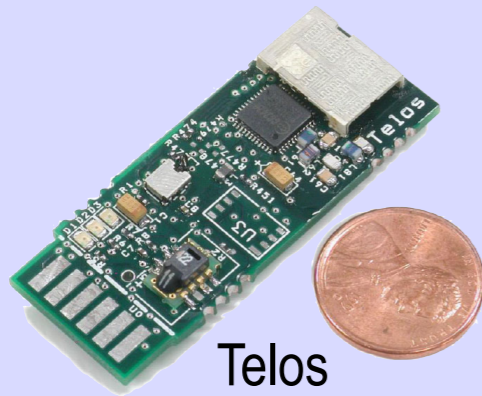


Software Verification for TinyOS Sensors

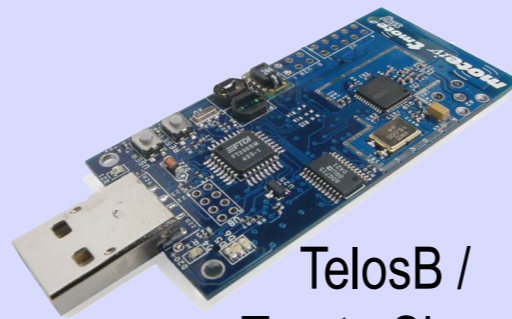
Doina Bucur and Marta Kwiatkowska



Sensor platforms



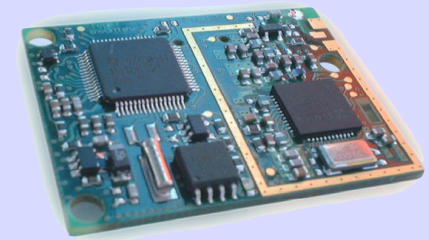
Telos



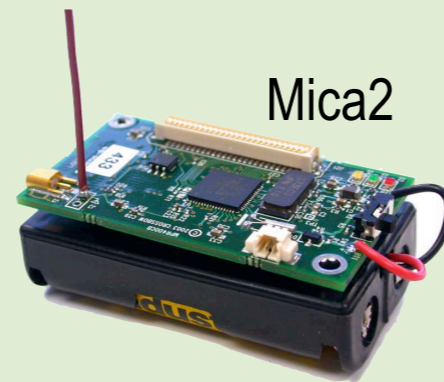
TelosB /
Tmote Sky



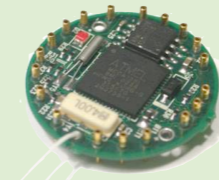
eyesIFX



TinyNode



Mica2



Mica2Dot



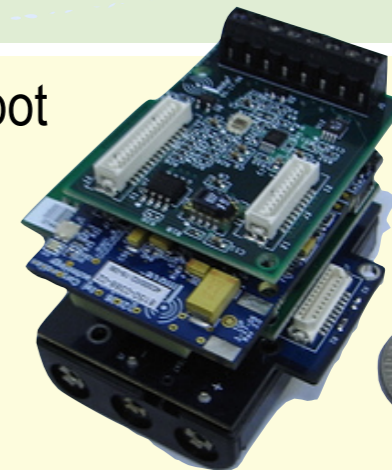
MicaZ

TinyOS
and **NesC**:

