

# ElectroTools – Educational software 2.0

Face parte din categoria software educational ce incearca sa studieze comportamentul circuitelor elementare in curent continu si alternativ. In cazul curentului alternativ singurele incercari de simulari au fost in regim stationar, adica pentru valori ale lui T mai mari decat constanta de timp al circuitelor, reamintim ca pentru valori mici ale timpului 0..., constanta de timp exista un regim aparte ale circuitelor electronice numit regim tranzitoriu pentru modelarea caruia se utilizeaza ecuatii diferentiale, in aceasta etapa softul meu nu studiaza aceasta parte.

Soft-ul permite studiul urmatoarelor parturi: Surse de tensiune, Surse de curent constant, surse de curent alternative cu frecventa fixa, defasaj si amplitudine fixa, pentru aceasta se editeaza amplitudinea, frecventa si faza. Sursele sunt ideale si se neglijeaza rezistenta lor interna dar poate fi adauga de utilizator prin utilizarea componentei R. Restul componentelor elementare sunt rezistenta, condensator, bobina.

S-a mers pe ideea ca utilizatorul sa poate edita orice instanta a problemei si nu doar niste animatii flash si niste animatii. Utilitatatea educationala a unui astfel de soft -s-a parut mai educational decat genul classic: lectie plus animatie flash.

Pentru realizarea lui intr-o prima versiune am utilizat teoremele lui Kirchhoff dar m-am lovit de o mare problema, rezolvarea sistemului liniar nu poate fi rezolvat cu algoritmul lui Cramer si teoremele lui Rouche si Kronecker-Capelli

Avem n ecuatii cu m necunoscute., unde n este este numarul teoremelor Kirchhoff1 plus Kirchhoff2 , intotdeauna n va fi mai mare decat m si intotdeauna sistemului este rezolvabil. Teorema lui Cramer se aplica pentru sisteme liniare omogene unde n=m, iar teormele lui Kronecker-Capelli si Rouche atunci cand cand numarul de necunoscute este mai mare decat numarul de ecuatii.

Marea problema este alegerea celor m ecuatii din multimea celor n ecuatii(m<n), cele m ecuatii trebuie sa fie intotdeauna liniar independente , pentru a asigura ca nu o sa fie solutia banala(toate necunoscute =0), deasemenea cele m ecuatii trebuie sa contina cele m necunoscute) , alegerea lor neputandu-se facanduse decat printr-un algoritm polinomial, combinari. Astfel se aleg combinari de n luate m

