MansOS un *SEAL*,
jeb
Vai sensoru tīklu programmatūra var būt viegli lietojama?

Atis Elsts
Latvijas Universitāte &
Elektronikas un datorzinātņu institūts

Iepazīšanās

Elektronikas un datorzinātņu institūts (EDI)

- dibināts 1960. gadā
- neatkarīga institūcija
- aktuālās pētījumu tēmas:
  - signālu apstrāde
  - attēlu analīze un apstrāde
  - biometrija
  - kiberfizikālās sistēmas, sensoru tīkli

Kiberfizikālo sistēmu laboratorija:

- dibināta 2012. gada februārī
- 10 darbinieku (divi zinātņu doktori, četri doktoranti)
- 12+ publikāciju šogad

Kontakti un saites:

E-pasts: info@edi.lv

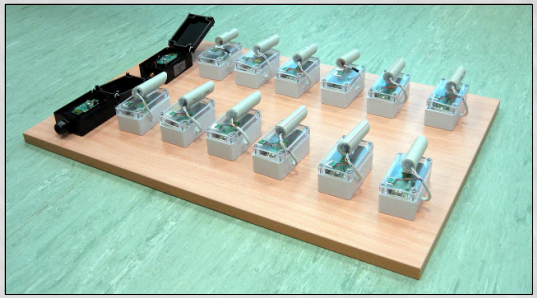
Web: http://www.edi.lv

ELEKTRONIKAS UN DATORZINĀTŅU INSTITŪTS
INSTITUTE OF ELECTRONICS AND COMPUTER SCIENCE

Iepazīšanās

Ar ko mūsu pētniecības grupa nodarbojas?

- **programmatūras** izstrāde bezvadu sensoru tīkliem
- sensoru tīklu **aparatūras** izstrāde & pielāgošana
- sensoru tīklu **pielietojumi** vides novērošanai u.c.
- sensoru tīklu **programmēšanas valodas & vides** izstrāde



Rezultāti

- *SADmote* – sensoru ierīce vides novērošanai
- *CarMote* – sensoru ierīce automašīnām
- *MansOS* – sensoru tīklu operētājsistēma
- *SEAL* – programmēšanas valoda sensoru tīkliem
- *SAD* – sensoru tīklu pielietojums lauksaimniecībā

Kontakti un saites:

E-pasts: atis.elsts@gmail.com

Web: http://mansos.net http://selavo.lv

Prezentācijas plāns

- Ievads (~5%)
- **MansOS (~45%)**
- BST programmēšanas abstrakcijas (~10%)
- Programmēšanas valoda SEAL (~40%)

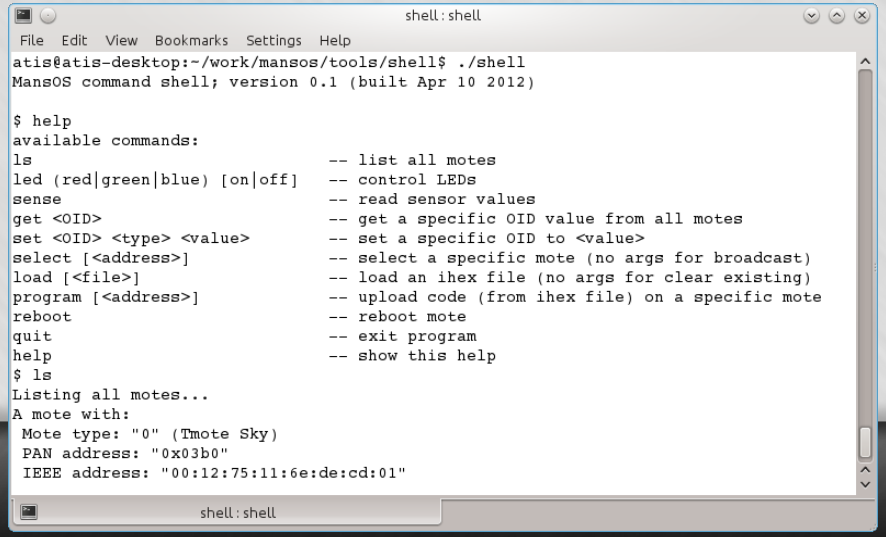
MansOS “filozofija”

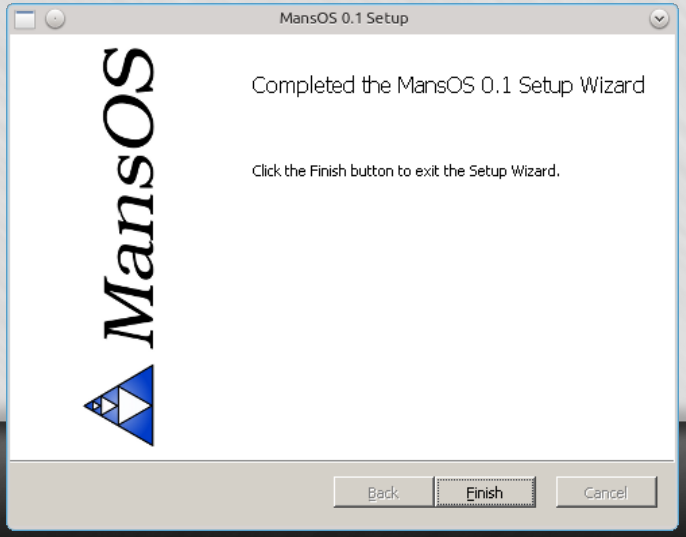
- ***“MansOS: Portable and easy-to-use WSN operating system”***
- **Viegli lietojama**
  - Kam?
  - Salīdzinot ar ko?
- **Portējama**
  - Kāpēc?
  - Kur?
- ***“The underlying principle is that you don't pay for what you don't use”***

(B. Stroustrup)

MansOS “filozofija”

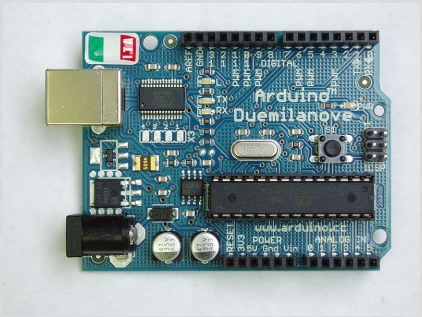
- **Viegli lietojama**
  - Sistēmprogrammētājiem ar UNIX un C pieredzi
  - Piemēram, TinyOS prasa specifiskas valodas (nesC) un programmēšanas paradigmas (event-based execution) apguvi
  - Contiki prasa vismaz *protothreads* abstrakcijas apguvi
  - MansOS pārsvarā var programmēt līdzīgi kā “klasiskas” UNIX sistēmas
  - Uzinstalēt TinyOS un Contiki var būt netriviāli
  - MansOS piedāvā *all included* binārās distribūcijas





MansOS “filozofija”

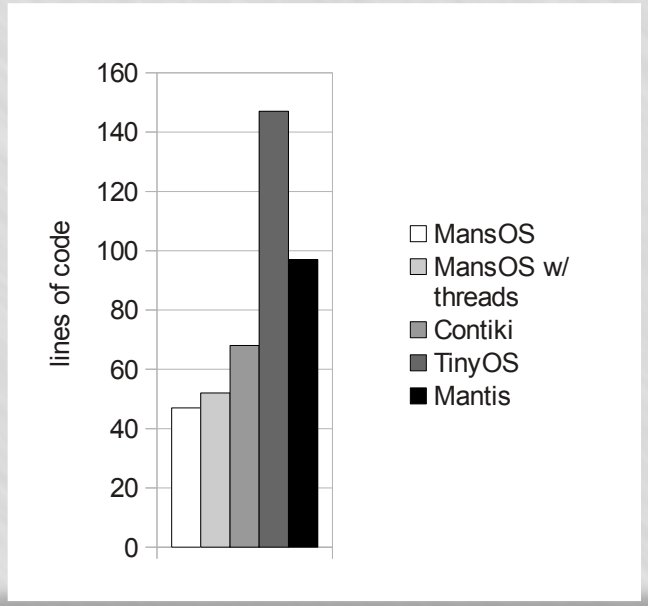
- **Portējamība**
  - Aparatūra bieži ir specifiska (vienam) lietojumam
  - Daudz eksistējošo platformu, maz standartu
  - MansOS darbojas gan uz MSP430, gan uz AVR arhitektūrām
  - T.sk. uz *Tmote Sky*, *Arduino Duemilanove*, *Zolertia* Z1 u.c. platformām



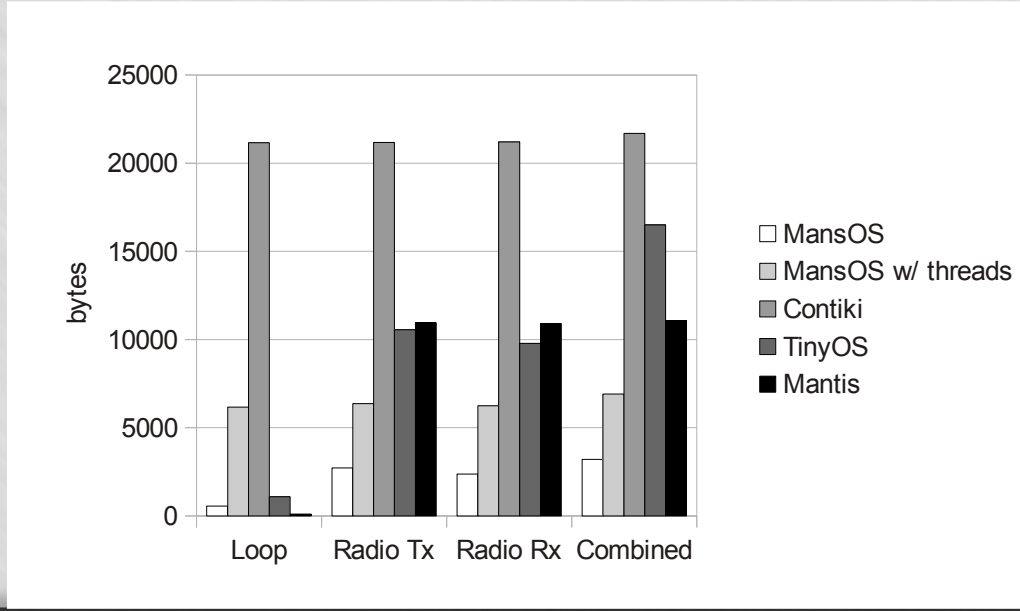
MansOS “filozofija”

