Worldsens : Development and Prototyping Tools for Application Specific Wireless Sensors Networks

Guillaume Chelius Éric Fleury **Antoine Fraboulet**

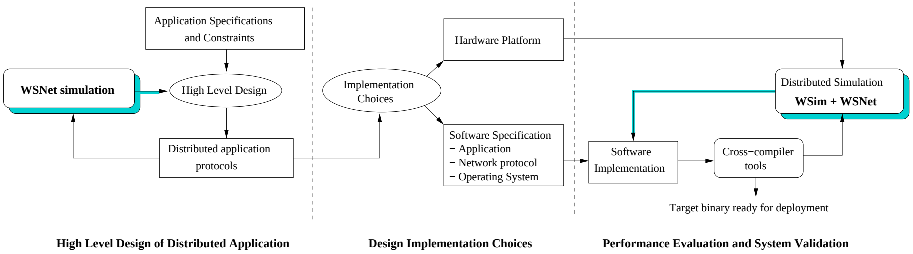
CITI Laboratory

INSA Lyon

INRIA ARES Project

Wasp IST-034963

Worldsens : Application Specific Design Flow


Application Specifications and Constraints
WSNet simulation
High Level Design
Distributed application protocols
Implementation Choices
Hardware Platform
Software Specification
− Application
− Network protocol
− Operating System
Distributed Simulation
WSim + WSNet
Software Implementation
Cross−compiler tools
Target binary ready for deployment
High Level Design of Distributed Application
Design Implementation Choices
Performance Evaluation and System Validation

Worldsens : Design Tools

WSim : hardware platform simulation

- Temporal simulation
- Complete hardware platform simulator
- Peripheral management
- Handles target application binary code

WSNet : wireless network simulation

- Classical event-driven simulator
- Physical layer & radio medium consideration
- UDP/IP frontend

WSNet + WSim : complete distributed system simulation

WSim platform simulator

Software oriented simulator

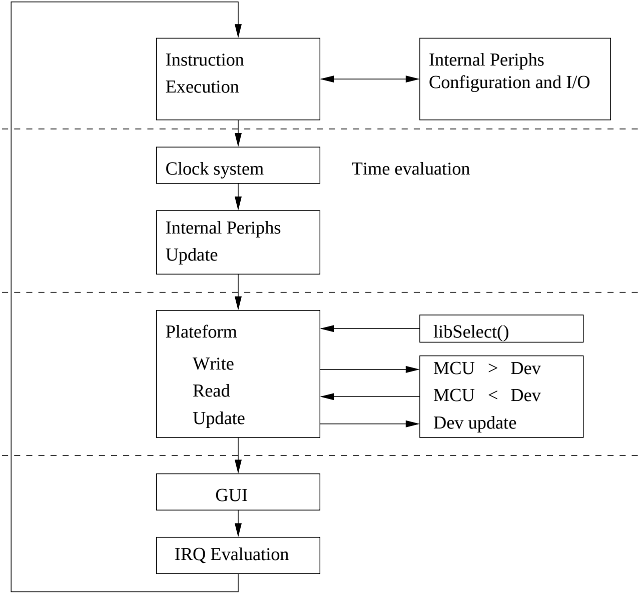
Models characteristics

- Instruction precise (asm)
- Models the complete platform (CPU + peripherals)
- Has a reduced complexity for simpler construction and faster usage.