OS and language for
embedded systems



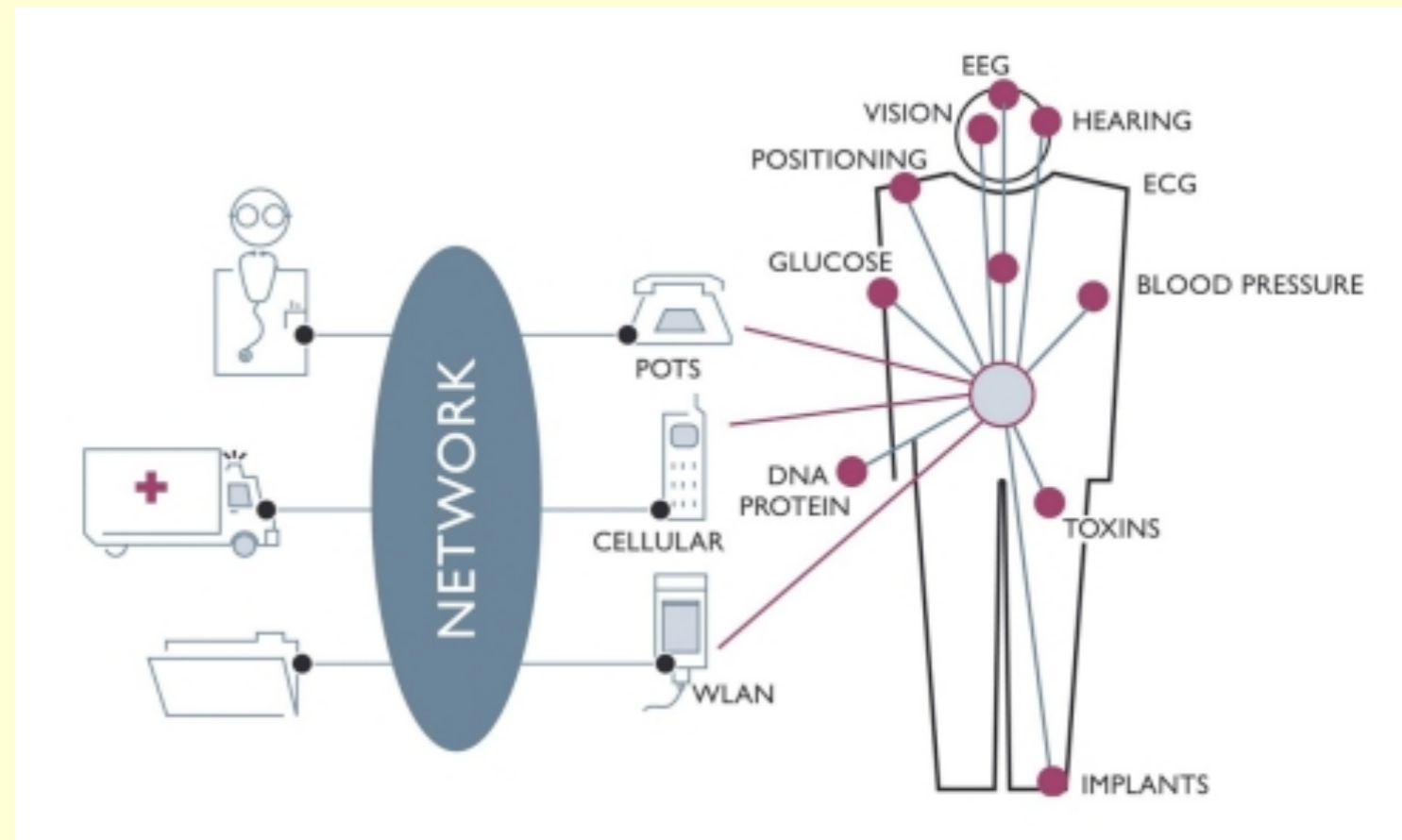
SunSpot



Imote2