- **“Don't pay for what you don't use”**
- MansOS priekšrocības, salīdzinot ar citām BST OS:
  - Īsāks lietotņu pirmkods
  - Mazāks lietotņu binārais kods
  - Vienkārši, bet robusti pavedieni (*threads*)

Lietotnes koda rindiņu skaits



Četru dažādu lietotņu binārā koda izmērs



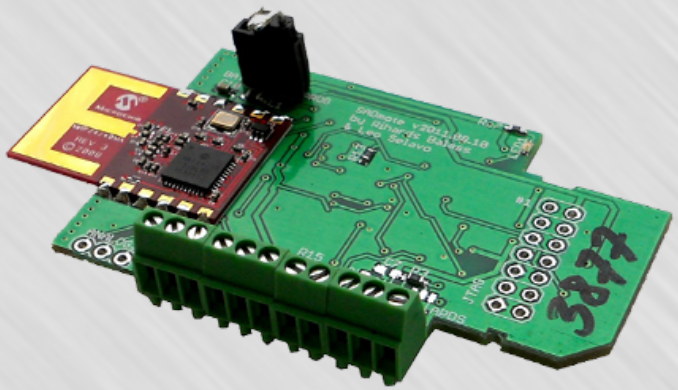
Operētājsistēma MansOS

Dažos teikumos:

- Paredzēta bezvadu sensoru tīkliem un resursierobežotām iegultajām iekārtām
- Mērķauditorija ir programmētāji ar C un UNIX programmēšanas pieredzi
- Veidojuši Leo Seļāvo, Ģirts Strazdiņš u.c., kā arī prezentācijas autors

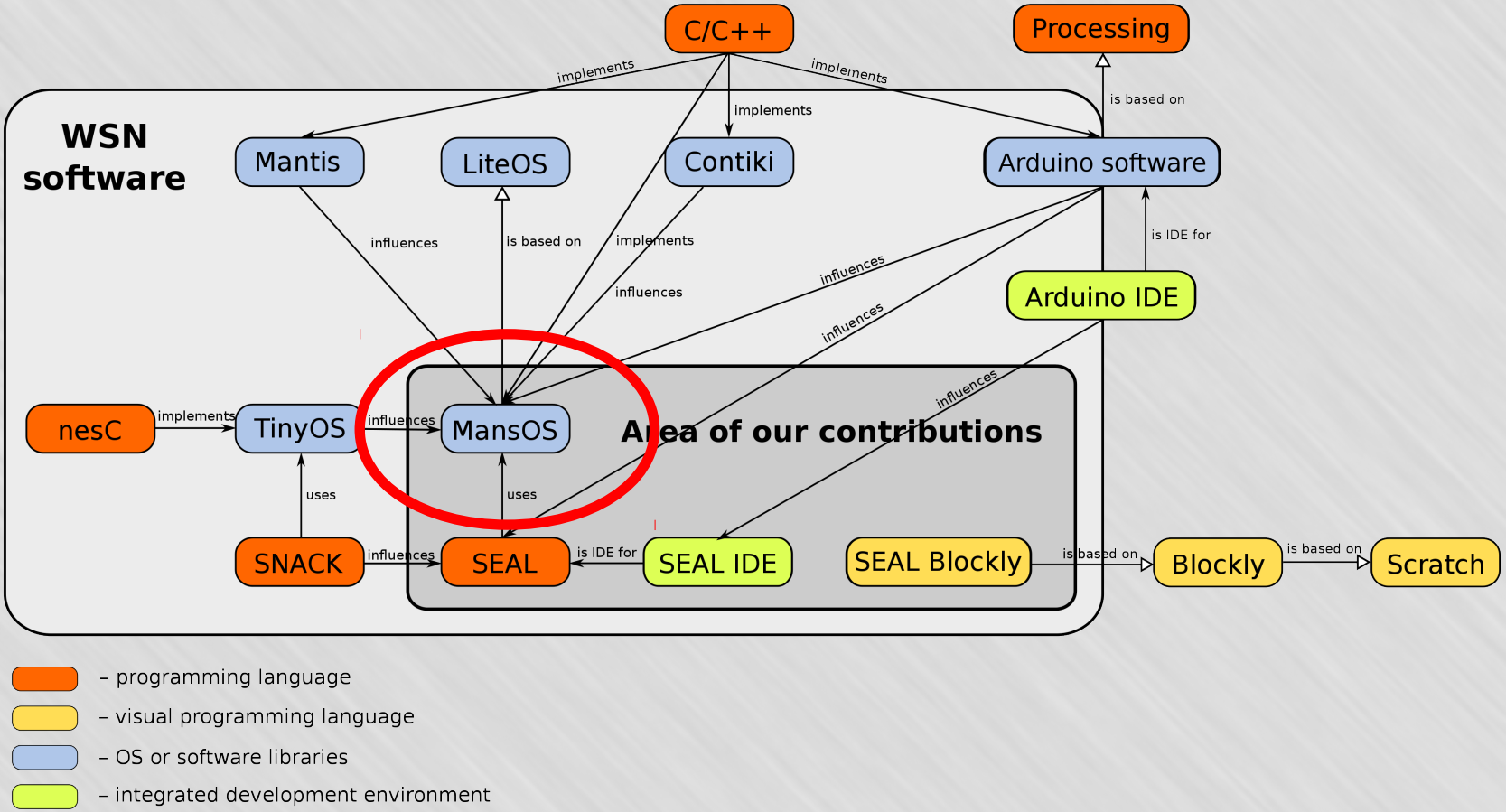
MansOS

http://mansos.net



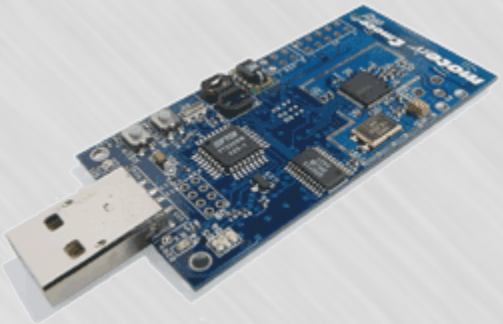
SADmote v02

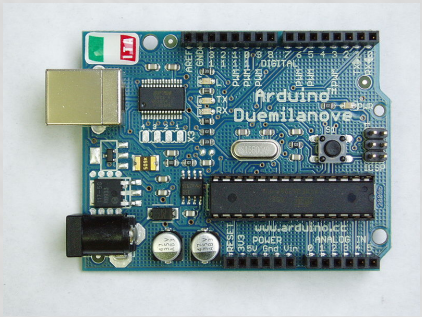
MansOS kontekstā



MansOS skaitļos

- Koda rindiņu skaits (pamatsistēma): 41057
- Pirmkoda failu skaits (pamatsistēma): 273
- Lietotņu paraugu & testa lietotņu skaits: aptuveni 100
- Atbalstītās aparatūras arhitektūras: MSP430, AVR, un PC (x86)
- Atbalstītās aparatūras platformas (šobrīd kopā 10):
- **Commercial**: TelosB (Tmote Sky), Atmega (Arduino), Zolertia Z1, AdvanticSYS XM1000, MSP430FR5739 evaluation board, MSP430 launchpad
- **Custom**: SADmote v2 & v3, FarmMote, TestBed mote, “energy measuring board”





MansOS funkcionalitāte

Analogie un digitāli I/O porti

Digitālie sensoru pieejas protokoli (SPI, I^2C, u.c.)

Zema enerģijas patēriņa režīmi

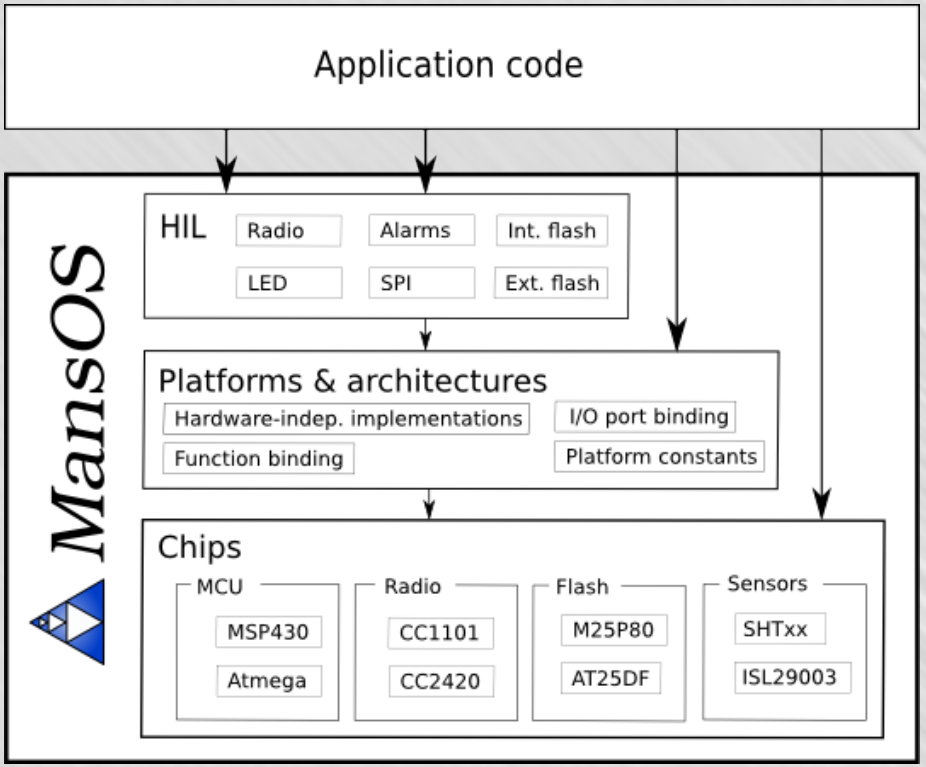
Radio komunikācija un tīkla steks

SD kartes un zibatmiņas atbalsts

Opcionāls IPv6 atbalsts

Run-time management

Run-time reprogramming



MansOS dokumentācija

http://mansos.net – galvenā lapa

http://mansos.googlecode.com - kods

http://selavo.lv/wiki/index.php/MansOS (vietām novecojusi!)

