

# Software Verification for TinyOS Applications

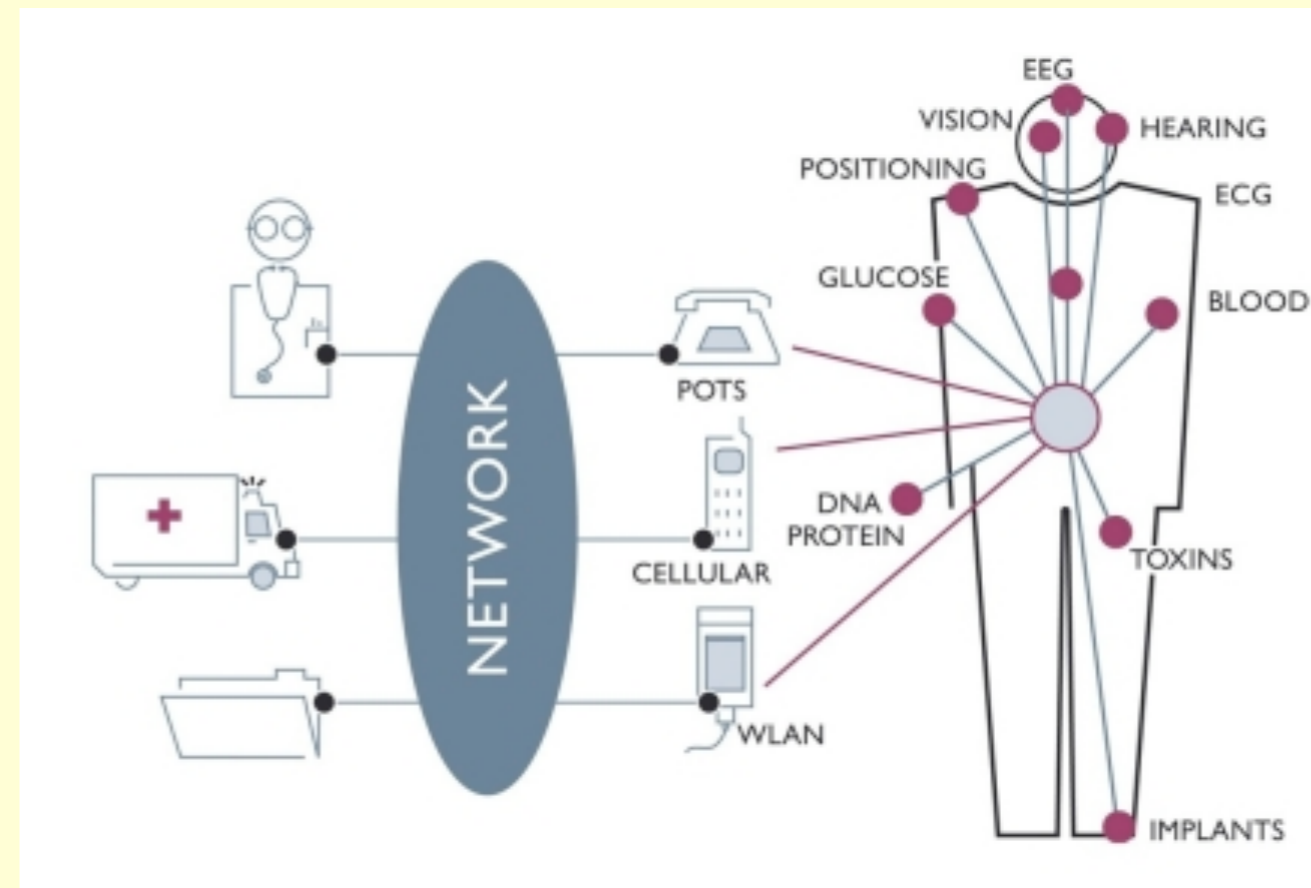
Doina Bucur and Marta Kwiatkowska



# Overview

# Sensor networks

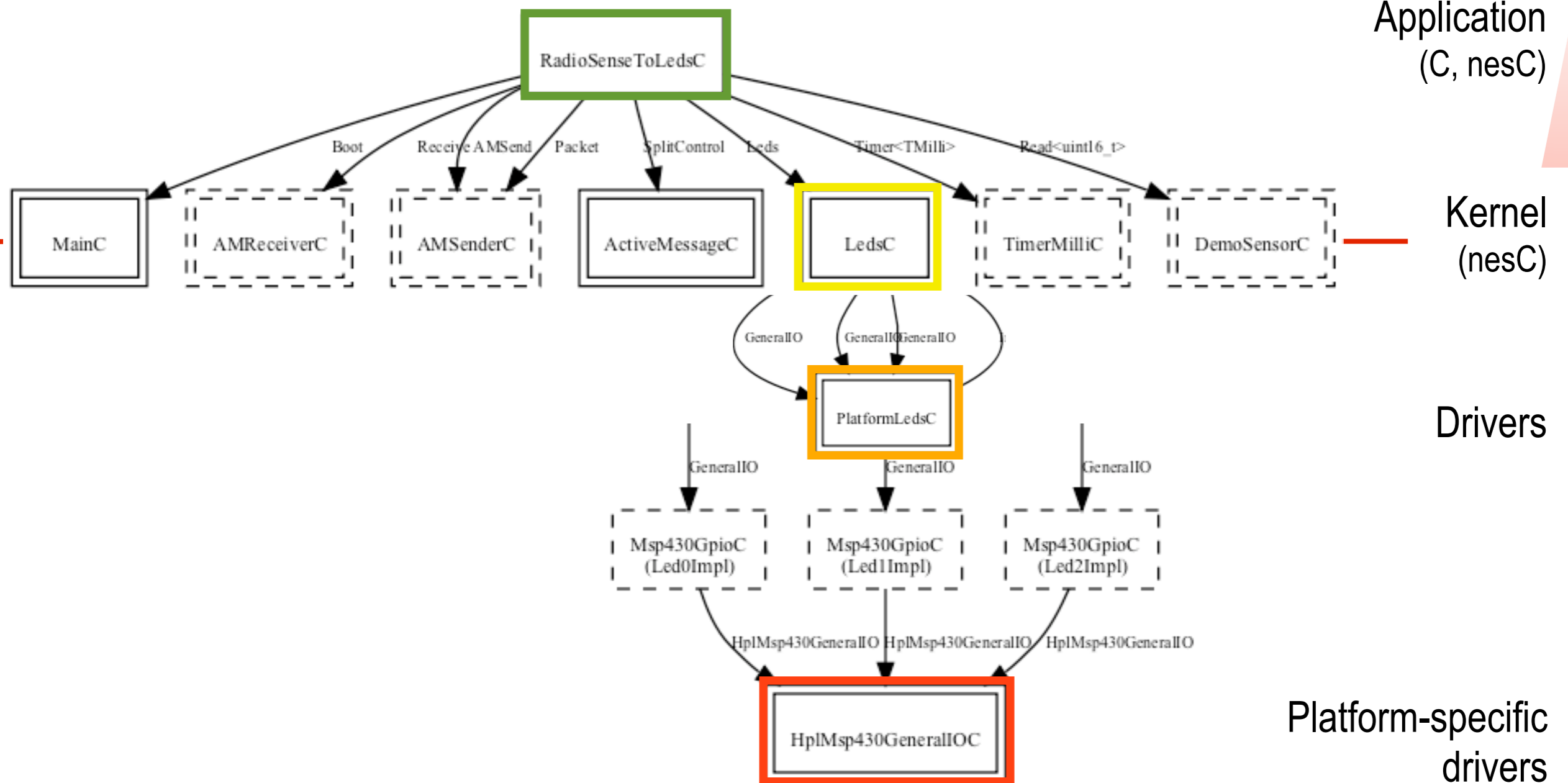
- Pervasive Healthcare.  
Body Sensor Networks  
[vip.doc.ic.ac.uk/bsn](http://vip.doc.ic.ac.uk/bsn)
- Sensor architecture / OS.  
TinyOS:
  - Modern OS and language in an embedded system