Kirchhoff1

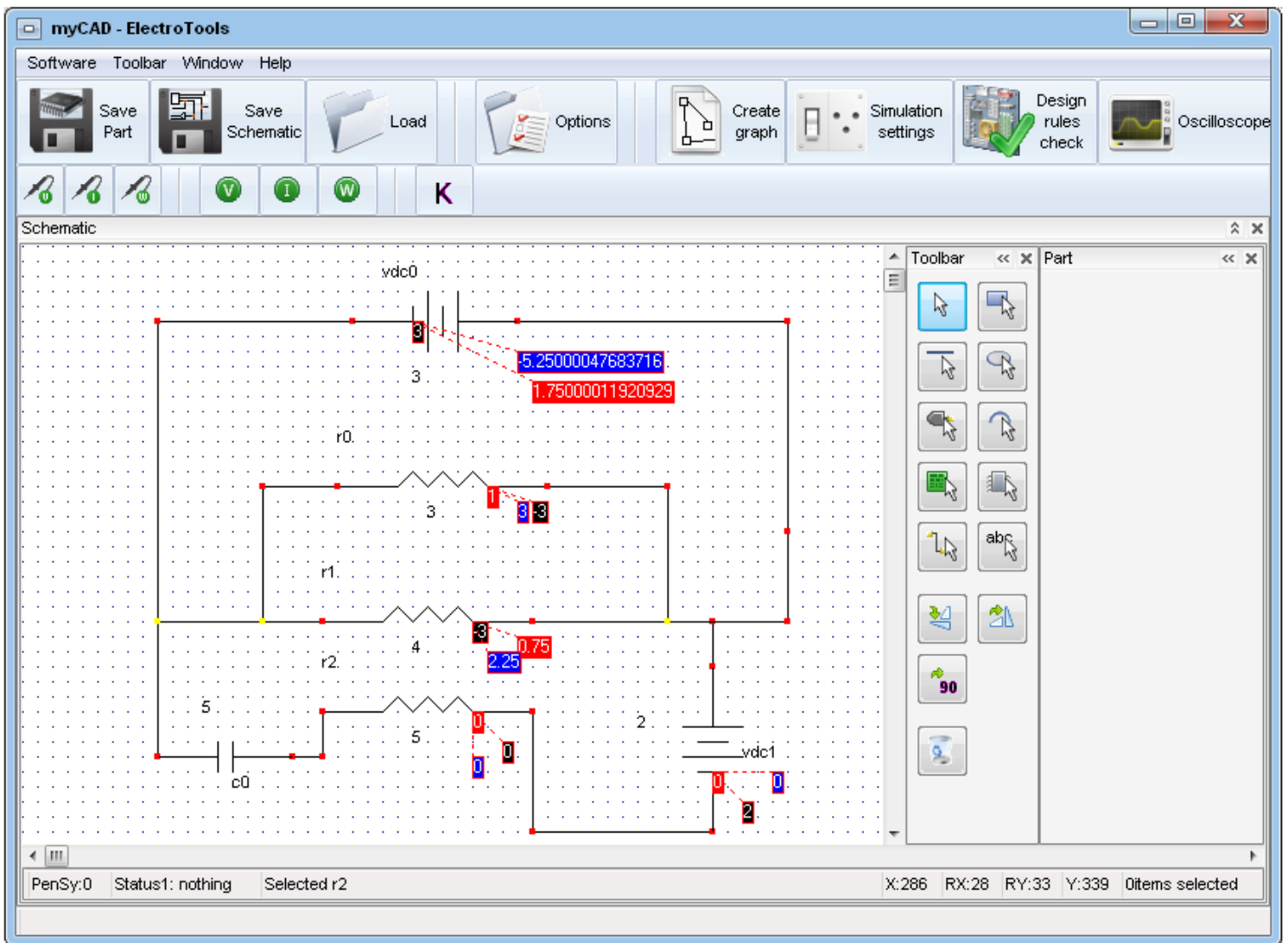
+Ir0 +Ivdc0 -idc1 +Ir4

-Ir0 +idc0 -Ir4

+Ir1 -Ir2 -Ivdc0 +Ir3

-Ir1 -idc0 -Ir3

+Ir2 +idc1



Kirchhoff2(nu este de la acest exemplu)

$$\begin{aligned}
 &+I_r0*r_0 +U_{idc0} -I_r1*r_1 -I_r2*r_2 +U_{idc1} \\
 &+I_r0*r_0 +U_{idc0} -I_r1*r_1 -v_{dc0} \\
 &+I_r0*r_0 +U_{idc0} -I_r3*r_3 -I_r2*r_2 +U_{idc1} \\
 &+I_r0*r_0 +U_{idc0} -I_r3*r_3 -v_{dc0} \\
 &+I_r0*r_0 -I_r4*r_4 \\
 &+v_{dc0} +I_r1*r_1 -U_{idc0} -I_r4*r_4 \\
 &+v_{dc0} +I_r1*r_1 -I_r3*r_3 -I_r2*r_2 +U_{idc1} \\
 &+v_{dc0} -I_r2*r_2 +U_{idc1} \\
 &+v_{dc0} +I_r3*r_3 -U_{idc0} -I_r4*r_4 \\
 &-U_{idc1} +I_r2*r_2 +I_r1*r_1 -U_{idc0} -I_r4*r_4 \\
 &-U_{idc1} +I_r2*r_2 +I_r3*r_3 -U_{idc0} -I_r4*r_4 \\
 &-I_r0*r_0 +v_{dc0} +I_r1*r_1 -I_r3*r_3 -I_r2*r_2 +U_{idc1} +I_r4*r_4 \\
 &-I_r0*r_0 +v_{dc0} -I_r2*r_2 +U_{idc1} +I_r4*r_4 \\
 &+I_r1*r_1 -I_r3*r_3
 \end{aligned}$$

Astfel pentru 8 componente in schema am avut 19 ecuatii pe care algoritmul facea combinari de 19 luate cate 8 de fiecare data se calcula un determinand de ordin 8, si se verifica cu teorema lui Rouché daca tot sistemul verifica solutiile gasite. Daca tot sistemului de 19 ecuatii verifica solutie gasite, pentru acest caz particular dura un sfert de ora.

Datorita faptului prezentat mai sus am implementat o alta metoda de rezolvare a sisteme

lor supradeterminate, prin metoda Gauss Elimination(numit Row Echelon Form), prin aceasta metoda aceeasi problema se rezolva sub 0.1 secunde pe aceleasi calculator.

Pentru a intelege metoda de rezolvare  
[http://en.wikipedia.org/wiki/Gaussian\\_elimination](http://en.wikipedia.org/wiki/Gaussian_elimination)

$$\begin{aligned}x+2y+ z&=1 \\2x- y+ z&=2 \\4x+3y+3z&=4 \\2x-1y+3z&=5\end{aligned}$$

sunt 4=n ecuatii 3=m necunoscute

Augmented matrix reprezinta matricea asociata

$$\left[ \begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 2 & -1 & 1 & 2 \\ 4 & 3 & 3 & 4 \\ 2 & -1 & 3 & 5 \end{array} \right]$$

Ideea principala este ca sub diagonala speciala sa fie 0, diagonala speciala sa fie egala cu valoarea 1.

$$\left[ \begin{array}{ccc|c} 1 & ? & ? & ? \\ 0 & 1 & ? & ? \\ 0 & 0 & 1 & ? \\ 0 & 0 & 0 & 0 \end{array} \right]$$

iar toate linile din intervalul [m+1, n] sa fie egale 0 , cat si termini liberi, ca sa verifice sistemul.