Publikācija *"Design and Implementation of MansOS: a Wireless Sensor Network Operating System"* no LU Zinātniskajiem rakstiem:

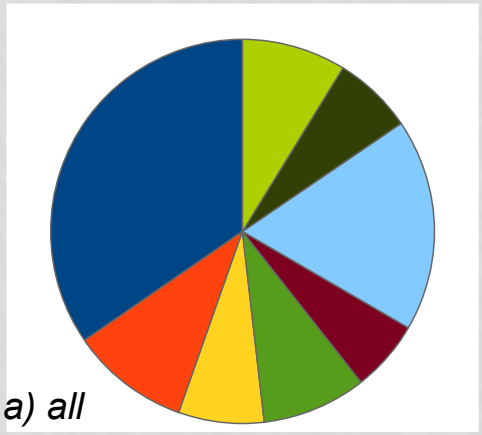
http://mansos.edi.lv/wp-content/uploads/2012/10/mansos-lu-2012.pdf

SEAL dokumentācija:

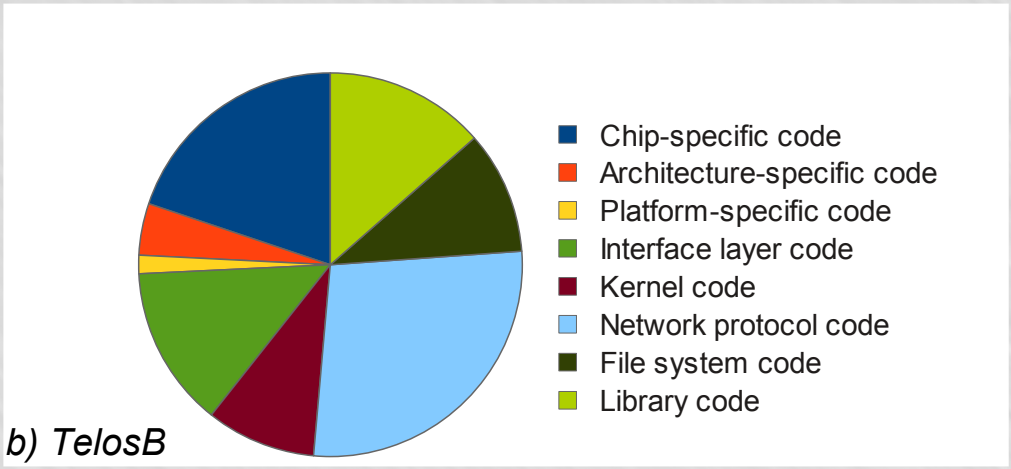
http://selavo.lv/wiki/index.php/SEAL

MansOS portējamība

Komponentu proporcijas:

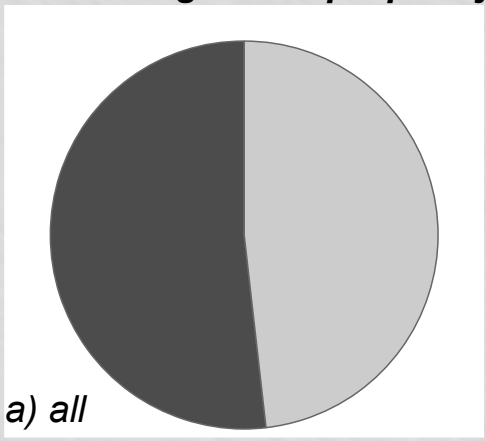


a) all

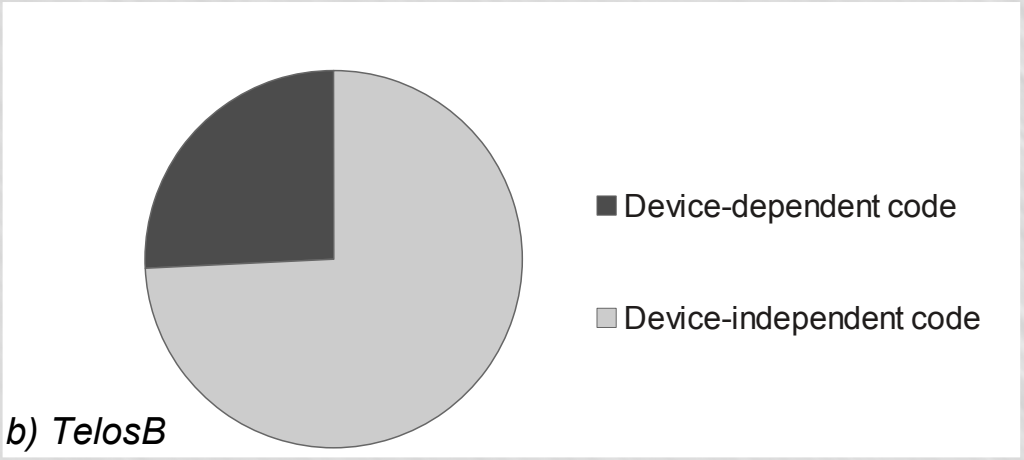


b) TelosB

Aparatūrneatkarīgā koda proporcijas:



a) all



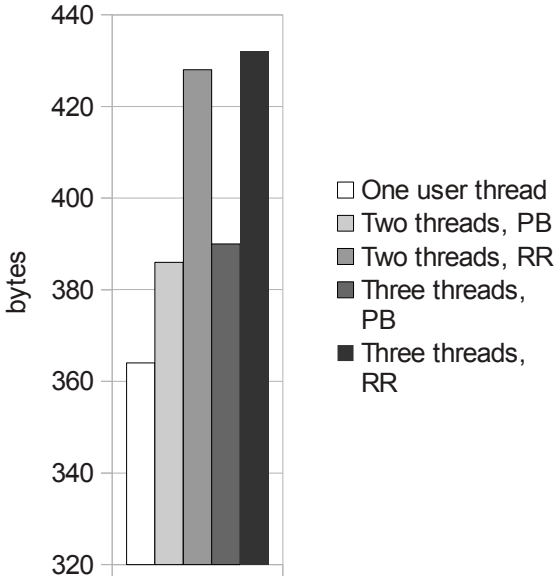
b) TelosB

MansOS pirmkoda sadalījums

MansOS pavedieni

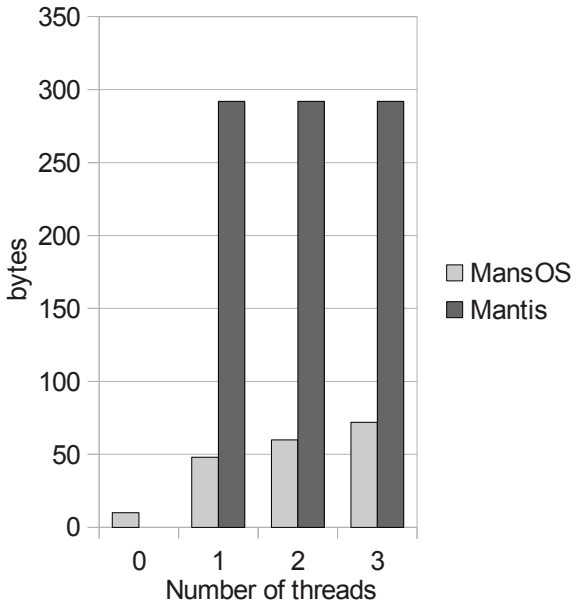
- Kāpēc pavedieni?
  - Vieglāk programmēt!
  - Koda izkārtojums labāk atspoguļo programmas izpildīšanos
  - Var taisīt bezgalīgos ciklus, bet sistēma turpinās strādāt...
- Pamatidejas:
  - Pavedienu ir maz (bieži pietiek ar diviem)
  - Pavedieni ir savstarpēji “draudzīgi”
- Risinājums: ***kooperatīvi pre-emptīvais daudzpavedienu režīms***
- Paviedieni tiek palaisti “pa riņķi” (*round robin scheduling*)
- Kodola pavedienam vienmēr priekšroka
- `sleep()` ieliek sistēmu zema enerģijas patēriņa režīmā ***tikai*** tad, ja citi pavedieni neaktīvi!