# Software verification

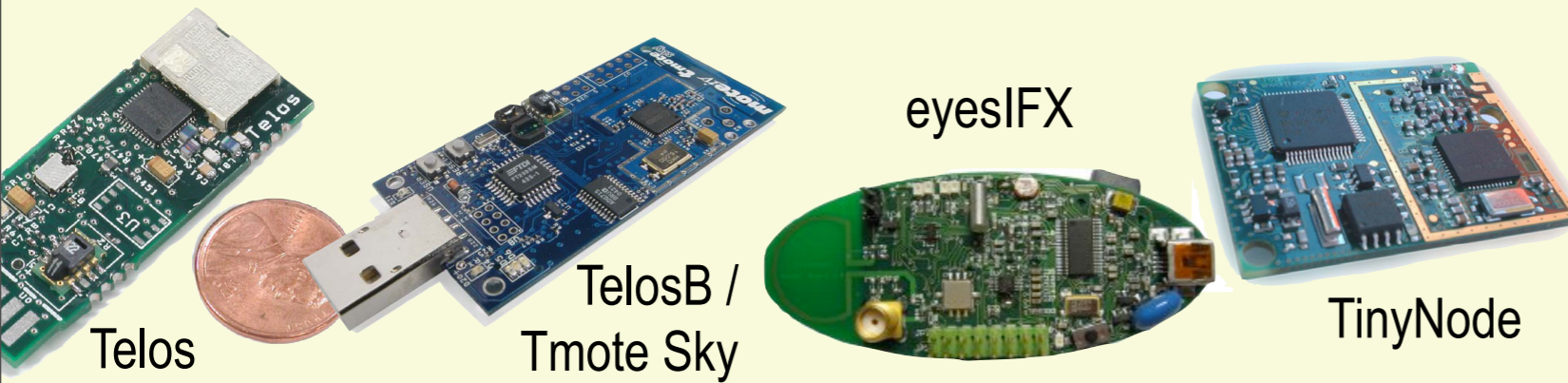
- Accountable applications
- **Verification** vs. simulation / formal verification
- Advances in software verification
- **CProver** tools [[cprover.org](http://cprover.org)] . Counterexample-guided abstraction refinement (CEGAR)