Se iau toate linia cu indice i de la [1..m] in ordine, se verifica daca element de pe pozitie i  $\neq 0$  atunci se incearca sa se imparta toata linia la o valoare astfel incat coeficientul elementului i sa fie 1. Daca este elemental i de pe linia i este = 0 atunci se cauta o alta linie din intervalul [i+1,n] astfel incat coeficientul elementului i sa fie diferit de 0, si se interschimba si se aplica acelasi procedeu din nou cu impartirea.

In cazul de fata pentru prima linie nu se aplica nimic. Dupa aceea se aduna la toatele celelalte lini ramase linia I astfel cat elementele de pe celelalte linii [i+1,n] sa fie egale cu 0.

In cazul de fata pentru a aplica procedeu:

A doua linie se va inmluti cu -2 prima linie si se aduna la linia 2.

A treia linie se va inmulti cu -4 prima linie si se aduna la linia 3

A patra linie se va inmulti cu -2 prima linie si se aduna la linia 4

Dupa acestea se obtine noua matrice

$$\left[ \begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & -5 & -1 & 0 \\ 0 & -5 & -1 & 0 \\ 0 & -5 & 1 & 3 \end{array} \right]$$

Dupa aceea i se face 2 si se selecteaza linia a doua. Elementul i-lea de pe linia i (elemental 2 de pe linia 2 = -5) este diferit 0 atunci toata linia se imparte la -5 ca elemental sa fie 1.

Linia se va transforma in 0, 1, 1/5, 0

Se obtine noua matrice

$$\left[ \begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & 1 & 1/5 & 0 \\ 0 & -5 & -1 & 0 \\ 0 & -5 & 1 & 3 \end{array} \right]$$

Dupa aceea la linie3 si linia4 se scad linia 2 astfel incat elemental 2 al fiecarei sa fie egal cu 0. Astfel se amplifica linia 2 cu 5 si se aduna la linia3, se amplifica linia2 cu 5 si se aduna la linia 4

. Dupa acestea se obtine noua matrice

$$\left[ \begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & 1 & 1/5 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 \end{array} \right]$$

Cand se selecteaza linia 3, se doreste ca elemental 3 sa fie diferit de 0, el nu este, se cauta in intervalul [3,4] o linie, care are elemental 3 diferit de 0, linia4 verifica aceasta cerinta si se inlocuete astfel:

Se obtine noua matrice

$$\left[ \begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & 1 & 1/5 & 0 \\ 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

Dupa aceea liniea3 se imparte la 2 astfel noua matrice

Se obtine noua matrice

$$\left[ \begin{array}{ccc|c} 1 & 2 & 1 & 1 \\ 0 & 1 & 1/5 & 0 \\ 0 & 0 & 1 & 2/3 \\ 0 & 0 & 0 & 0 \end{array} \right]$$

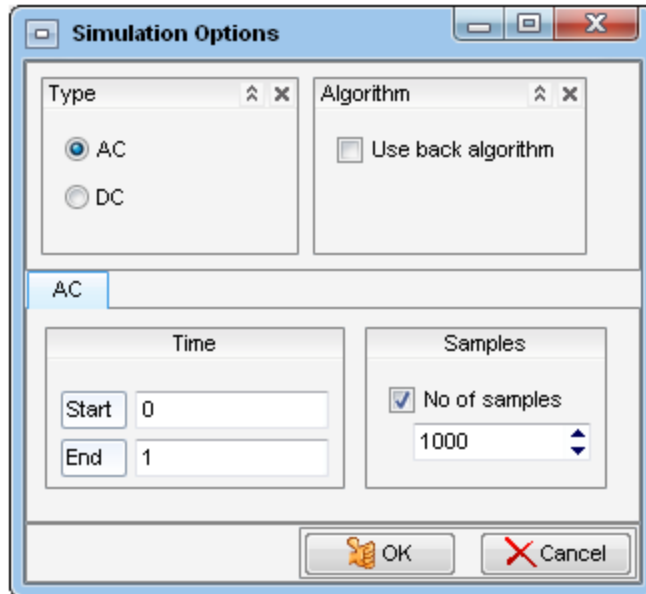
Algoritmul s-a terminat, acum se verifica daca sistemul admite solutii.. Pentru aceasta inseamna ca toate noile ecuatii [m...n] trebuie sa fie egale cu 0 si termeni liberi =0. In acest caz exista doar o singura ecuatie suplimentara ecuatia4(linia4), verifica ca 0 = 0, deci sistemul este compatibil determinat)

Iar recunostituirea este foarte simpla. Se porneste de la ecuatia 3 si se determina ca  $z=2/3$ .

