

Show pagesource	Old revisions	Recent changes	<input type="text"/>	Search
-----------------	---------------	----------------	----------------------	--------

Trace: » lab2 » lab3 » lab4 » lab5r » lab6 » lab7 » lab8 » lab9 » lab10 » lab11

## Avrora

### Table of Contents

Avrora
Instalati Avrora
Prima simulare
Simularea unei platforme
Monitorizare si profiling
Monitorizarea energiei
Monitorizarea Intreruperilor
Monitorizarea Interfetei Seriale
Simularea unei retele de senzori

Avrora este o colectie de tool-uri de simulare si analiza pentru programe scrise pentru microcontrollerul AVR si in special pentru nodurile senzoriale Mica2. Avrora contine un framework flexibil pentru simularea si analiza programelor scrise in limbaj de asamblare, punand la dispozitie un **API** usor de folosit, scris in Java si infrastructura necesara pentru experimentarea, profilingul si analiza retelelor senzoriale.

### Instalati Avrora

Pentru a folosi Avrora trebuie sa urmati pasii de mai jos:

1. Instalati-va Java pe masina unde lucrati.
2. Descarcati arhiva jar cu Avrora de [aici](#).
3. Daca nu il aveti deja pe masina voastra, instalati avr-gcc.
4. Avrora poate primi ca intrare fisiere scrise in Atmel assembler (.asm), GNU Assembler (.s) si fisiere convertite cu utilitarul avr-objdump (.od)
5. Avrora ruleaza in linia de comanda. Pentru a invoca Avrora trebuia sa rulati comanda java cu optiunea -jar pentru a specifica locatia fisierului JAR. Rularea fara optiuni va duce la listarea meniului help. Este indicat sa va faceti un alias dupa comanda, in felul urmatoar:

```
% alias avrora='java -jar avrora-beta-1.6.0.jar'
```

### Prima simulare

Pentru inceput, sa luam un program [simplu](#) si sa-l simulam.

Avrora nu contine si un dezasamblor, deci nu poate incarca direct cod masina. Daca programul vostru este scris in orice alt limbaj decat assembler, trebuie sa compilati codul si apoi sa folositi utilitarul avr-objdump pentru fisierul binar pentru a obtine cod scris in assembler care poate fi interpretat de Avrora.

Primul pas este sa compilati programul de mai sus:

```
% avr-gcc -o simple.elf -mmcu=atmega128 simple.c
```

In urma compilarii vi s-a generat un fisier binar ELF. Pe acesta il vom pasa mai departe dezasamblorului, folosind comanda:

```
% avr-objdump -zhD simple.elf > simple.od
```

Optiunile specifica programului sa dezasambleze si zero-urile din fisier (zero este codul operatiei NOP), printeaza informatiile din header si sa dezasambleze fiecare sectiune in parte.

Dupa acest pas ati obtinut fisierul [simple.od](#). Incercati sa-l simulati, rezultatul trebuie sa fie ceva asemanator:

```
% avrora simple.od
Avrora [Beta 1.6.0] - (c) 2003-2005 UCLA Compilers Group

This simulator and analysis tool is provided with absolutely no warranty,
either expressed or implied. It is provided to you with the hope that it be
useful for evaluation of and experimentation with microcontroller and sensor
network programs. For more information about the license that this software is
provided to you under, specify the "license" option.

Loading simple.od...[OK: 0.928 seconds]
=={ Simulation events }=====
Node      Time      Event
=====
=====
Simulated time: 166 cycles
Time for simulation: 0.475 seconds
Total throughput: 3.4947367E-4 mhz
% _
```

Pentru acest exemplu simplu, Avrora va calculeaza numarul de cicli executati, va arata timpul necesar pentru simularea programului si throughput-ul total.

### Simularea unei platforme

Un exemplu ceva mai greu este [blink.od](#). Acesta este un dump al unui program TinyOS care face sa clipeasca led-ul rosu de pe nodul [Mica](#) odata pe secunda. Pentru a simula bine acest exemplu trebuie sa specificam platforma pe care vrem sa rulam codul, pentru a avea toate device-urile hardware conectate.

```
% avrora -platform=mica2 -seconds=5.0 blink.od
```

```

Avrora [Beta 1.6.0] - (c) 2003-2005 UCLA Compilers Group

This simulator and analysis tool is provided with absolutely no warranty,
either expressed or implied. It is provided to you with the hope that it be
useful for evaluation of and experimentation with microcontroller and sensor
network programs. For more information about the license that this software is
provided to you under, specify the "license" option.

Loading blink.od...[OK: 1.082 seconds]
=={ Simulation events }=====
Node      Time      Event
-----
0         318      Red: on
0         320      Yellow: on
0         322      Green: on
0         364      Red: off
0         366      Yellow: off
0         368      Green: off
0        7208139 Red: on
0       14408142 Red: off
0       21608143 Red: on
0       28808142 Red: off
0       36008143 Red: on
=====

Simulated time: 36864000 cycles
Time for simulation: 0.812 seconds
Total throughput: 45.399014 mhz
% _