# A TinyOS application

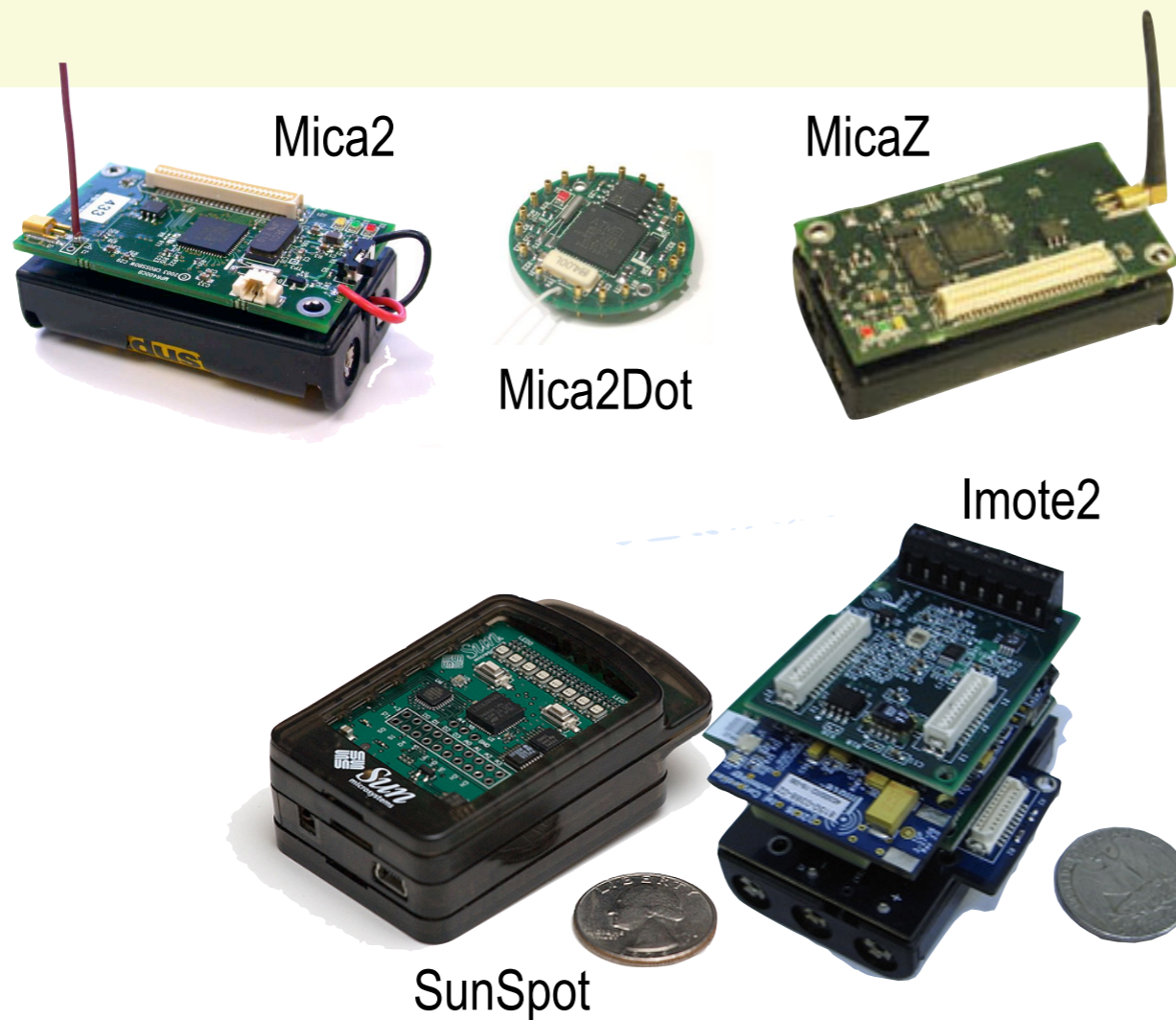


# Architectures

# CPU



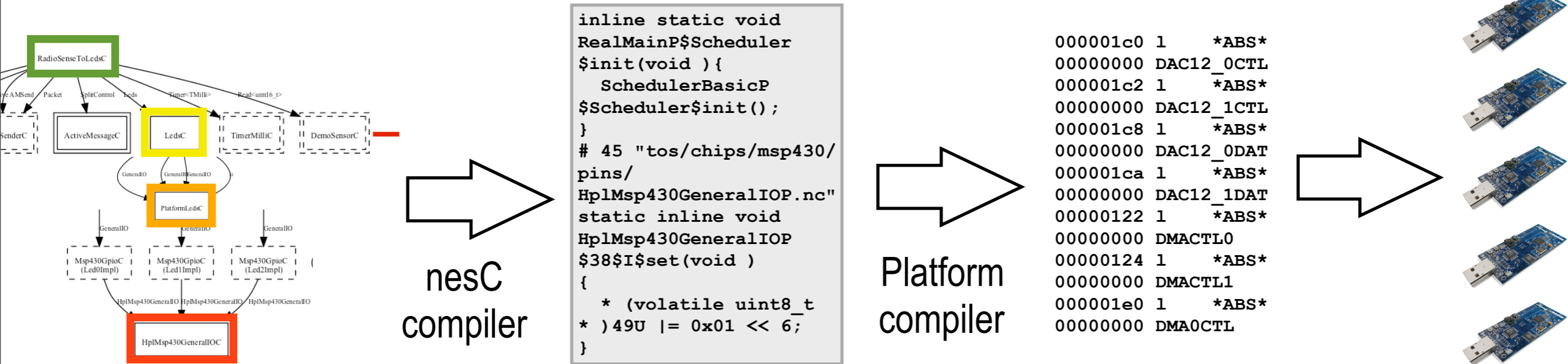
MSP430  
(Texas Instruments)