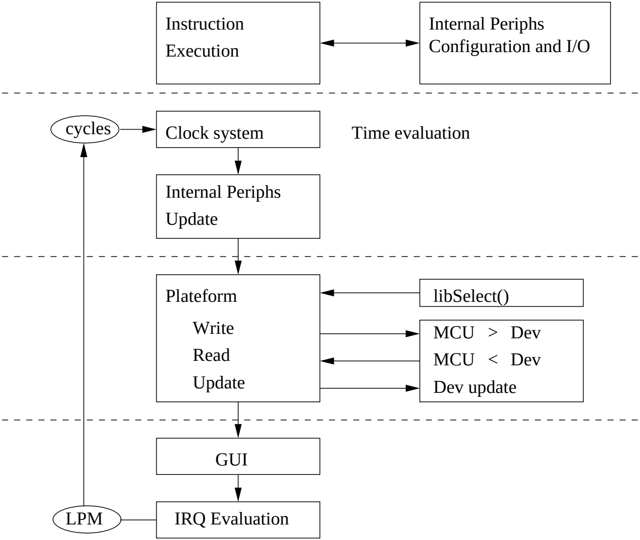
Features

- Can interact with external tools and simulators.
- Fully instrumented for
  - Debug
  - Performance estimation
  - Energy consomption analysis