```

## Monitorizare si profiling

O componenta importanta a Avrora o reprezinta monitorizarea. Cu ajutorul acesteia puteti sa faceti profiling la aplicatia voastra, puteti sa monitorizati numarul si tipul de intreruperi declansate in timpul rularii codului vostru, vizualizati accesele la memorie, apelurile la rutine, modul de lucru al procesorului (cat timp a stat in sleep, idle sau a fost activ) si, cel mai important, puteti sa estimati foarte exact consumul de energie al platformei voastre in timpul rularii unui program.

Monitorizarea se face prin adaugarea optiunii -monitors la linia de comanda, optiunile fiind: profiling, memory, energy, sleep, interrupts si calls.

De exemplu, vrem sa facem profiling la programul simple.od:

```

% avrora -monitors=profile -instr-classes simple.od
Avrora [Beta 1.6.0] - (c) 2003-2005 UCLA Compilers Group

This simulator and analysis tool is provided with absolutely no warranty,
either expressed or implied. It is provided to you with the hope that it be
useful for evaluation of and experimentation with microcontroller and sensor
network programs. For more information about the license that this software is
provided to you under, specify the "license" option.

Loading simple.od...[OK: 0.871 seconds]
=={ Simulation events }=====
Node      Time      Event
-----

Simulated time: 166 cycles
Time for simulation: 0.469 seconds
Total throughput: 3.5394458E-4 mhz
=={ Monitors for node 0 }=====
=={ Profiling results }=====
      Address      Count  Run      Cycles      Cumulative
-----
      0x0000:         1          3 =  1.8072 %
0x0004-0x0088:         0   x34
0x008C-0x00A6:         1  x14      15 =  9.0361 %
0x00A8-0x00AA:         2   x2      10 =  6.0240 %
0x00AC-0x00B0:         3   x3      11 =  6.6265 %
0x00B2-0x00B8:         1   x4         5 =  3.0120 %
      0x00BA:         0
0x00BC-0x00C2:         1   x4         6 =  3.6144 %
      0x00C6:         0
0x00CA-0x00D4:         1   x6         6 =  3.6144 %
0x00D6-0x00E0:         5   x4      46 = 27.7108 %
      0x00E2:         4          8 =  4.8192 %
      0x00E4:         1          1 =  0.6024 %
0x00E6-0x00EC:         0   x4
0x00F0-0x00FC:         5   x6      55 = 33.1325 %
0x00FE-0x0102:         0   x3

=={ Profiling Results by Instruction Type }=====
Instruction      Count      Cycles      Percent
-----
      call:         5         20  12.0481 %
      ret:         5         20  12.0481 %
      ldi:        20         20  12.0481 %
      rjmp:         6         12   7.2289 %
      lds:         5         10   6.0240 %
      sts:         5         10   6.0240 %
      sbiw:         5         10   6.0240 %
      muls:         5         10   6.0240 %

```

```

elmpi:      2      6    3.6144 %
sbrs:      5      6    3.6144 %
eor:       6      6    3.6144 %
out:       6      6    3.6144 %
brne:      4      6    3.6144 %
jmp:       2      6    3.6144 %
movw:      5      5    3.0120 %
stpi:      2      4    2.4096 %
cpi:       4      4    2.4096 %
cpc:       4      4    2.4096 %
break:     1      1    0.6024 %