Dupa aceea stim  $z$ , si incercam sa calcula  $y$ , ne vom duce la ecuatia nr2. si acolo avem ca  $y+1/5z = 0$ , deci  $y+1/5*3/2 = 0 \Rightarrow y = -3/2$ , stim  $y$  si  $z$  atunci putem sa calculam  $x$ , deoarece ecuatia implica  $x$ (nu stim)  $y$ (stim)  $z$ (stim), deci putem sa amplifica prima ecuatie.  $x+2y+z=1$  deci  $x-6/10+15/10-10/10 = 1/10$ .

Deci solutiile sunt  $z=2/3$ ,  $y=-3/2$ ,  $x=1/10$ .

## Simulation



Deasemenea se poate atasa un osciloscop pentru vizualizarea in domeniul timp-amplitudine a tensiuni si curentului in orice punct al schemei. Pentru modelarea in curentul alternative am simplificat problema in aceasta etapa prin utilizarea doar a surselor cu aceeasi frecventa, neputandu-se utiliza surse cu frecvente diferite.

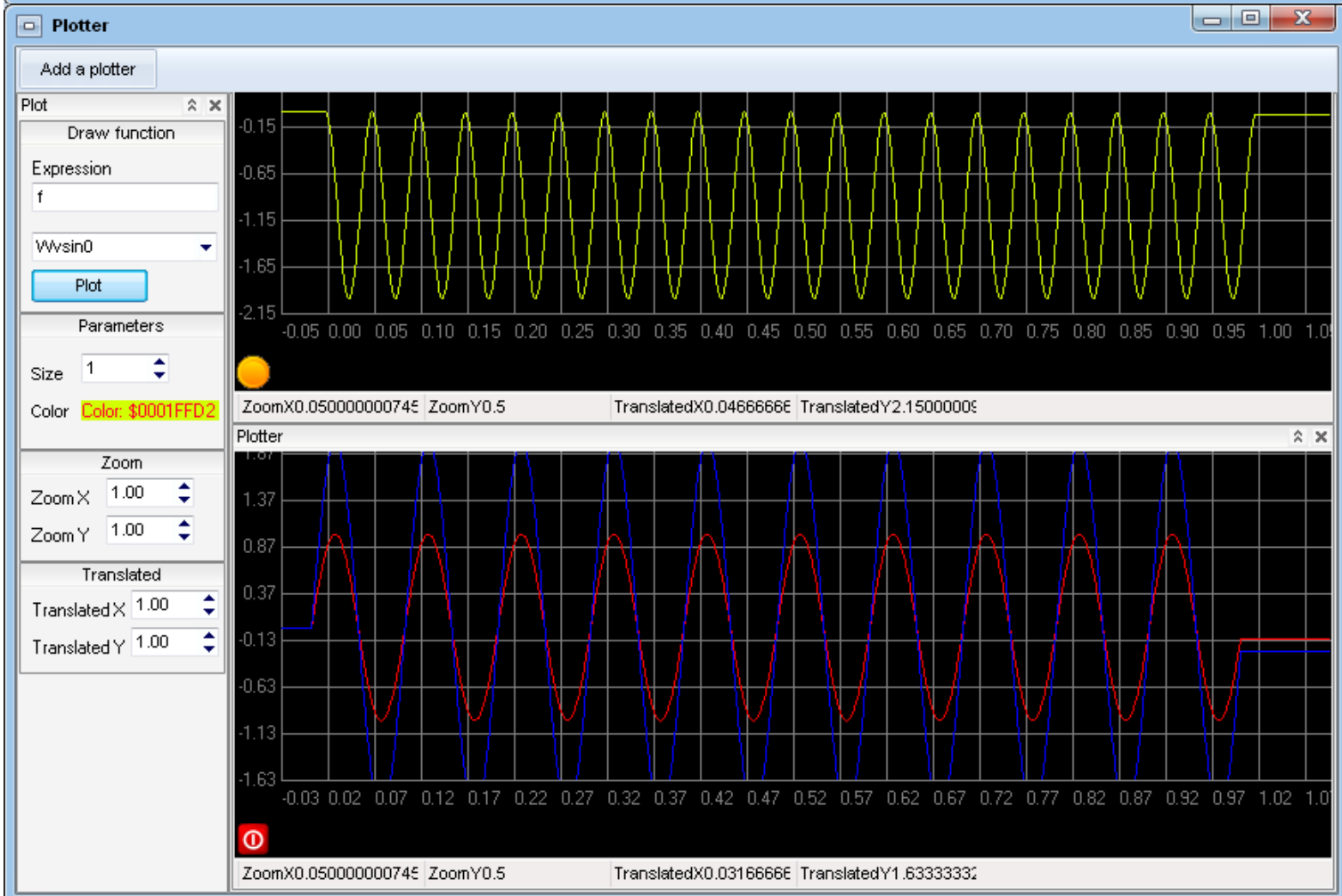