MansOS pavedieni

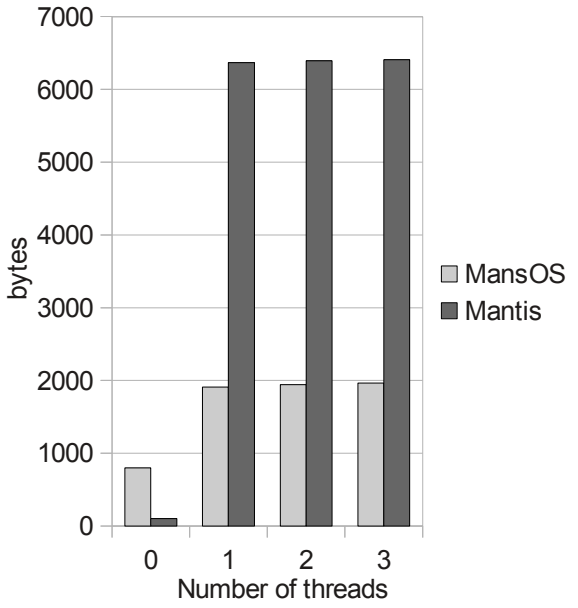


`schedule()` funkcijas binārā koda izmērs atkarībā no pavedienu skaita

(RR – round robin, PB – priority-based scheduling)



Pavedienu moduļa RAM lietojuma salīdzinājums



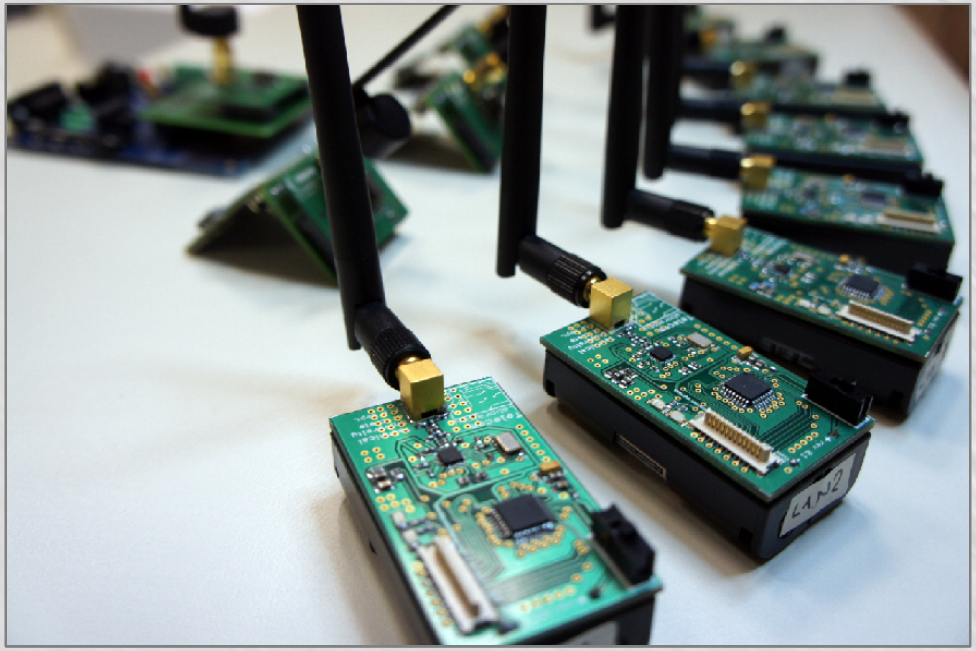
Pavedienu moduļa koda atmiņas lietojuma salīdzinājums

Prezentācijas plāns

- Ievads (~5%)
- MansOS (~45%)
- **BST programmēšanas abstrakcijas (~10%)**
- Programmēšanas valoda SEAL (~40%)

Pieejas BST programmēšanai

- Node-local programming
- Regional programming
- Macroprogramming
- Query systems



© http://www.acsu.buffalo.edu/~pmarkopo/projects.html

Node-local programming

1) Zema līmeņa valodas: *C, nesC*

2) OS komponentu līmeņa valodas: *TinyScript, SNACK* u.c

***TinyScript* koda piemērs:**

```
buffer data;
private tmp;
private average;
private index;
private i;
tmp = int(photoactive) - int(totalsolar);
if tmp < 0 then
   tmp = -tmp;
end if
buffer[index] = tmp;
index = (index + 1) % 10;
average = 0;
for i = 0 to 10
    average = average + buffer[i] / 10;
next i
uart(average);
```

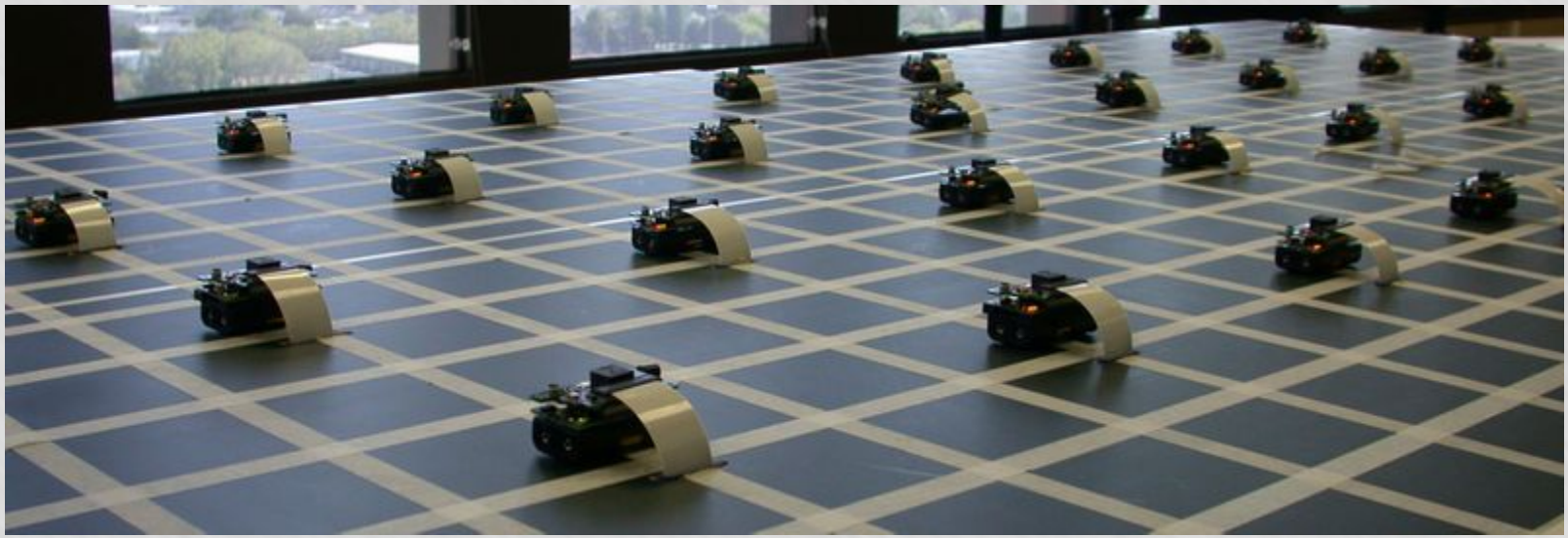
Regional programming

- ***Logical Neighborhoods*** **abstrakcija, SPIDEY valoda**
- **SPIDEY koda piemērs:**

```
neighborhood template HighTempSens(threshold)
  with Function = "sensor" and
       Type = "temperature" and
       Reading > threshold

create neighborhood hts100
  from HighTempSens(threshold : 100)
  max hops 2
  credits 30
```

Macroprogramming

- **“Programming whole network at once”**
- **Valodas & abstrakcijas: *Pleiades, Kairos*, u.c.**



© http://blog.memsic.com/2009/04/testbed-testing.html

```
1  #include "Pleiades.h"
2  boolean nodelocal isFree=TRUE;
3  nodeset nodelocal neighbors;
4  node nodelocal neighborIter;
5
6  void reserve (pos dst) {
7    boolean reserved = FALSE;
8    node nodeIter, reservedNode = NULL;
9    node n=closest_node(dst);
10   nodeset loose nToExamine = add_node(n, empty_nodeset());
11   nodeset loose nExamined = empty_nodeset();
12
13   if (isfree@n) {
14     reserved = TREE; reservedNode = n;
15     isfree@n = FALSE;
16     return;
17   }
18
19   while (!reserved && !empty(nToExamine)) {
20     cfor (nodeIter=get_first(nToExamine);
21           nodeIter!=NULL;
22           nodeIter = get_next(nToExamine)) {
23       neighbors@nodeIter=get:neighbors(nodeIter);
24       for (neighborIter@nodeIter=get_first(enighors@nodeIter);
25            neighborIter@nodeIter!=NULL;
26            neighborIter@nodeIter=get_next(neighbors@nodeIter)) {
27         if (!member(neighborIter@nodeIter,nExamined))
28           add_node(neighborIter@nodeIter,nToExamine);
29       }
30       if (isfree@nodeIter) {
31         if (!reserved) {
32           reserved=TRUE; reservedNode=nodeIter;
33           isfree@nodeIter=FALSE;
34           break;
35         }
36       }
37       remove_node(nodeIter,nToExamine);
38       add_node(nodeIter,nExamined);
39     }
40   }
41 }
```

Fig. 22. A street-parking application in Pleiades.

