Software Verification for TinyOS

A tool chain for proving assertions over
MSP430, nescc-generated code at compile-time

Doina Bucur, Marta Kwiatkowska
University of Oxford, UK

In perspective:

Existing debugging techniques:
- runtime safety: SafeTinyOS [1]
  error detection -> reboot
- emulation: runtime safety on cycle-accurate
  hardware emulators. WSim [2]
- simulation and verification
  error detection before deployment

Properties to verify: assert(_P50UT & 0x0008);

Our tool chain:

assertions (application-based)

C program transformation, instrumentation with IRQs

tos2prover [3]

Illegal-address dereferences

cprover assertions

Cprover- readable ANSI C

Program transformation

mspgcc C to ANSI C:

Low-level C accesses
MCU peripherals (left) and
registers directly.

This is emulated in
ANSI C (right).

Instrumentation with
IRQ calls:

A minimum emulation of the MSP430 hardware. For all:

void sig_ADC_VECTOR(void) __attribute__((wakeup))
  __attribute__((interrupt(14)));

we insert guarded calls:

if (int_enabled()) {
  disable_int();
  sig_ADC_VECTOR();
  enable_int();
}

and minimize their occurrence with a partial-order reduction.

References

  and docs at http://www.daimi.au.dk/~doina/wtinyos.php
  http://www.cprover.org/cbmc/