```

## Monitorizarea energiei

Pentru monitorizarea consumului de energie, folositi fisierul blink.od:

```

% avrora -colors=false -platform=mica2 -monitors=energy -seconds=2 blink.od
Avrora [Beta 1.7.109] - (c) 2003-2007 UCLA Compilers Group

Loading blink.od...OK
=={ Simulation events }=====
Node      Time      Event
-----
-----

Simulated time: 14745600 cycles
=={ Energy consumption results for node 0 }=====
Node lifetime: 14745600 cycles,  2.0 seconds

CPU: 0.02501201918701172 Joule
  Active: 0.024673174908284506 Joule, 8013638 cycles
  Idle: 0.0 Joule, 0 cycles
  ADC Noise Reduction: 0.0 Joule, 0 cycles
  Power Down: 0.0 Joule, 0 cycles
  Power Save: 3.3884427872721355E-4 Joule, 6731962 cycles
  RESERVED 1: 0.0 Joule, 0 cycles
  RESERVED 2: 0.0 Joule, 0 cycles
  Standby: 0.0 Joule, 0 cycles
  Extended Standby: 0.0 Joule, 0 cycles

Yellow: 5.371093750000001E-9 Joule
  off: 0.0 Joule, 14745594 cycles
  on: 5.371093750000001E-9 Joule, 6 cycles

Green: 0.0028081519368489587 Joule
  off: 0.0 Joule, 11608639 cycles
  on: 0.0028081519368489587 Joule, 3136961 cycles

Red: 0.0028082181803385418 Joule
  off: 0.0 Joule, 11608565 cycles
  on: 0.0028082181803385418 Joule, 3137035 cycles

Radio: 0.0 Joule
  Power Off:      : 0.0 Joule, 14745600 cycles

SensorBoard: 0.0042 Joule
  on: : 0.0042 Joule, 14745600 cycles

flash: 1.1999999999999999E-5 Joule
  standby: 1.1999999999999999E-5 Joule, 14745600 cycles
  read: 0.0 Joule, 0 cycles
  write: 0.0 Joule, 0 cycles
  load: 0.0 Joule, 0 cycles

```

Optiunea -seconds opreste simularea dupa numarul specificat de secunde. Pentru a simula si mai bine un nod senzorial, exista optiunea -battery care specifica energia totala a nodului in Jouli. Incercati exemplul anterior impreuna cu comanda -battery=1

```
avrora -colors=false -platform=mica2 -monitors=energy -battery=1 blink.od
```

Cat timp functioneaza nodul cu un singur Joule de energie?

## Monitorizarea Intreruperilor

Avrora poate sa monitorizeze toate intreruperile declansate pe platforma simulata. Completati fisierul `main.c` astfel incat timer0 sa declanseze o intrerupere la overflow cu o perioada de `clk/1024`. In corpul intreruperii faceti sa clipeasca ledul rosu. Dupa compilare si dezasamblare rulati:

```
avrora -platform=mica2 -monitors=interrupts -seconds=2 main.od
```

## Monitorizarea Interfetei Seriale

Avrora poate sa emuleze o conexiune seriala intre platforma simulata si calculatoarul gazda. Prin adaugarea optiunii -serial, simularea asteapta conectarea unui program la portul 2390 de pe localhost apoi ruleaza simularea. Completati fisierul `main.c` astfel incat interfata usart0 sa faca echo la toate caracterele primite. Dupa compilare, rulati:

```
avrora -platform=mica2 -monitors=interrupts,serial main.od
```

Deschideti alta consola si conectati-va prin telnet la portul 2390, apoi scrieti ceva in consola si vedeti daca vi se returneaza caracterele trimise.

```
% telnet 127.0.0.1 2390
```

## Simularea unei retele de senzori

Ca sa simulati o retea de noduri trebuie sa specificati optiunea `-simulation=sensor-network` si sa furnizati fisiere `.od` pentru fiecare nod din retea. Descarcati [CntToRfm.od](#) si [RfmToLeds.od](#) si rulati-le cu optiunile:

```
% avrora -simulation=sensor-network -seconds=5.0 -nodecount=1,1 CntToRfm.od RfmToLeds.od
```

Putem sa adaugam simularii si un monitor pentru pachetele radio transmise si receptionate de catre cele doua noduri:

```
% avrora -simulation=sensor-network -seconds=5.0 -nodecount=1,1 -monitors=packet CntToRfm.od RfmToLeds.od
```

Pentru simularea unei retele de mari dimensiuni puteti specifica amplasamentul nodurilor printr-un fisier ce contine topologia retelei. Salvati exemplul de mai jos ca `grid.top`

```
#OL: 10/29/2004
#simple topology specification
#
#this is a sample topology file
#
#syntax: nodeName x y z
#nodeName: any string without space
#cooridantes: integer values,
#per node one line
#negative cooridantes are welcome
node0 0 0 0
node1 10 0 0
node2 15 0 0
node3 20 0 0
node4 25 0 0
node5 40 0 0
```

Descarcati [net.od](#) si rulati simularea pentru topologia data:

```
% avrora -action=simulate -simulation=sensor-network -stagger-start=500000 -platform=mica2 -nodecount=6 -monitors=energy,ene
```

simple.c.zip

si/lab/lab11.txt · Last modified: 2010/01/06 11:07 by Andrei

Show pagesource

Old revisions

Login

Index

Back to top

Except where otherwise noted, content on this wiki is licensed under the following license: [CC Attribution-Noncommercial-Share Alike 3.0 Unported](#)