Query languages

- **Uztver sensoru tīklu kā datubāzi**
- **Valodas un abstrakcijas: TinyDB, SwissQM, u.c.**
- **TinyDB koda piemēri:**

```
SELECT nodeid, light, temp
  FROM sensors
  SAMPLE PERIOD 1s FOR 10s
```

```
SELECT AVG(volume),room FROM sensors
  WHERE floor = 6
  GROUP BY room
  HAVING AVG(volume) > threshold
  SAMPLE PERIOD 30s
```

```
SELECT nodeid, accel
  FROM sensors
  LIFETIME 30 days
```

Prezentācijas plāns

- Ievads (~5%)
- MansOS (~45%)
- BST programmēšanas abstrakcijas (~10%)
- **Programmēšanas valoda SEAL (~40%)**

Valoda SEAL

SEAL ir domēnspecifiska valoda lietojumprogrammu izstrādei

- “domēns” šeit ir BST

SEAL ir balstīta uz MansOS

SEAL ir komplektēta ar vizuālo programmēšanas vidi (IDE)

NesC kods (fragments):

```
#include "Timer.h"
#include "RadioSenseToLeds.h"

module RadioSenseToLedsC @safe(){
  uses {
    interface Leds;
    interface Boot;
    interface Receive;
    interface AMSend;
    interface Timer<TMilli> as MilliTimer0;
    interface Timer<TMilli> as MilliTimer1;
    interface Packet;
    interface Read<uint16_t>;
    interface SplitControl as RadioControl;
  }
}
implementation {

  message_t packet;
  bool locked = FALSE;

  event void Boot.booted() {
    call RadioControl.start();
  }

  event void RadioControl.startDone(error_t err) {
    if (err == SUCCESS) {
      call MilliTimer0.startPeriodic(2000);
      call MilliTimer1.startPeriodic(6000);
    }
  }
  event void RadioControl.stopDone(error_t err) {}

  event void MilliTimer0.fired() {
  }

  event void MilliTimer1.fired() {
  }

  event void Read.readDone(error_t result, uint16_t data) {
    if (locked) {
      return;
    }
    else {
      radio_sense_msg_t* rsm;
```

C kods:

```
emacs23@desktop <2>
File  Edit  Options  Buffers  Tools  C++  Help
#include "stdmansos.h"
#include <hil/alarm.h>
#include <string.h>

#define SENSOR_READ_INTERVAL 3000 // milliseconds

// our timer
Alarm_t timer;

//-----------------------------------------
//      Timer callback
//-----------------------------------------
void onTimer(void *param)
{
    // read light sensor value
    uint16_t light = readLight();
    PRINTF("light = %u\n", light);

    // send the value read to radio
    radioSend(&light, sizeof(light));

    // blink LED
    toggleRedLed();
}

//-----------------------------------------
//      Entry point for the application
//-----------------------------------------
void appMain(void)
{
    // initialize and schedule the alarm
    alarmInitAndRegister(&timer, onTimer, SENSOR_READ_INTERVAL, tru
    // set radio packet receive callback
    radioSetReceiveHandle(onRadioInterrupt);
    // turn radio listening on
    radioOn();
    // our job here is complete
    return;
}
```

SEAL kods:

```
emacs23@desktop <2>
File  Edit  Options  Buffers  Tools  C++  Help
read Light, period 2s;
read Humidity, period 2s;
output Radio;
```

Datu plūsma BST motē

- Dati tiek **ievākti** no sensoriem un *pseidosensoriem*
- Programmas loģika datus **apstrādā**
- Dati tiek **izvadīti** uz tīklu, SD karti, seriālo interfeisu u.c.

SEAL mērķauditorija - 1

Zinātnieki un praktiķi:

- fiziķis, kas vēlas izveidot vidi antenu testēšanai
- vides pētniece, kas pēta mikroklimata ietekmi uz augiem
- “hakeris”, kas grib paaugstināt savas mājas energoefektivitāti

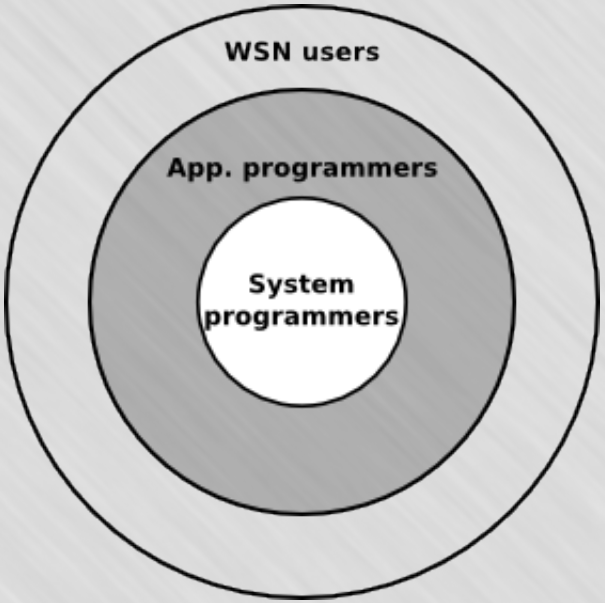
Viņu programmēšanas zināšanas ir:

- iegūtas pašmācības ceļā, ja vispār:
  - bioloģijas, ķīmijas, ekonomikas programmās LU nav programmēšanas kursu!
  - fizikas programmā programmēšana ir kā izvēles kursi
- viņiem nav pieredzes ar lieliem projektiem
- viņi nevēlas tērēt daudz laika programmēšanas vides un valodas apguvei

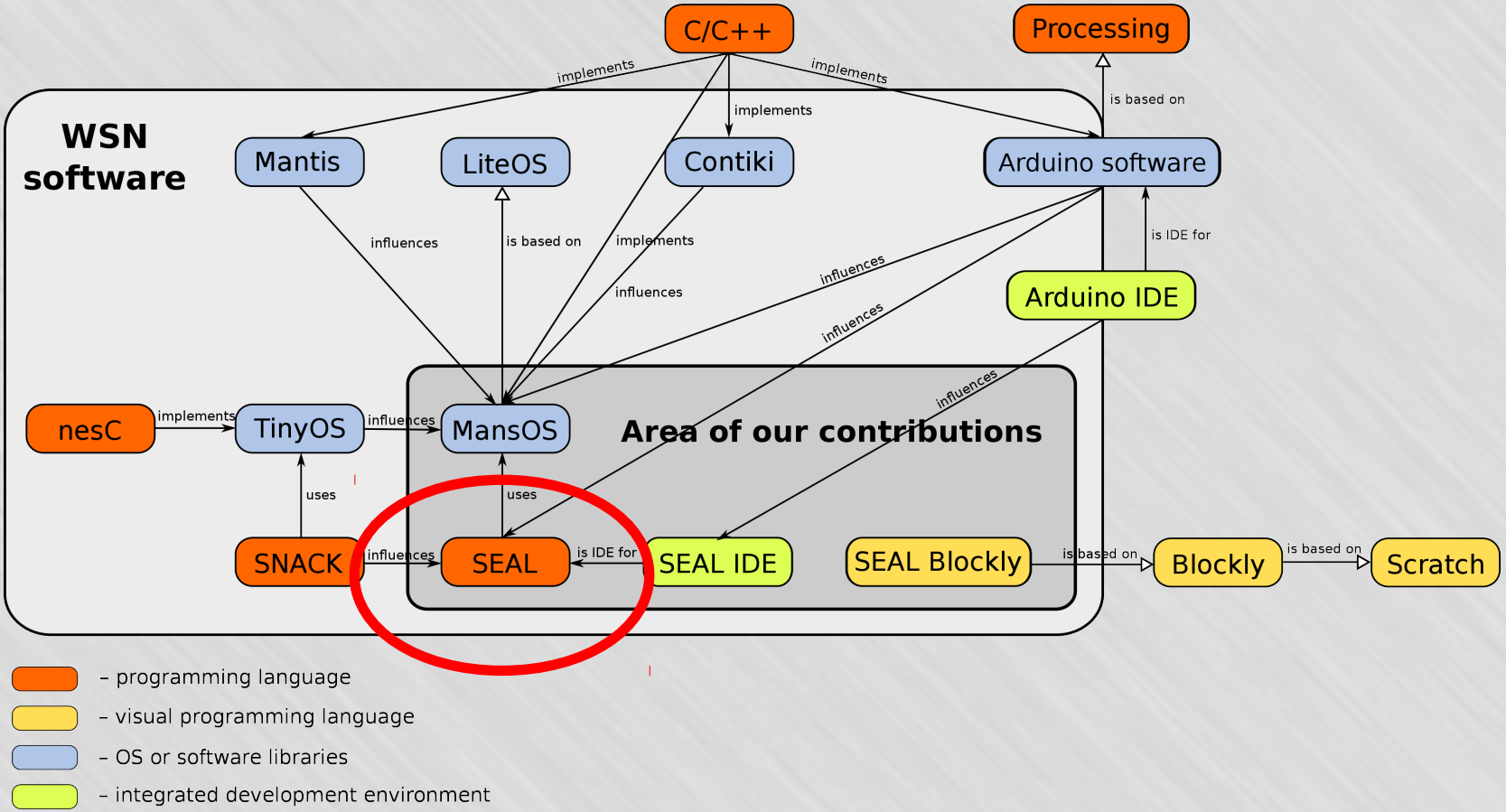
SEAL mērķauditorija - 2

Bezvadu sensoru tīklu pētnieki, kas vēlas ātri prototipēt lietojumprogrammas:

- piemēram, elektroniķi un aparatūras izstrādātāji
- sistēmas programmatūras izstrādātāji, kas izmanto SEAL programmas kā testu scenārijus



SEAL kontekstā



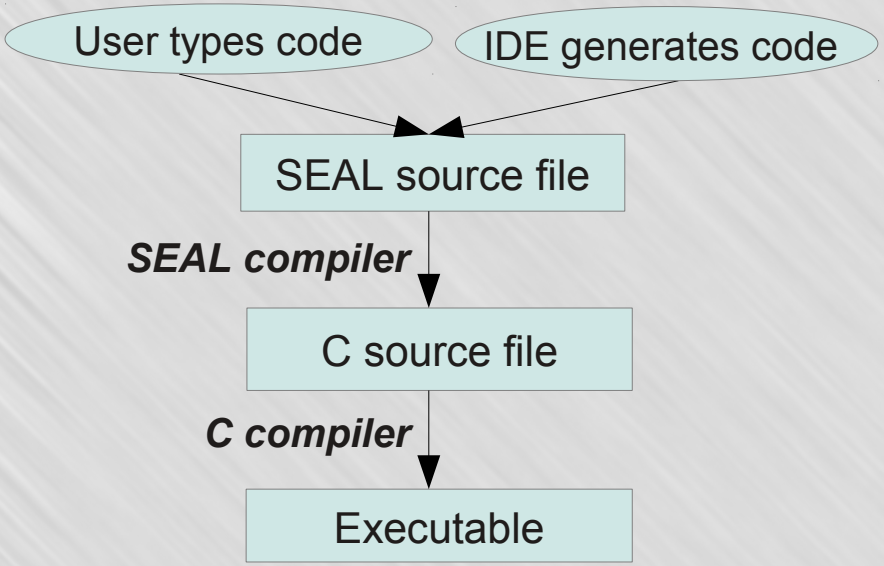
SEAL programmas piemērs

Ko dara šī programma?