Sensor networks

- Pervasive Healthcare.
Body Sensor Networks

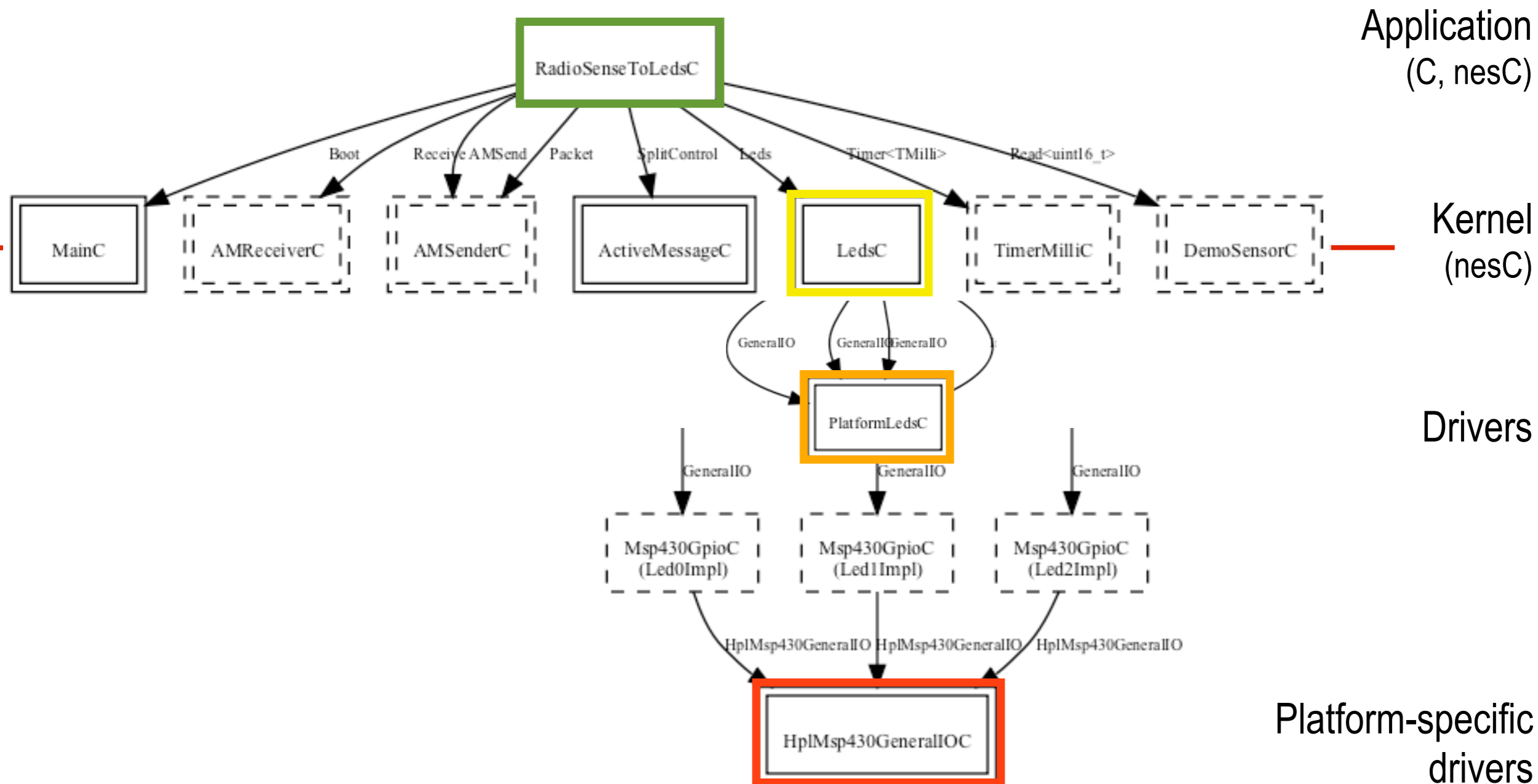


Overview:

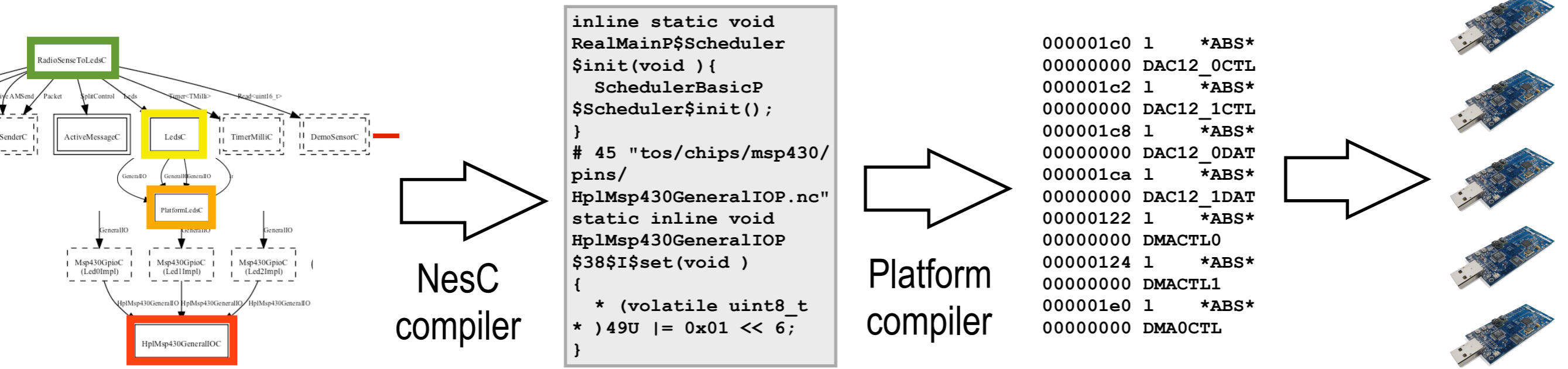
Node-based verification

- Language:
 - NesC, C
- “Bounded” verification:
 - few **IRQ** calls
 - little recursion unwinding (CBMC)
- Specifications:
 - **assertions** upon program states

A TinyOS application



...and TinyOS's compile stages



```

inline static void
RealMainP$Scheduler
$init(void ){
    SchedulerBasicP
    $Scheduler$init();
}
# 45 "tos/chips/msp430/
pins/
HplMsp430GeneralIOP.nc"
static inline void
HplMsp430GeneralIOP
$38$I$set(void )
{
    * (volatile uint8_t
    * )49U |= 0x01 << 6;
}
    
```

```

000001c0 1    *ABS*
00000000 DAC12_0CTL
000001c2 1    *ABS*
00000000 DAC12_1CTL
000001c8 1    *ABS*
00000000 DAC12_0DAT
000001ca 1    *ABS*
00000000 DAC12_1DAT
00000122 1    *ABS*
00000000 DMACTL0
00000124 1    *ABS*
00000000 DMACTL1
000001e0 1    *ABS*
00000000 DMAOCTL
    
```

Components
(threaded, C, NesC)