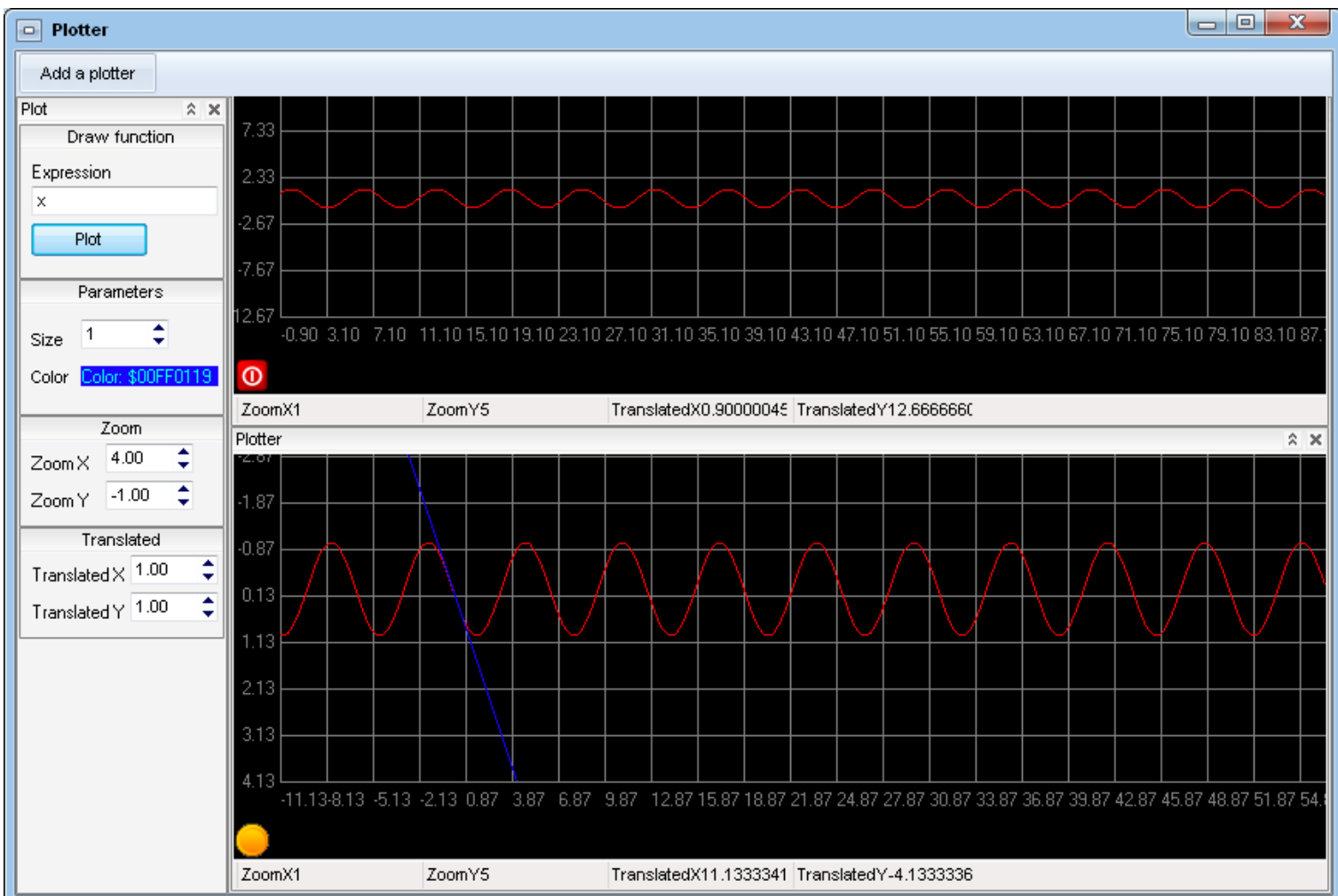
Pentru fiecare frecventa, in domeniul curent alternativ, bobinele si condensatoarelor se inlocuesc cu rezistenele lor echivalente(reactanta inductiva  $X_L$ , reactanta capacitive  $X_C$ )

$$X_L = \omega * l = 2 * \pi * f * l$$

$$X_C = 1 / (\omega * c) = 1 / (2 * \pi * f * c)$$

$$\text{perioada} = T = 1/f$$

Pentru plotarea comportamentului in curent alternativ am calculat esantioanele surselor la diverse momente  $t$  si am creat pentru fiecare esantion instanta ca si cum ar fi in curent continuu. Si se poate plota raspunsul circuitului la momentul  $T$ , plotandu-se toate raspunsurile pentru fiecare esantion, se obtine raspunsul circuitului la semnalul sinusoidal initial. Reamintind ca condensatoarele si bobinele au fost inlocuite cu rezistene echivalente.