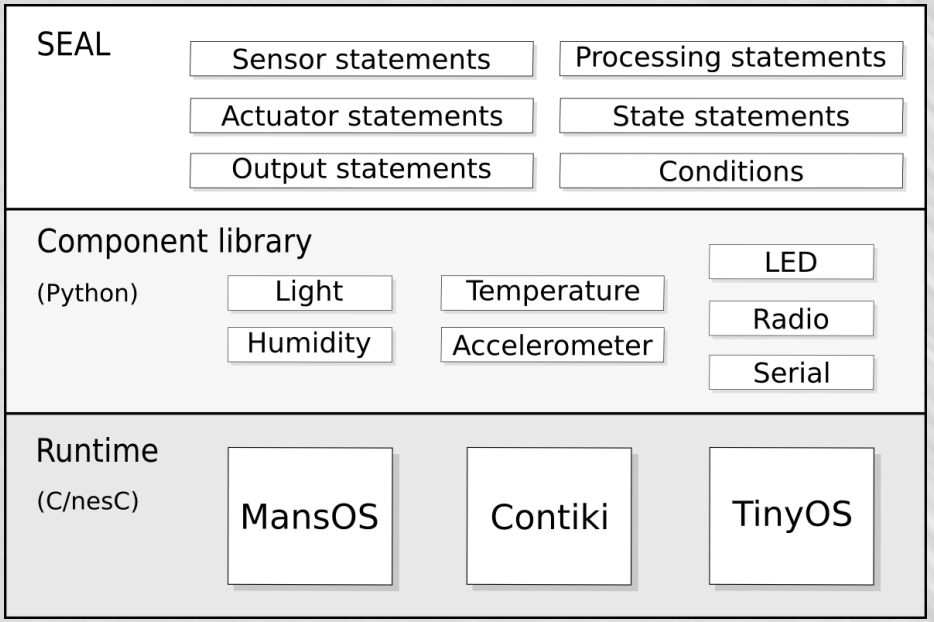
```
read Temperature, period 10s;
output Network;
when Temperature > 40C:
    use RedLed, on;
end
```

SEAL arhitektūra

- SEAL – **Se**nsor **A**pplication Development **L**anguage
- Deklaratīva ar dažiem imperatīviem elementiem
- Kompakta un dabīgi lasāma sintakse
- Divi līmeņi: sintakse & komponentu bibliotēka, kā arī “runtime support” no OS)



SEAL lietotņu izstrādes process



SEAL konceptuālā arhitektūra

SEAL elementi

Komponentu lietojumi:

```
use RedLed;
read Humidity;
output Radio;
```

Komponentu definīcijas:

```
define MaxLight max(Light);
```

Nosacījuma izteikumi:

```
when MaxLight > 1000: ...
```

Konstanšu un mainīgo definīcijas:

```
const INTERVAL 10s;
set IsLedOff True;
```

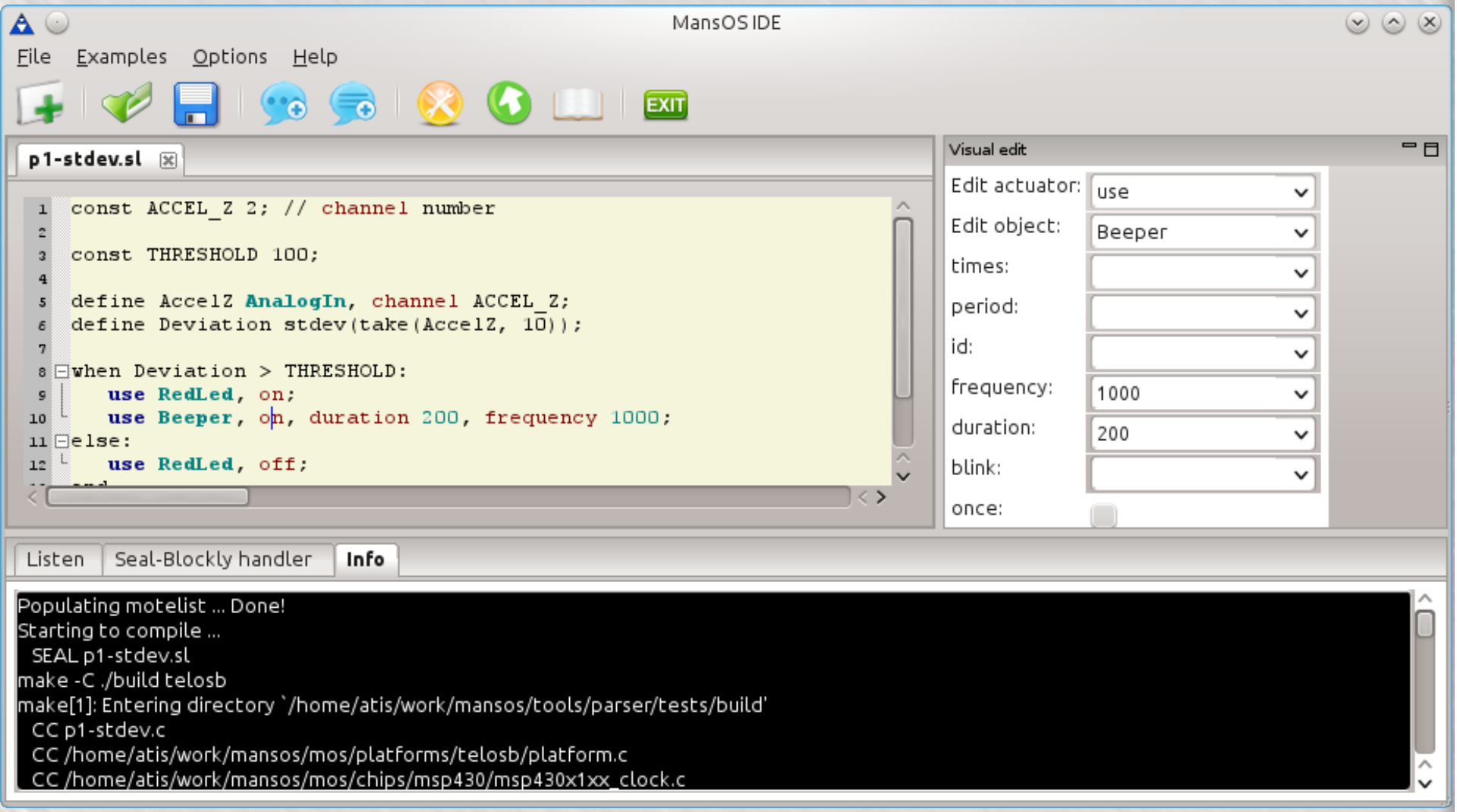
SEAL pamatizvēles

- **Domain-specific** *vs* library *vs* embedded domain specific
  - More intuitive & faster to learn to the users
- **Declarative** *vs* imperative
  - Specify only *what* to do, not *how* to do it
- **Natural language-like** syntax *vs* mathematical notation *vs* ...
  - Reading & understanding should be possible without special skills
- **Node-local** *vs* macroprogramming *vs* regional
  - Concrete concepts like “node” and “sensor” have closer mapping to physical world objects compared to “an abstract region”, “the network”, etc.