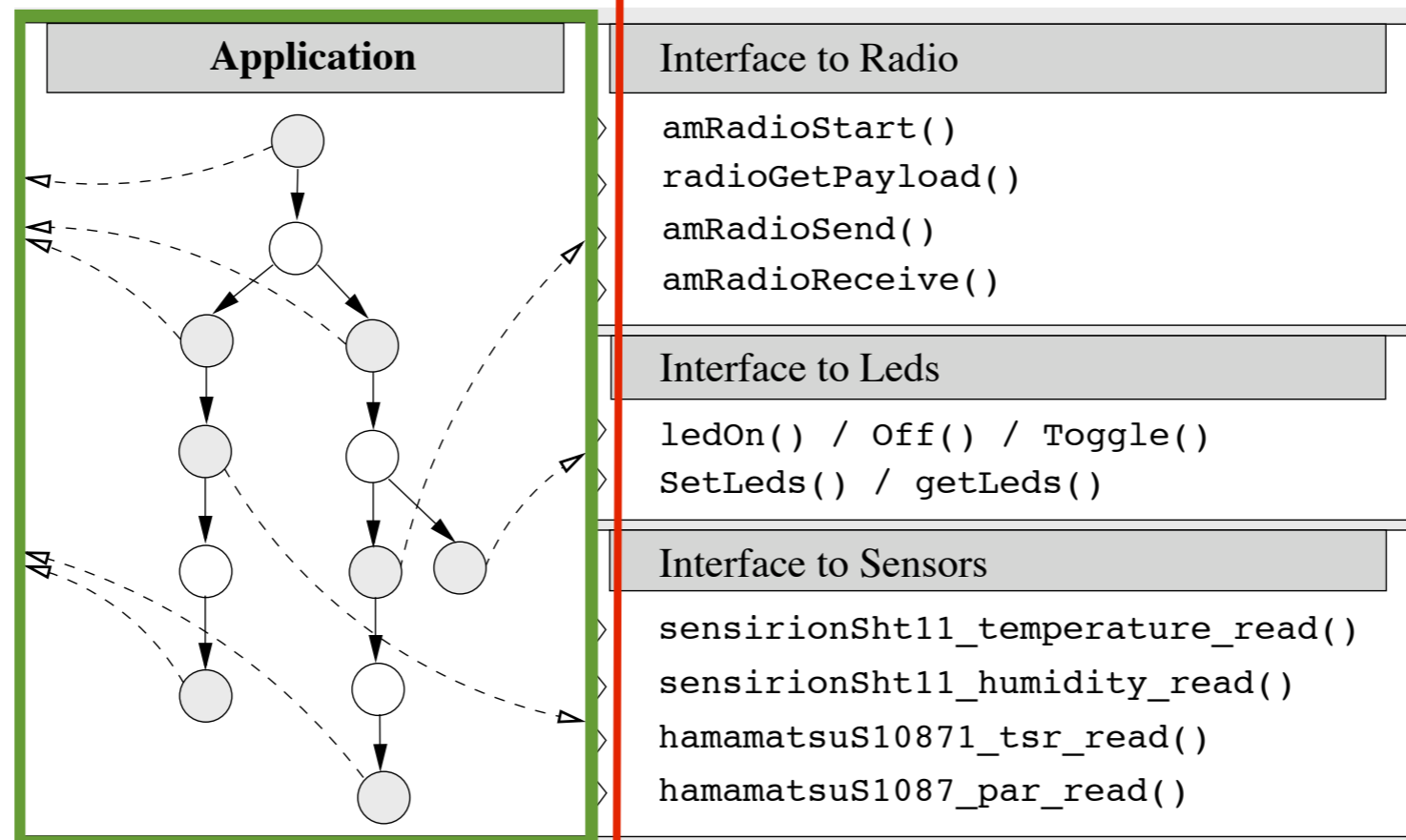
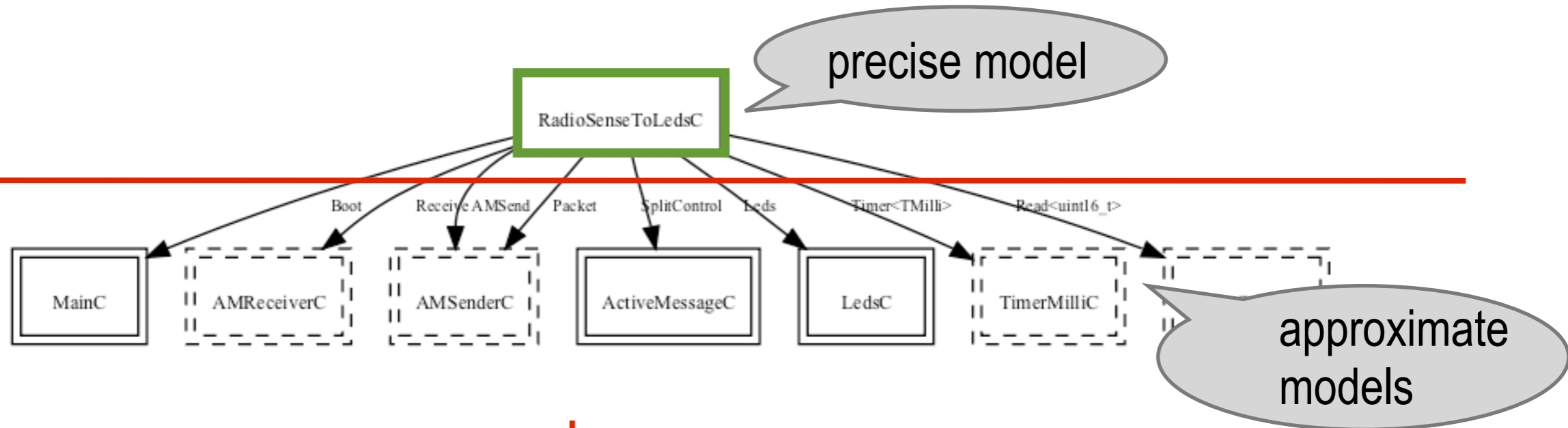
Platform-specific
inlined program
(sequential, C + asm)

Machine code

Deployment
on sensors

model extraction?