AVR  
(Atmel)

ARM

# ...and the software tool chain



Components  
(C, nesC)

Platform-specific  
inlined program  
(C + asm)

Machine code



## Verification, but at which stage?

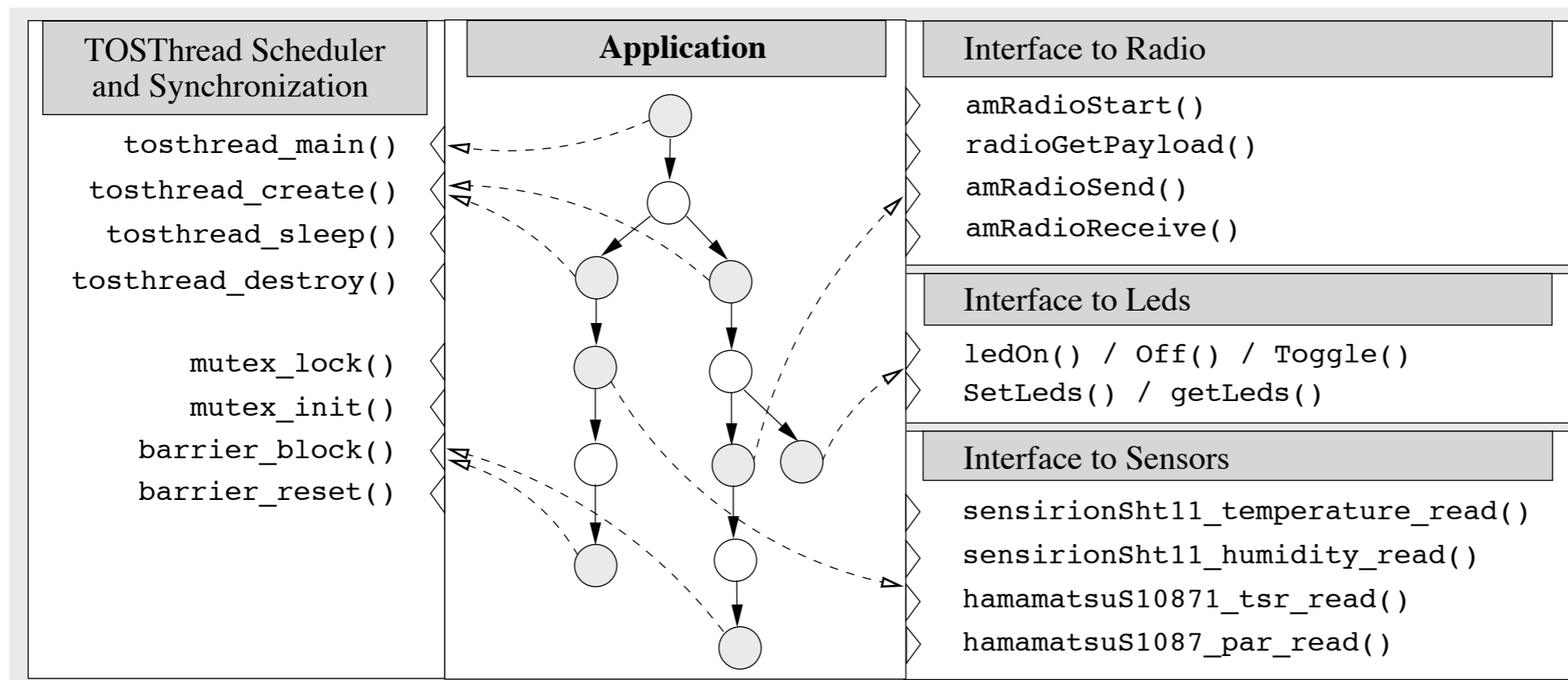
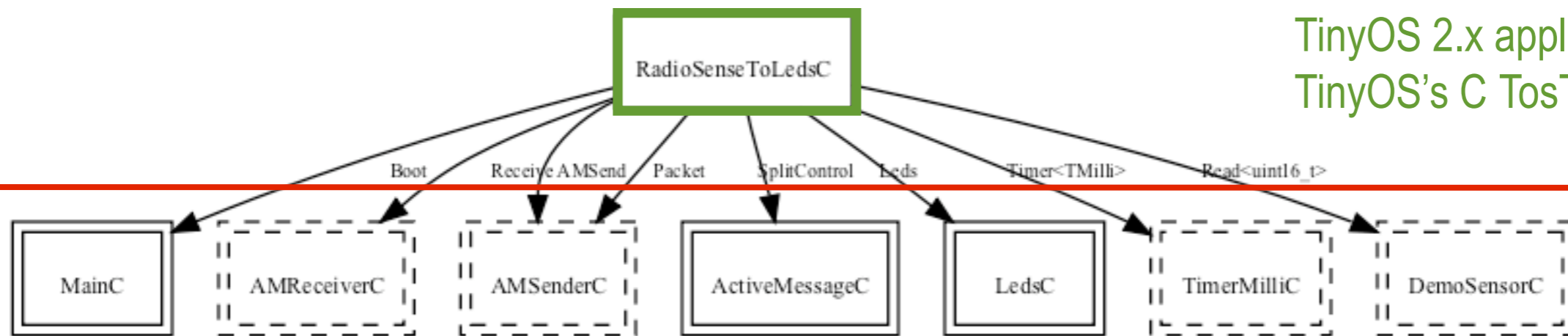
# Bug-Free Sensors: The Automatic Verification of Context-Aware TinyOS Applications