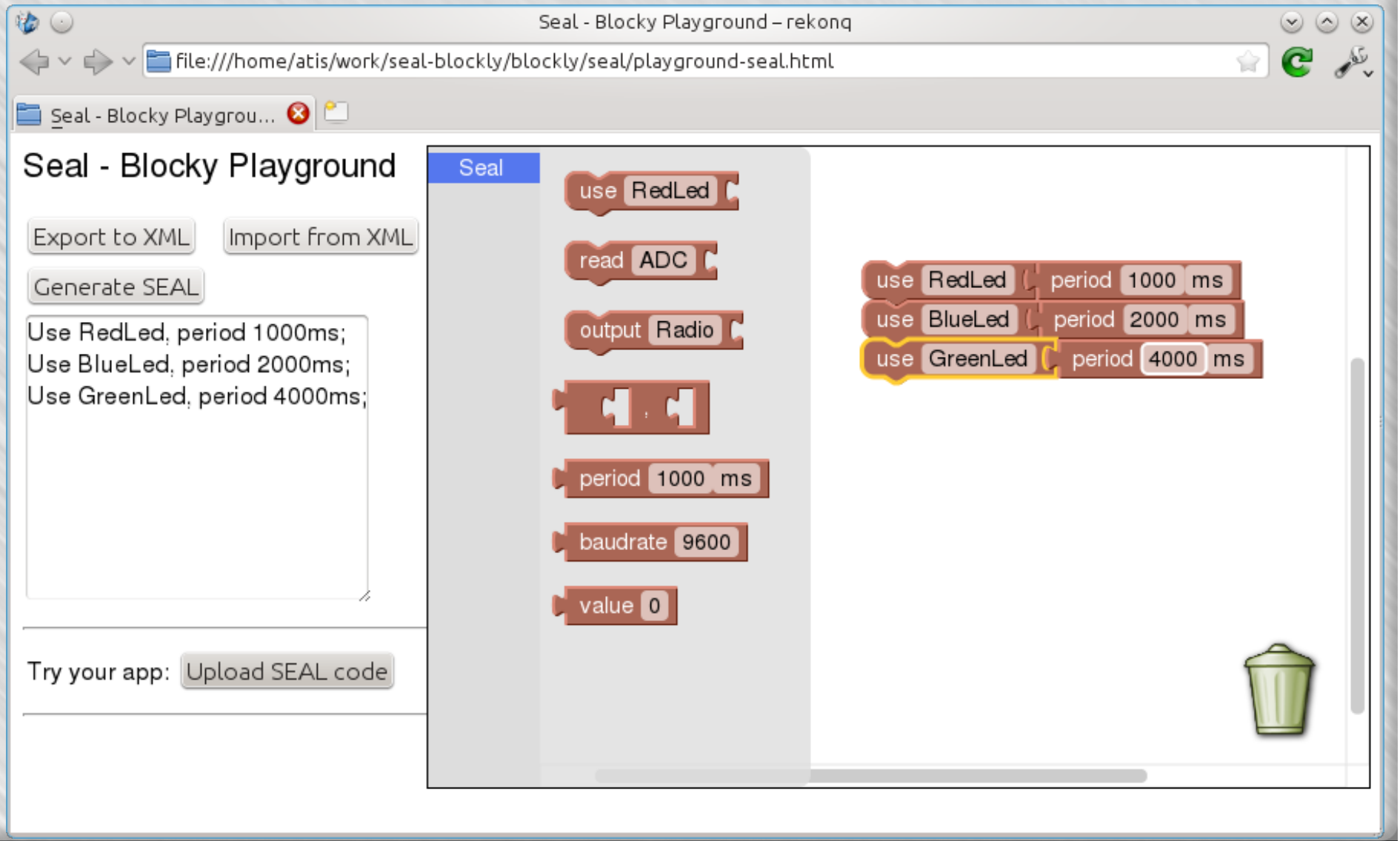
SEAL implementācija

- Kompilators implementēts Python
- Aptuveni 7000 koda rindiņas
- Parseris: Python-Lex-Yacc (PLY)
- Koda ģenerators: generē MansOS C kodu
- Viss SEAL pirmkods pieejams zem atvērtā koda licences (*NewBSD license*), kā daļa no MansOS distributīva
- Iekļauti 115 koda piemēri (pus)automātiskai SEAL testēšanai

SEAL/MansOS grafiskā vide

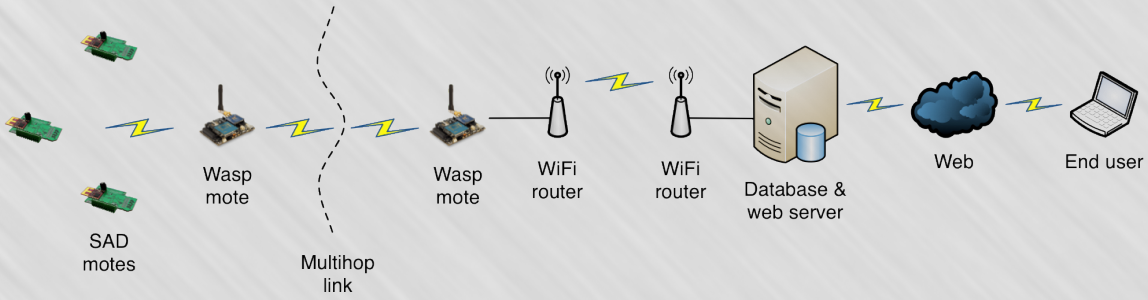


SEAL programmēšana no Web



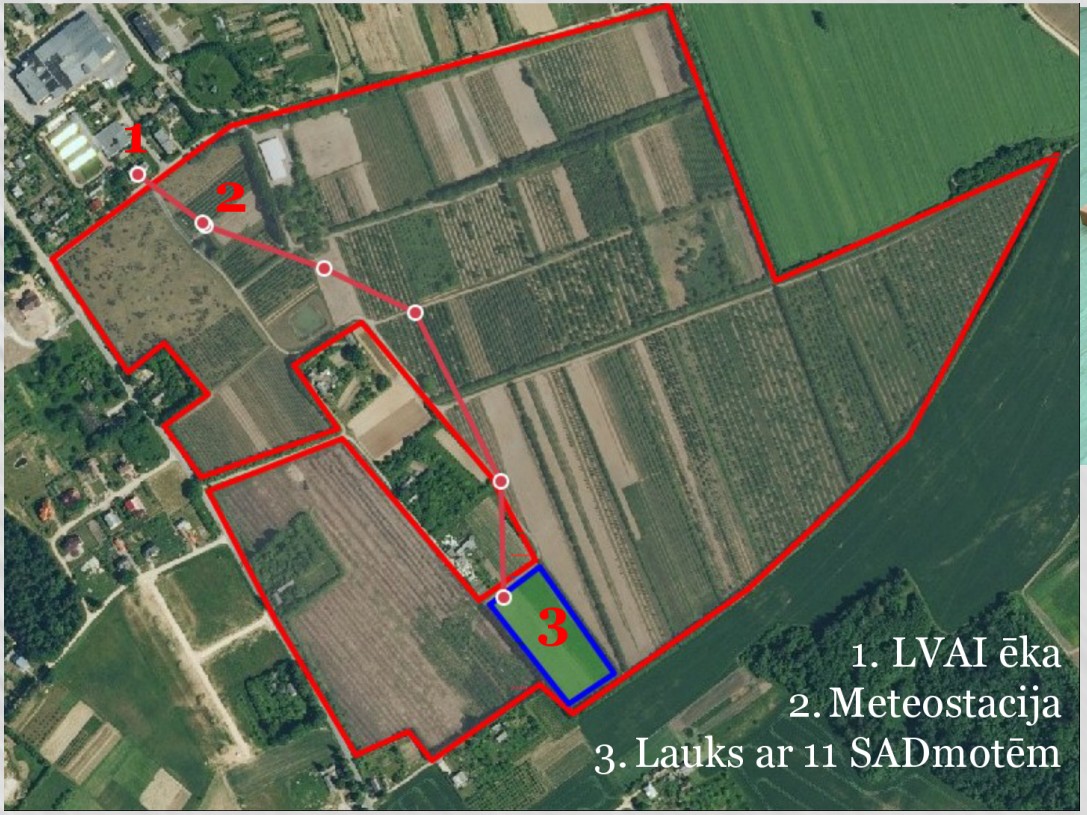
Lietojuma piemērs vides novērošanā

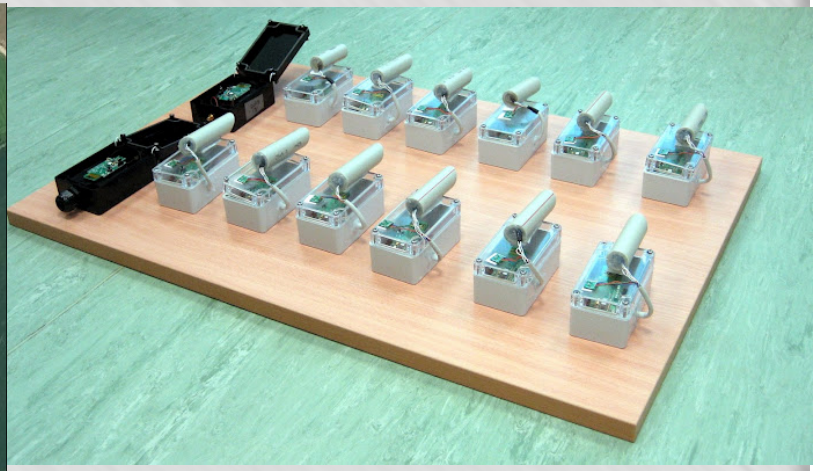
- Precīzā lauksaimniecība
- Dalīti mikroklimata mērījumi augļu dārzā:
  - temperatūra, mitrums, gaisma
- Lietotāji ir agrozinātnieki Latvijas Valsts Augļkopības institūtā (*http://lvai.lv*)



Testa tīkla arhitektūra

Lietojuma piemērs vides novērošanā





Risinājums

Konceptuāls prototips:

```
read Light, period 10min;
read Humidity, period 10min;
read Temperature, period 10min;
output SdCard;
output Network;
```

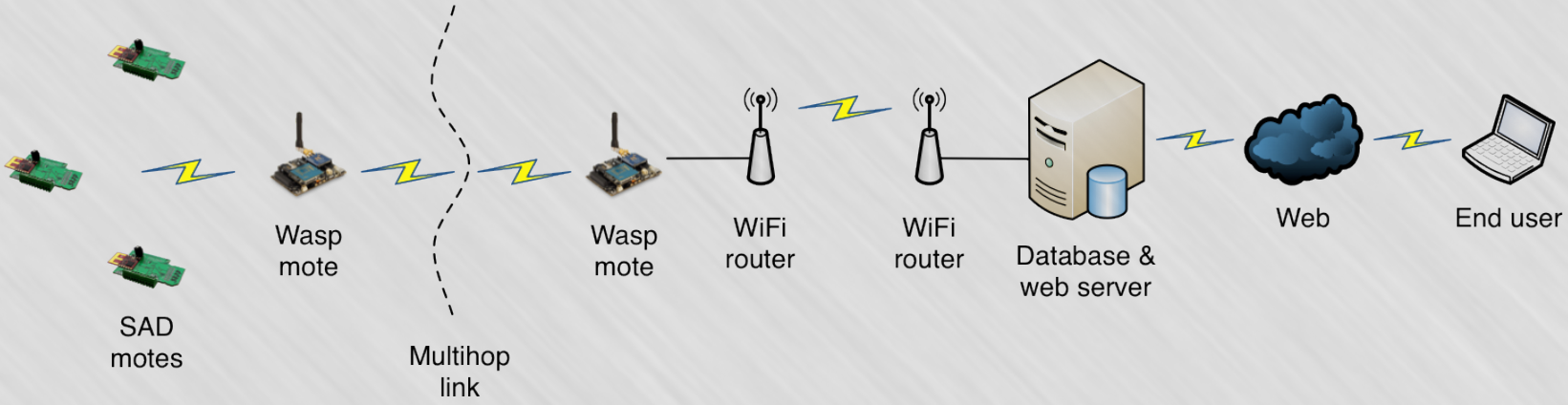
Risinājums

Reālās pasaules lietotne:

```
config "USE_LONG_LIFETIME=y"; // use 64-bit counter

define HelperLed RedLed, blink;
define Battery AnalogIn, channel constants.ADC_INTERNAL_VOLTAGE;

when Variables.localAddress != 0x0796:
    define MyLight Light; // use default light sensor
else:
    define MyLight SQ100Light; // use high-precision light sensor
end

define AllSensors sync(MyLight, Humidity, Temperature, Battery);
read AllSensors, period 10min, associate HelperLed, turnOnOff;

output SdCard;
output Network, protocol SAD, routing SAD;
```

19 mezglu sensoru tīkls tika izvietots ar šo kodu LVAI augļu dārzā Dobelē, 30. augustā, 2012.

