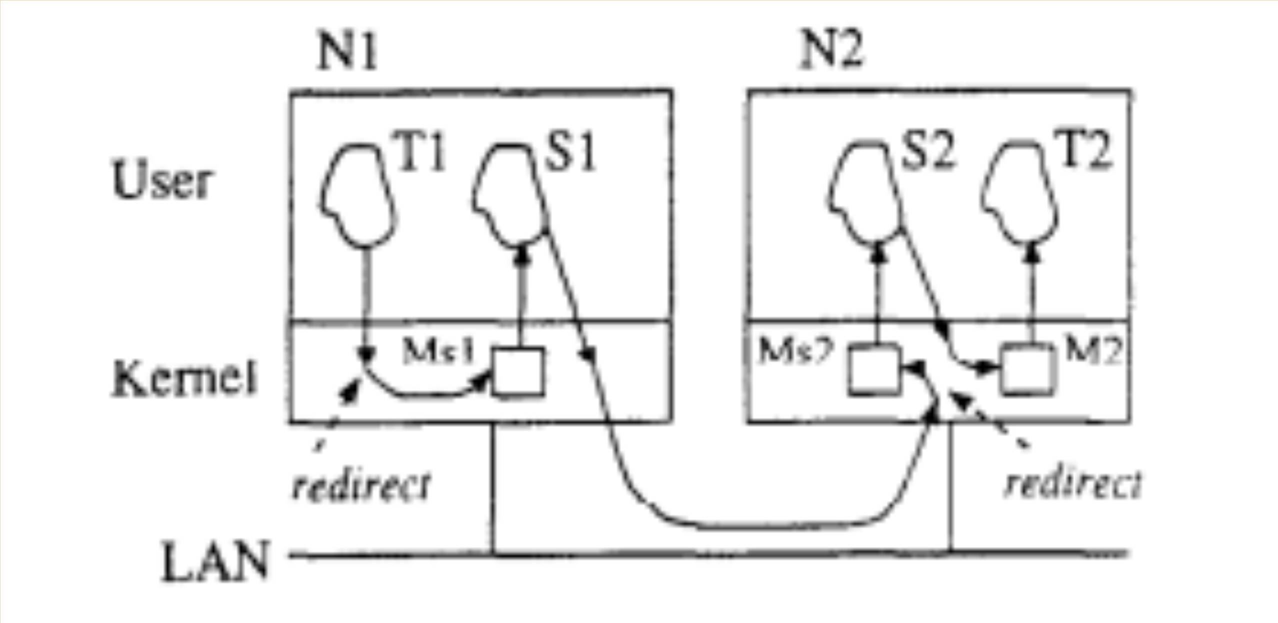
Everything is always in memory

EMERALDS - Threads

Priority can be set dynamically

Inherited to prevent inversion

Mailboxes provide IPC, local and across nodes



Contiki

Event-driven microkernel

Multi-threaded as a user library

Reprogrammable at runtime

Contiki - Process

Replaceable

Is an event handler, may have a poll handler

Communicate by posting events

Contiki - Events & Interrupts

Similar to "tasks" in TinyOS

Run to completion

All share the same address space

Only preemptable by interrupts

Interrupts can not create events, can only request polling

Contiki - Dynamic Code

The kernel is in the "Core"

- mediates services (shared libraries)
- provides communication (event handlers)
- program loader
- Made at compile time, application specific

The rest is all programs (collections of processes)

- Replaceable

Summary of Provided Abstractions

	Unit of Execution	Concurrency	Communication	Architecture
TinyOS	Split - Phase task / event	run to completion	Events and function calls	statically compiled component based
EMERALDS	Multi-threaded process	semaphores	Mailboxes, local and over network	microkernel
Contiki	"process" - events, registered with handlers	run to completion	Events	microkernel

Tiny References

EMERALDS: A microkernel for embedded real-time systems

KM Zuberi… - Real-Time Technology and Applications …, 1996 - ieeexplore.ieee.org

http://sslab.cs.nthu.edu.tw/~kclee/meeting/EMERALDS%20A%20Microkernel%20for%20Embedded%20Real%20Time%20Systems.pdf

The nesC language: A holistic approach to networked embedded systems

D Gay, P Levis, R Von Behren, M Welsh… - Acm Sigplan …, 2003 - dl.acm.org

http://www.dbis.ethz.ch/education/ss2007/tatbul/hotdms/papers/nesc.pdf

TinyOS: An operating system for sensor networks

P Levis, S Madden, J Polastre… - Ambient …, 2005 - books.google.com

From: Ambient Intelligence, edited by Werner Weber, Jan Rabaey, Emile H.L. Aarts

http://www.dbis.ethz.ch/education/ss2007/tatbul/hotdms/papers/tinyos_chapter.pdf

Contiki-a lightweight and flexible operating system for tiny networked sensors

A Dunkels, B Gronvall… - Local Computer Networks, …, 2004 - ieeexplore.ieee.org

http://redback.sics.se/~adam/dunkels04contiki.pdf

A survey of configurable, component-based operating systems for embedded applications

LF Friedrich, J Stankovic, M Humphrey… - Micro, …, 2001 - ieeexplore.ieee.org

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.7260&rep=rep1&type=pdf