with **Marta Kwiatkowska**

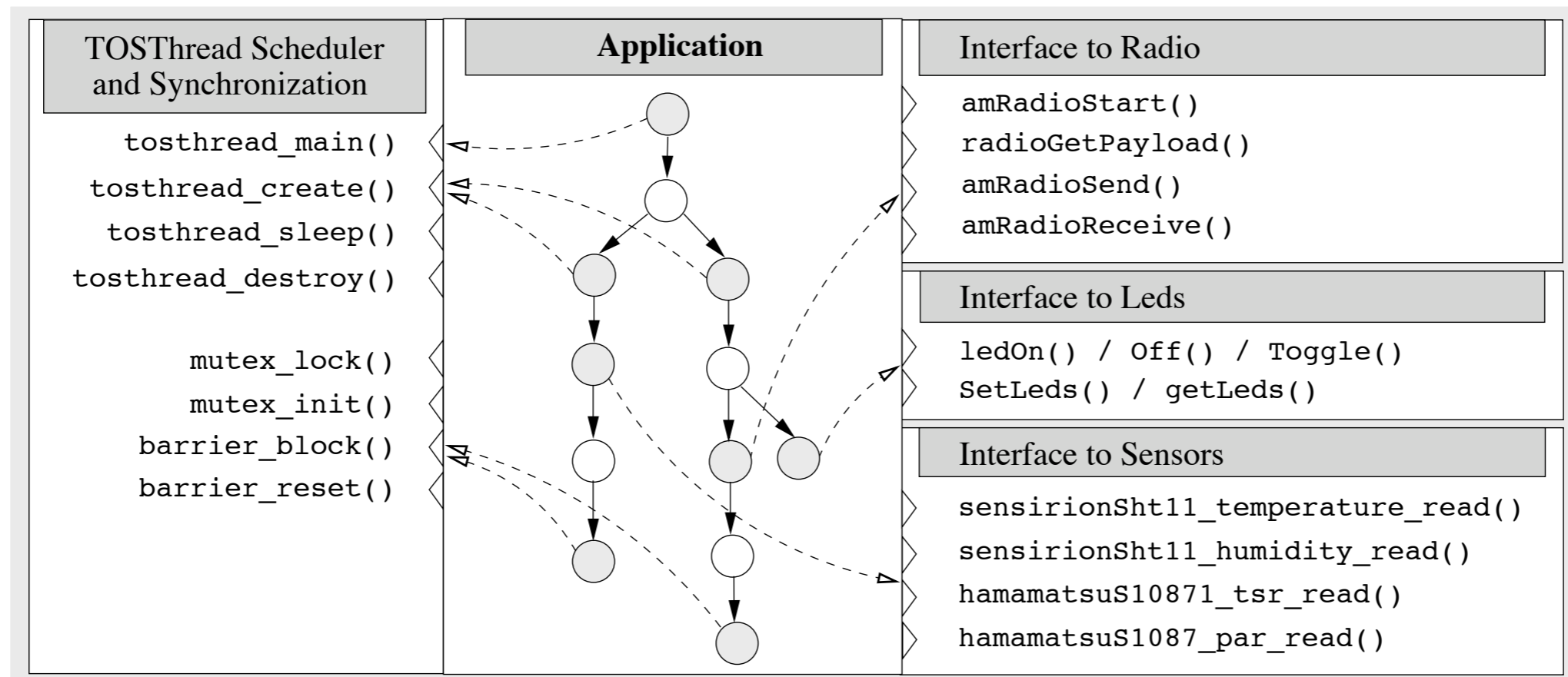
Proceedings of the **European Conference on Ambient Intelligence (Aml 2009)**  
Springer Lecture Notes in Computer Science (LNCS), Nov 2009