Risinājums

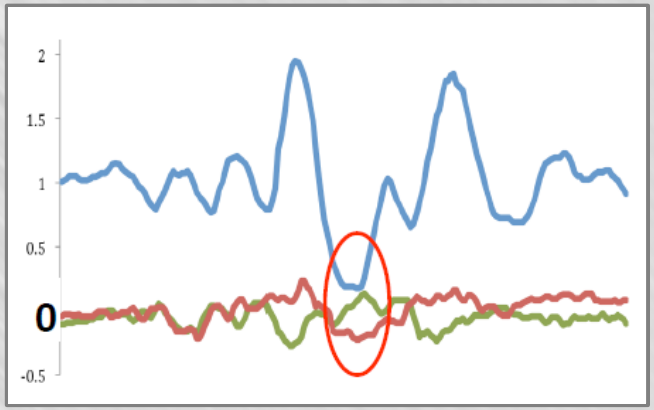


SEAL koda bonusi:

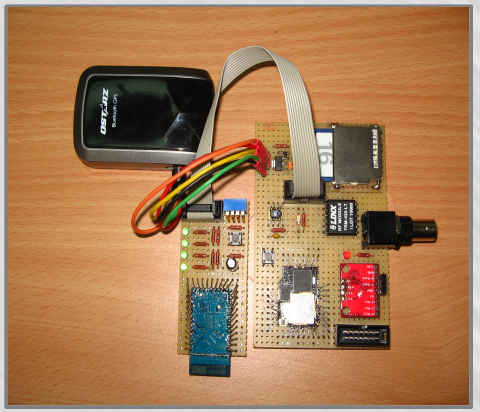
- Bāzes stacijas kods (MansOS C) ģenerēts automātiski!
- Pārsūtītāju mezglu kods (MansOS C) ģenerēts automātiski!
- SD kartes datu *parsēšanas* kods (Python) ģenerēts automātiski!
  - (Nākotnes plānos – FAT failu sistēma un .csv formāta faili...)

Lietojumpiemērs mobilajiem aģentiem

- Ar sensoriem aprīkotas ierīces, kas atrodas automašīnās
- Komunicē ar citām automašīnām un aparatūru ceļa malā
- Lietojums: noteikt bedres ceļa virsmas segumā [Mednis et al. 2011]
- EDI ir tikuši izstrādāti un izmēģināti četri algoritmi (valodā Java)
- Lietotāji ir aparatūras izstrādātāji un, potenciāli, auto īpašnieki



Bedru noteikšana ar akselerometra mērījumiem



Sensoru ierīces prototips

Risinājums

Izmantojot STDEV algoritmu Z asij:

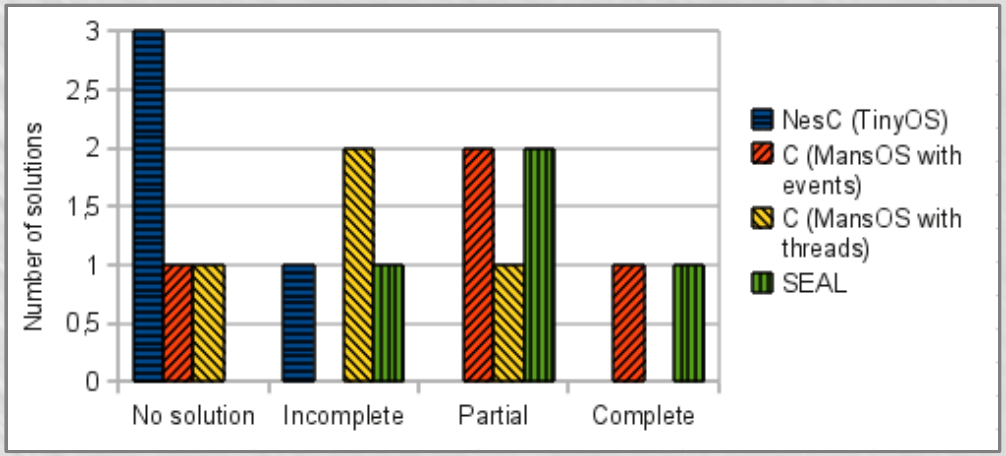
```
const ACCEL_Z 2; // define channel number

const THRESHOLD 100; // define threshold

define AccelZ AnalogIn, channel ACCEL_Z;
define Deviation stdev(take(AccelZ, 10));

when Deviation > THRESHOLD:
   use RedLed, on;
   use Beeper, on, duration 200, frequency 1000;
else:
   use RedLed, off;
end
```

Izmēģinājumu rezultāti - 2011

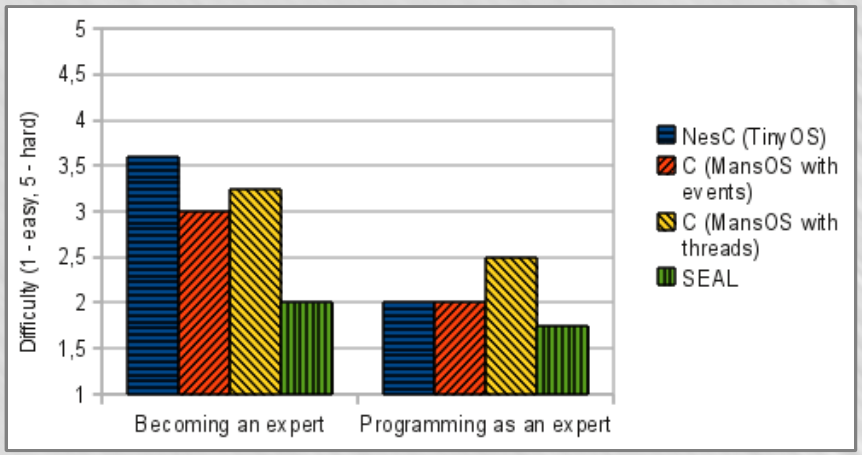
Veikti 2011. gada novembrī & decembrī

Veikti mērķauditorijā un tiem pielīdzināmā grupā:

- LVAI pētnieki bez programmēšanas pieredzes
- LU datorzinātņu studenti bez BST pieredzes



Studentu pārbaudījuma rezultāti, SEAL un konkurējošie risinājumi



Studentu dotais vērtējums, SEAL un konkurējošie risinājumi

Izmēģinājumu rezultāti - 2012

Veikti 2012. gada oktobrī

Norisies vieta, datums, dalībnieki (14 kopā):

- *University of Latvia, Faculty of Physics and Mathematics, October 5 2012, 4 physicists*
- *University of Latvia, Faculty of Computing, October 10 2012, 7 computer sci undergrad students*
- *Institute of Fruit Growing, October 11 2012, 2 agriculture scientists*
- *Institute of Electronics and Computer science, October 15 2012, 1 electrical engineer*

Secinājumi:

- Galīgi bez algoritmiskās domāšanas nevar!
- Datorikas studentiem: rezultāti analogi *TinyScript* rezultātiem
- Citiem SEAL rezultāti salīdzināmi ar ideoloģiski līdzīgu valodu (*SensorBASIC, WASP*) rezultātiem (>50% *success rate*)
- Īsāk ne vienmēr ir intuitīvāk
- Piemēri ir pieprasīta lieta

Kopsavilkums

- Operētājsistēma MansOS kalpo kā zinātniskās darbības bāze
- Izveidota jauna BST programmēšanas valoda SEAL:
  - Mērķauditorija ir tieši domēnu eksperti
  - Motivēta ar reālās pasaules lietojumiem
  - Atbalsta periodisku sensoru nolasīšanu, datu apstrādi, saglabāšanu un nosūtīšanu
- Šobrīd “virs” SEAL tiek izstrādāta vizuālā programmēšanas vide, kā arī adaptēta vizuālā programmēšana izmantojot “klucīšus” un tīmekļa interfeisu.

Studentu zinātniskā darbība

EDI pētniecības grupā:

- programmu, aparatūras izstrāde & izmēģinājumi
- noslēguma darbu izstrāde
- piedalīšanās zinātnisko rakstu tapšanā
- braukšana uz konferencēm (SenSys 2010, EWSN 2012, ARCS 2012, AICT 2012)

Rezultāti (LU studentu noslēguma darbi):

- Jānis Judvaitis, 2012: “Vizuāla bezvadu sensoru tīklu programmēšanas vide” (4. vieta valsts IT darbu konkursā)
- Andrejs Vihrovs, 2011: “Strukturēta datu glabāšana bezvadu sensoru tīklos”
- Rihards Balašs, 2012: “Enerģijas mērītāja izveide sensoru tīkliem” (kursa darbs)
- Jānis Ģenģeris, 2008: “Dalītā virtuālā failu sistēma bezvadu sensoru tīkliem”
- Ģirts Strazdiņš, 2008: “Satiksmes optimizācija ar mobiliem sensoru aģentiem urbanizētā vidē” (1. vieta valsts IT darbu konkursā)

MansOS un SEAL

Paldies par uzmanību!
Jautājumi?