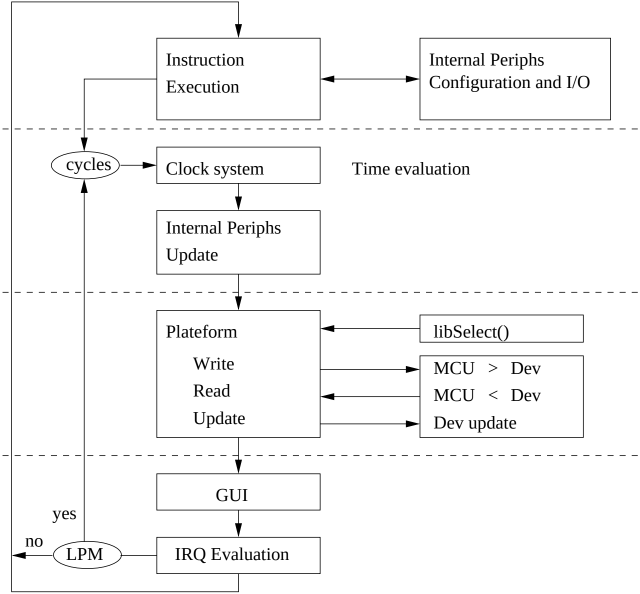
WSim Simulation/Execution Loop



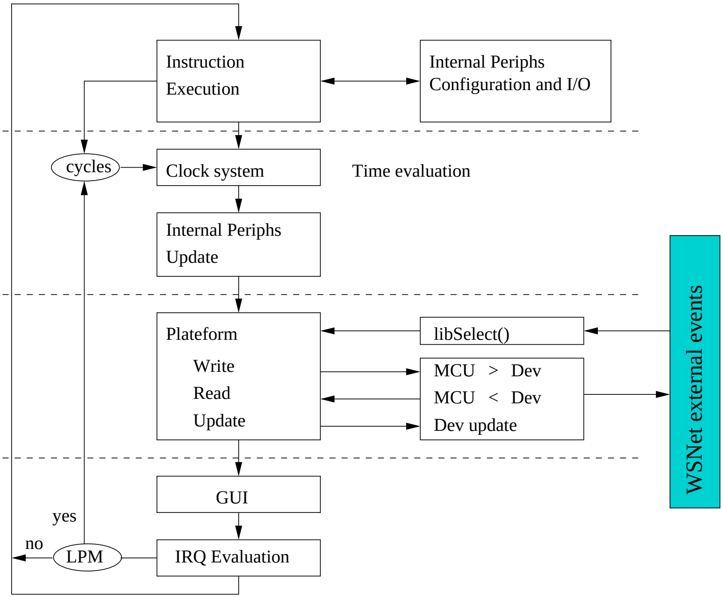
WSim Simulation/Execution Loop



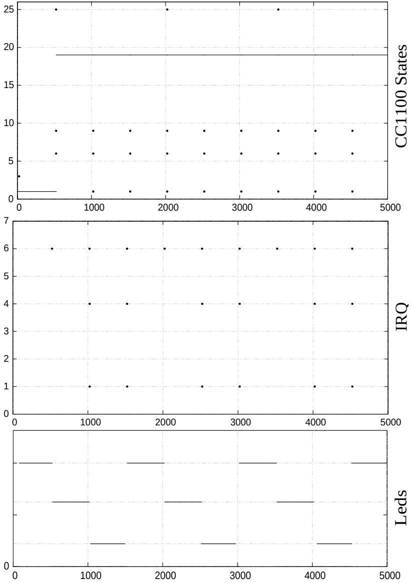
WSim Simulation/Execution Loop



WSim Simulation/Execution Loop



Instruction Precise Simulation



Precise reports

- Interrupts
- Power Modes
- Communications

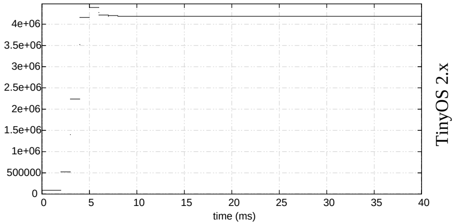
Node activity

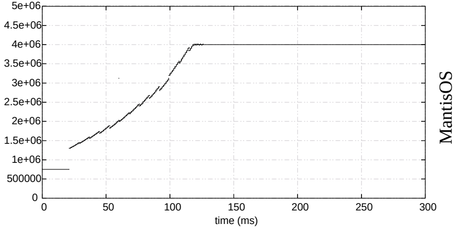
- Simultaneous events
- Off-line analysis

Performance Evaluation

- Code performance
- Memory footprint
- *Power consumption*

Frequency Scaling

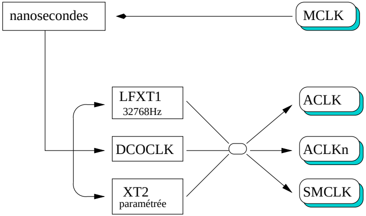




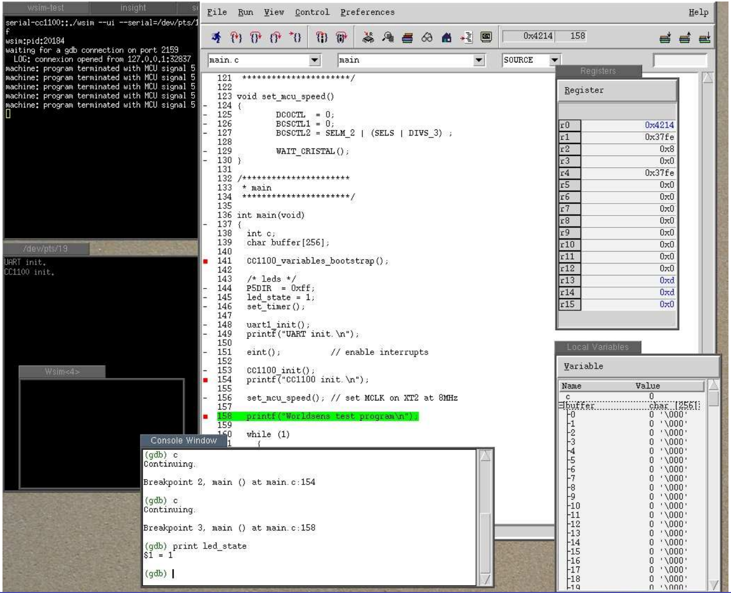
Clock modules simulation

- Variable Frequencies
- Nodes clock skew and drift
- Energy estimation

Clocking System Simulation



Target Code Debug



```
serial-cc1100::./wsim --ui --serial=/dev/pts/1
f
wsim:pid:20184
waiting for a gdb connection on port 2159
  LOG: connexion opened from 127.0.0.1:32837
machine: program terminated with MCU signal 5
machine: program terminated with MCU signal 5
machine: program terminated with MCU signal 5
machine: program terminated with MCU signal 5
machine: program terminated with MCU signal 5
```

```
UART init.
CC1100 init.
```