# Which software?

TinyOS 2.x applications written in TinyOS's C TosThreads API.



# What method?



Model kernel services, ensuring that their interface behaviour is preserved.

# Which programming errors?

**Table 2.** Categories of bugs in generic concurrent software

---

**Data race** Multithreaded (write) access to shared resource. Not necessarily a bug.

---

**Atomicity violation** Failure to enforce the atomicity of a code region.

---

**Order violation** Failure to enforce execution order between two code regions.

---

**Deadlock** A thread's failure to release a lock-like resource, halting execution.

---

Application (Threads/LOC)	Claim line	Verified?	Time	Bug: context awareness	Bug: concurrency
<i>Blink</i> 4/64	66	yes	2.9s	-	-
<i>SenseAndSend</i> 6/347	79	no	32.2s	interface use	order violation
	136	no	1m08s	sensing exception	-
	146	yes	4m25s	-	-
<i>PatientNode</i> 6/439	172	yes	29.9s	(interface use)	(order violation)
	254	yes	3m55s	(sensing exception)	-
	230	no	35m07s	network exception	deadlock
	268	yes	2m38s	(false reasoning)	-
	262	yes	61m12s	(false reasoning)	-

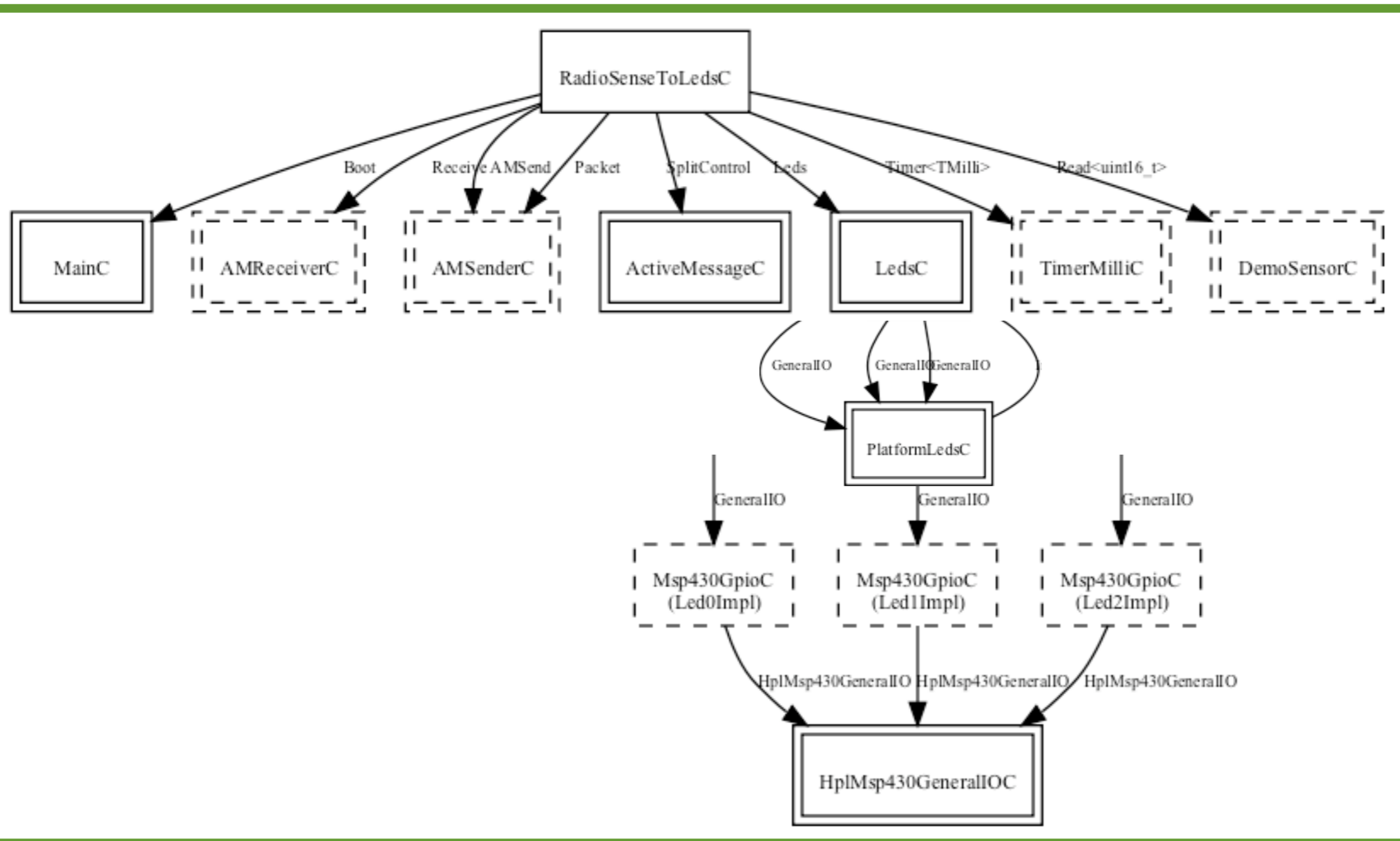
# Software Correctness for TinyOS

with **Marta Kwiatkowska**

In preparation.

# Which software?

TinyOS x.x applications written in any API.



Application  
(C, nesC)

Kernel  
(nesC)

Drivers

Platform-specific  
drivers

# What method?

Model the microcontroller's working:  
**memory map**, **interrupt system**.

```
# 53 "tinyos-1.x/tos/platform/telos/hardware.h"
static inline
void TOSH_MAKE_GREEN_LED_OUTPUT(void)
{
    static volatile uint8_t r __asm ("0x0032");
    r |= 1 << 5;
}

# 90 "tinyos-1.x/tos/platform/msp430/MSP430TimerM.nc"
void __attribute__((interrupt(12))) __attribute__((wakeup)) sig_TIMER_A0_VECTOR(void)
{
    MSP430TimerM$CompareA0$-fired();
}
```

Thank you!

Software Verification for  
TinyOS Applications