Solution 1, high-level



[**Aml'09** -- European Conf. on Ambient Intelligence]

Solution 2, close to hardware

Preserve system-wide code

but

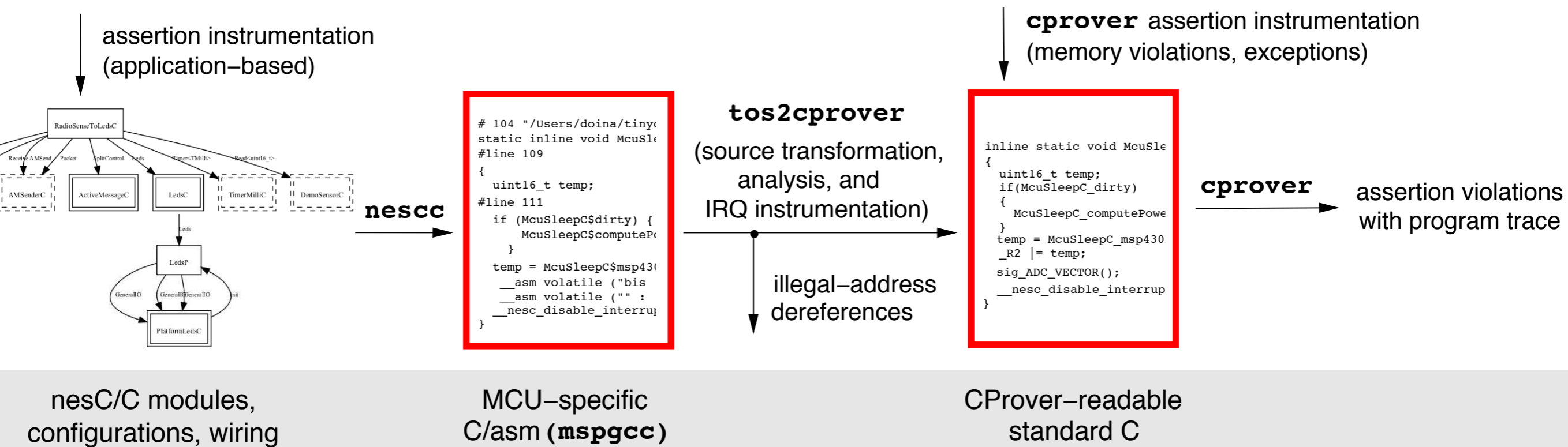
Model the microcontroller's working:

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();
}
```

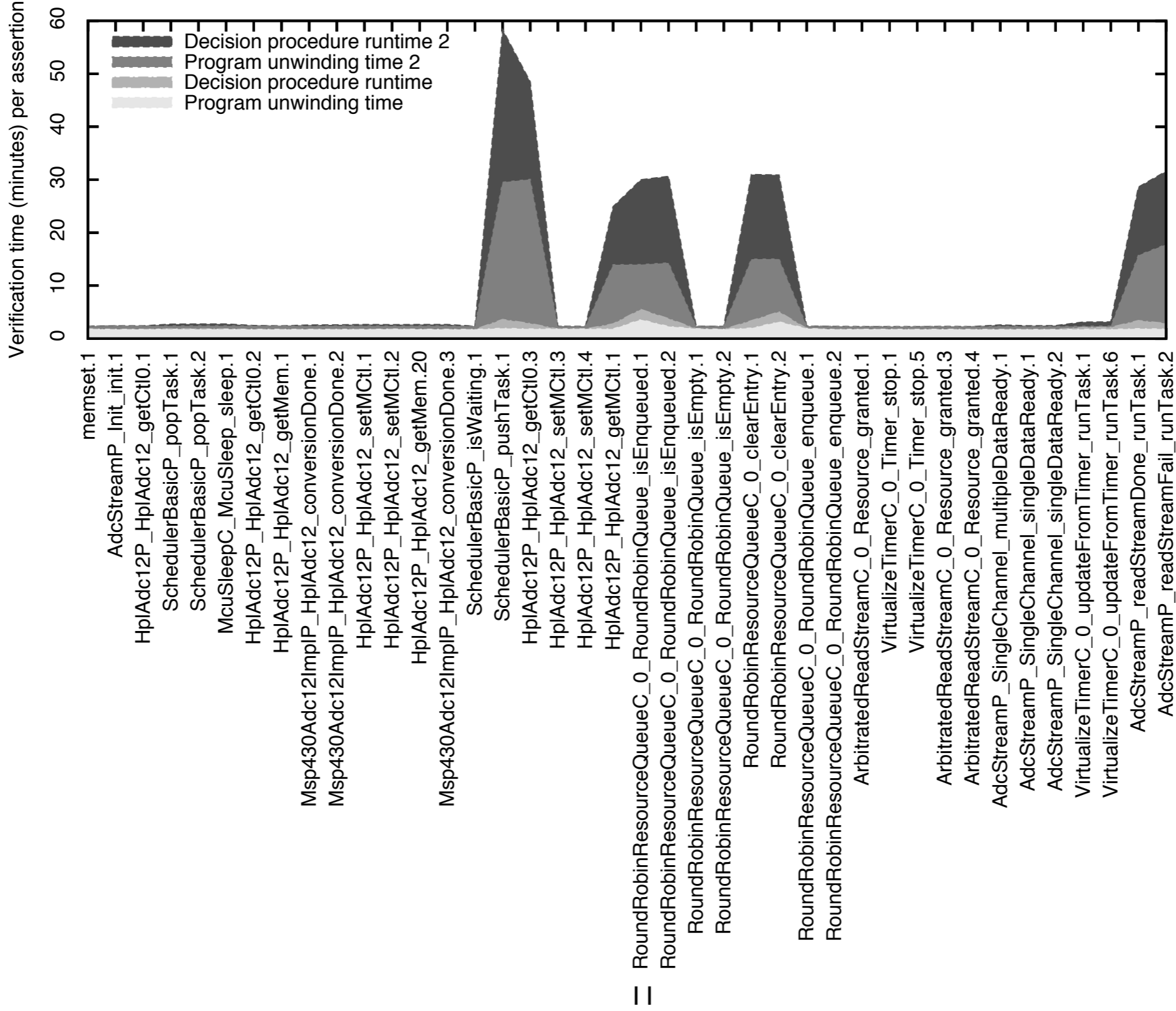
Our toolchain



Problem size

	<i>Blink</i>	<i>Sense</i>	<i>TestDissemination</i>
functionality	timer	sensor, timer	CC2420 radio, timer
lines of code, number of loops	3340, 8	7181, 16	13388, 31
memory-violation assertions	35	132	747
expected interrupts	TIMERB0	TIMERB0, ADC	TIMERB0, PORT1, PORT2, UARTORX, UARTOTX
reachable functions	total: 248, TIMERB0: 114	total: 520, TIMERB0: 185, ADC: 166	total: 1022, TIMERB0: 364, PORT1: 153, PORT2: 25, UARTORX: 268, UARTOTX: 16
potentially raced global variables	TIMERB0: 6	TIMERB0: 7, ADC: 11	TIMERB0: 15, PORT1: 13, PORT2: 0, UARTORX: 19, UARTOTX: 0
IRQ instrumentations	initial 21, minimized to 4	initial 92, minimized to 8	initial 422, minimized to 30

Verification times



Thank You!