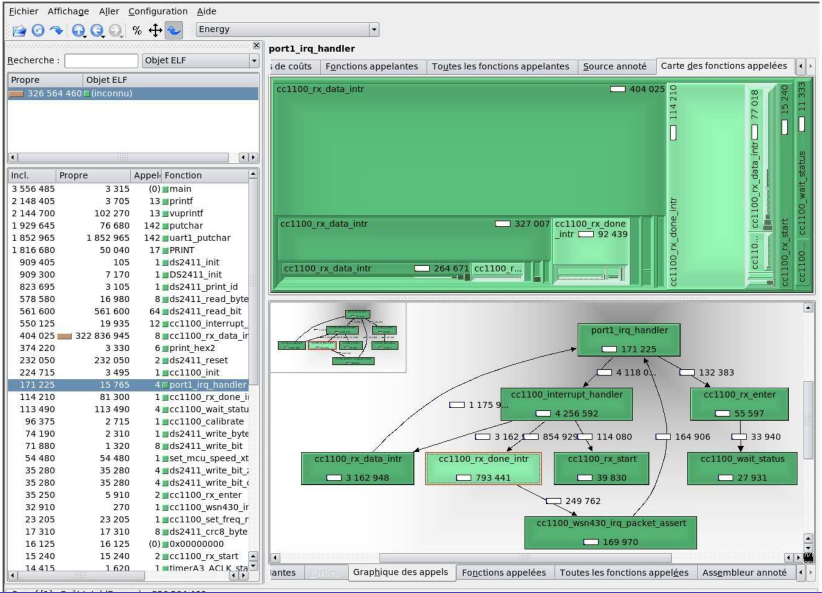
```
121  **********************/
122
123 void set_mcu_speed()
124 {
125         DCOCTL  = 0;
126         BCSCTL1 = 0;
127         BCSCTL2 = SELM_2 | (SELS | DIVS_3) ;
128
129         WAIT_CRISTAL();
130 }
131
132 /**********************
133  * main
134  **********************/
135
136 int main(void)
137 {
138   int c;
139   char buffer[256];
140
141   CC1100_variables_bootstrap();
142
143   /* leds */
144   P5DIR  = 0xff;
145   led_state = 1;
146   set_timer();
147
148   uart1_init();
149   printf("UART init.\n");
150
151   eint();           // enable interrupts
152
153   CC1100_init();
154   printf("CC1100 init.\n");
155
156   set_mcu_speed(); // set MCLK on XT2 at 8MHz
157
158   printf("Worldsens test program\n");
159
160   while (1)
161     {
```

Register	
r0	0x4214
r1	0x37fe
r2	0x8
r3	0x0
r4	0x37fe
r5	0x0
r6	0x0
r7	0x0
r8	0x0
r9	0x0
r10	0x0
r11	0x0
r12	0x0
r13	0xd
r14	0xd
r15	0x0

Name	Value
c	0
⊟buffer	char [256]
0	0 '\000'
1	0 '\000'
2	0 '\000'
3	0 '\000'
4	0 '\000'
5	0 '\000'
6	0 '\000'
7	0 '\000'
8	0 '\000'
9	0 '\000'
10	0 '\000'
11	0 '\000'
12	0 '\000'
13	0 '\000'
14	0 '\000'
15	0 '\000'
16	0 '\000'
17	0 '\000'
18	0 '\000'
19	0 '\000'

```
(gdb) c
Continuing.

Breakpoint 2, main () at main.c:154

(gdb) c
Continuing.

Breakpoint 3, main () at main.c:158

(gdb) print led_state
$1 = 1

(gdb) 
```

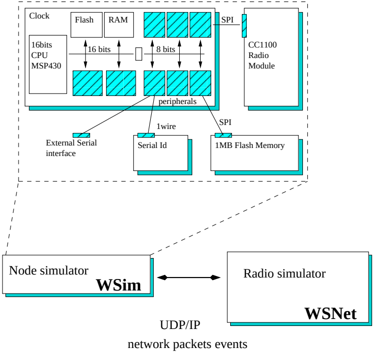
Performance Evaluation : Source Code Annotation