## Toolbar

Toolbar-ul este unelata pentru a putea pune componentele si figurile pentru a putea crea parturi cat si scheme. Pentru a realiza flexibilitatea propusa programului am implementat functii pentru drag&drop put part, delete part, am realizat deasemenea posibilitatea de atasare in fiecare punct al schemei al unui voltmetru, ampermetru cat si posibilitatea de vizualizare a puteri pentru fiecare.



Permite crearea de dreptunghiuri. Este folosit deoseori pentru a delimita un part.



Permite crearea de lini pentru a putea crea elemente distinctive in part.



Permite crearea de elipse in design-ul part-ului.





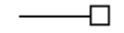
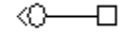
Permite crearea de arce in design-ul part-ului.

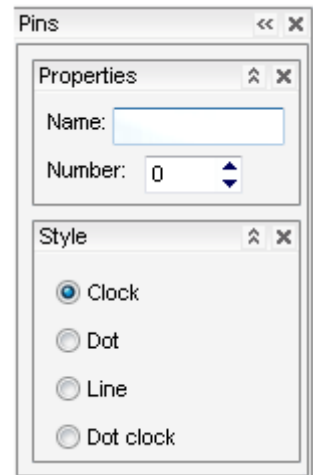


Permite crearea de pini intr-un part. Fiecare part contine pini, cu acesta se pot introduce pini la part-ul respective.

Proprietatile pinilor sunt precum:  
 Nume –folosit pentru a afisa in schematic  
 Numar- folosit pentru a numerota pini si a stabili semnu ecuatiei

Stilul – tipul pinului.

-  Pin prin insertie si prezinta clock, reprezinta clockul pentru cipurile digitale
-  Prin prin lipire
-  Pin prin insertie
-  Pin prin lipire si prezinta clock

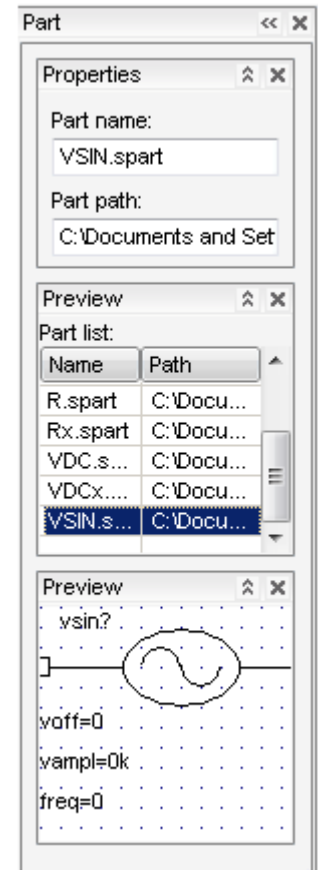
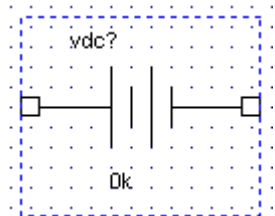


Permite stergerea elementelor din schematic.



Permite insertia in schematic a part-urilor deja create.  
 Preview este o instanta a part-ului pentru a putea vedea diferenta dintre parturi.

Part Path este calea efectiva partului de unde este inclus.

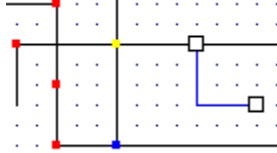


Permite creare de suprafata, pentru a putea delimita zona de selectie a mousului pentru part. Nu se poate crea un part fara acest component.

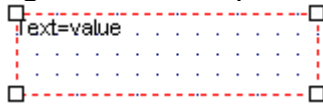




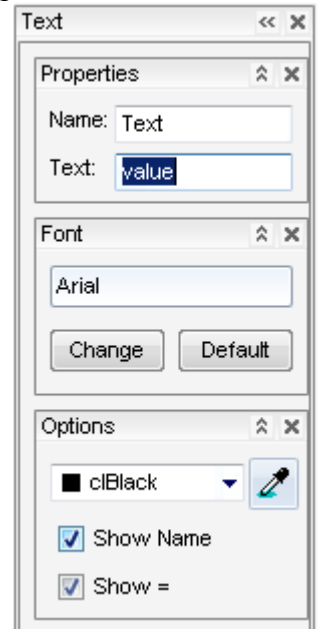
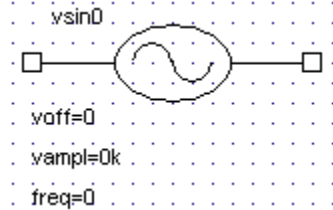
Firele sunt create a de a lega pini de la parturi intre ei.



Se pot folosi pentru a putea lasa comentarii in schematic, dar si pentru a putea asigra variabile de parturi precum (nume, frecventa, etc...)