Software Verification for TinyOS Sensors

Doina Bucur and Marta Kwiatkowska



Doina Bucur

Ph.D, Dec 2008, University of Aarhus, Denmark, “On Context Awareness in Global Ubiquitous Computing”, with Mogens Nielsen

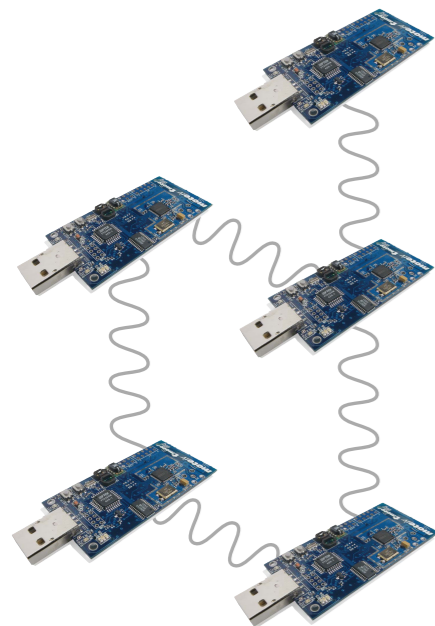
Post-doctoral researcher, Oxford University Computing Laboratory, with Marta Kwiatkowska

Bio

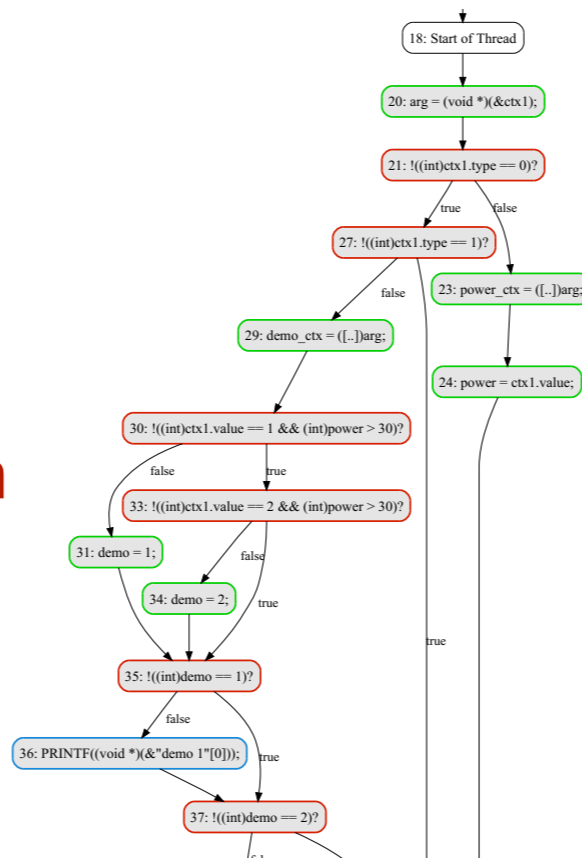
Background in networking, operating systems (teaching for Cisco Systems); ubiquitous computing, especially sensor networks.

Some formal and software verification. Currently working at the intersection of systems and verification.

Mission statement

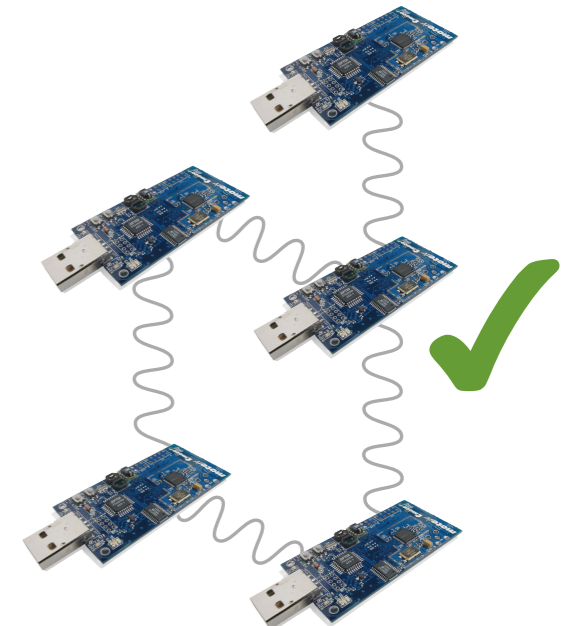


Model extraction



Suitable specification

Verification



Real **sensor** deployments (MSP430 TinyOS nodes)

Program-state assertions and Bounded model checking, CBMC (qualit.)