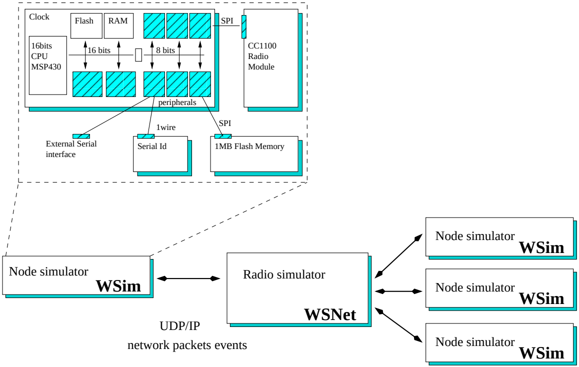
Propre	Objet ELF
326 564 460	(inconnu)

Incl.	Propre	Appel	Fonction
3 556 485	3 315	(0)	main
2 148 405	3 705	13	printf
2 144 700	102 270	13	vuprintf
1 929 645	76 680	142	putchar
1 852 965	1 852 965	142	uart1_putchar
1 816 680	50 040	17	PRINT
909 405	105	1	ds2411_init
909 300	7 170	1	DS2411_init
823 695	3 105	1	ds2411_print_id
578 580	16 980	8	ds2411_read_byte
561 600	561 600	64	ds2411_read_bit
550 125	19 935	12	cc1100_interrupt_
404 025	322 836 945	8	cc1100_rx_data_ir
374 220	3 330	6	print_hex2
232 050	232 050	2	ds2411_reset
224 715	3 495	1	cc1100_init
171 225	15 765	4	port1_irq_handler
114 210	81 300	1	cc1100_rx_done_i
113 490	113 490	4	cc1100_wait_statu
96 375	2 715	1	cc1100_calibrate
74 190	2 310	1	ds2411_write_byte
71 880	1 320	8	ds2411_write_bit
54 480	54 480	1	set_mcu_speed_xt
35 280	35 280	4	ds2411_write_bit_
35 280	35 280	4	ds2411_write_bit_
35 250	5 910	2	cc1100_rx_enter
32 910	270	1	cc1100_wsn430_ir
23 205	23 205	1	cc1100_set_freq_r
17 310	17 310	8	ds2411_crc8_byte
16 125	16 125	(0)	0x00000000
15 240	15 240	2	cc1100_rx_start
14 415	1 620	1	timerA3_ACLK_sta

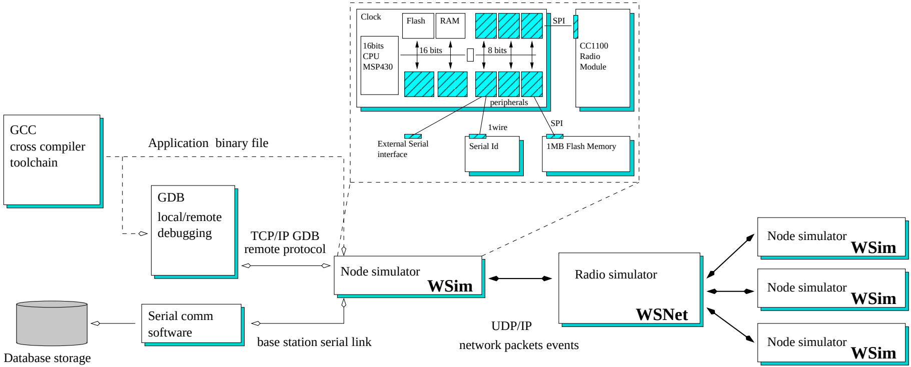
Hybrid simulation



Hybrid simulation



Hybrid simulation



Tutorial website

http ://wsim.gforge.inria.fr/tutorials/wasp/