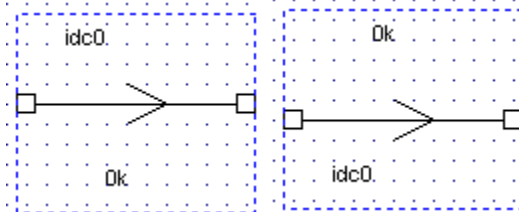
Proprietatea Name, este numele etichetei, are un rol important pentru parturi, pentru a desemna anumit parametri ai partului. De exemplu part-ul VSIN prezinta 4 parametri: precum nume – numele partului  
voff – offset(deplasament)  
vAMPL – amplitudinea semnalului generat de sursa????  
freq - frecventa semnalului generat de sursa.



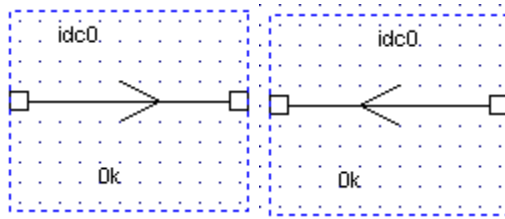
Acesti 4 parametri sunt introdus in ecuatiile din Kirchhoff1 si Kirchhoff2



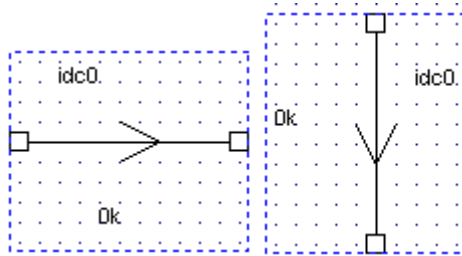
Permite rotirea partului/componentului pe verticala.



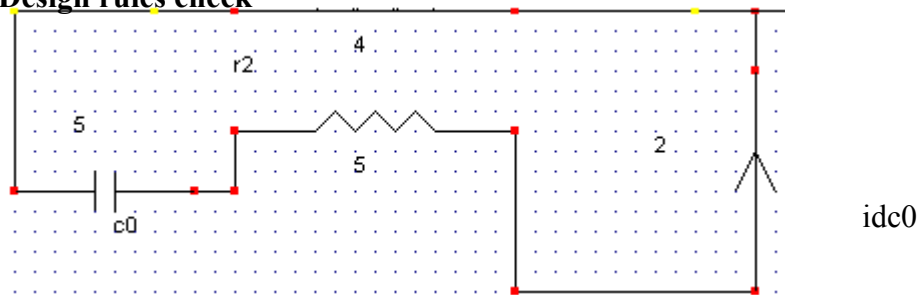
Permite rotirea partului/componentului pe orizontala.



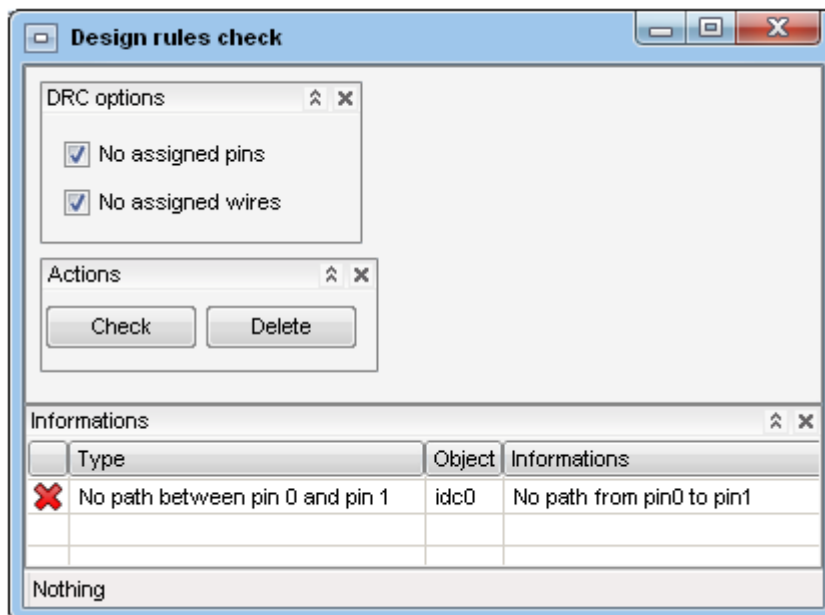
Permite rotirea cu 90 de grade a partului/componentul



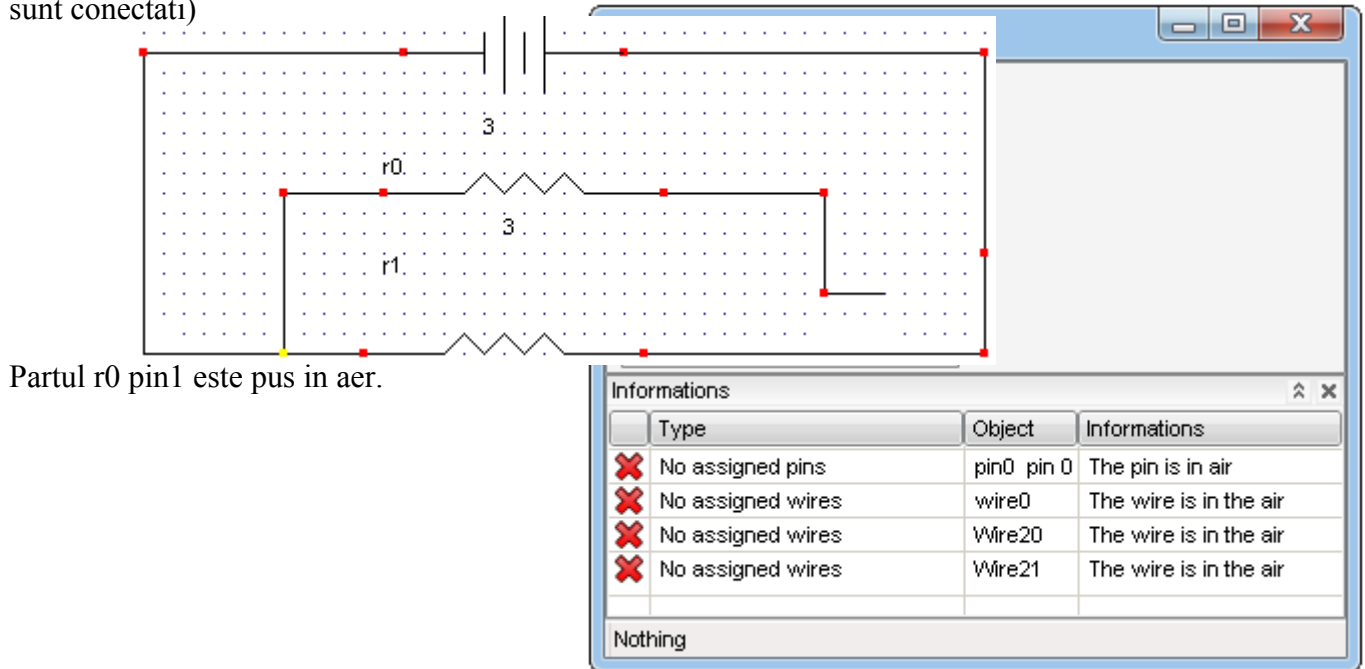
### Design rules check



Daca am rula o simulare in curentul continu, partul c0, care este un condensator se comporta ca un fir interupt astfel nu exista un drum(lant) la partul idc0 de la pin0 la pin1. Deoarece firul este interupt, astfel nu se poate simula .



Deasemenea nu se poate simula daca unul dintre pini la orice part sunt in aer(nu sunt conectati)



### Sistem expert

Softul permite interpretarea limbajului uman. Contine si o foarte mica baza de cunostinta care in etapa actuala nu am reusi sa o utilizez inca. Interpretarea limbajului uman presupune o analiza lexicala si semantica unor propozitii iar apoi o interpretarea propozitiilor.

Initial s-a scris folosind turbo prolog, scriind cu ajutorul programarii logice, dar din cauza cresterii complexitatii s-a renuntat si s-a implementat in C++.

Momentan permite: Substantive, Verbe, Prepozitii, Legaturi, Adjective, Adverbe, Articole... s-a utilizat metoda de rezolvare cu automate finite recursive.

Se poate face un paralelism intre procesarea unui limbaj uman si un limbaj de programare.

Limbajele de programare au o gramatica mica, in ideea:

Pascal – sa fie cat mai umana, de exemplu pentru a crea un bloc de scrie begin, si end;

C/C++ - sa fie cat mai putin deschis, optim, de exemplu aceeași problema se scrie cu doar 2 litere { si }

In limbajul uman, problema este: ca in decursul timpului oamenii au incercat sa creeze o sintaxa cat mai frumoasa, cat mai interesanta, de exemplu in engleza de ce exista cazuri particulare pentru plural? Din cauza sa "sune mai frumos" (o rezonanta) la cuvantul address plural devine addresses, intrucat era destul de greu si "ciudat" de pronuntat address si din cauza aceasta limbajul uman contine foarte multe cazuri particulare, interpretari in functie de context al cuvintelor, si alte particularitati. Cand se incearca scrierea unui soft ce permite natural language processing (NLP) este foarte complicat, deoarece trebuie luate toate in calcul foarte multe cazuri particulare cat si semantica exprimarii..

De exemplu avem un substantiv: calculator, prima forma este masculin, singular, daca am incerca sa punem un articol nehotarat singurul care se potriveste este un calculator, intrucat si articolul contine aceasi informatii(art nehot este masc si singular) daca am incerca sa pune o calculator(este incorect intrucat suna "ciudat" si pentru a nu mai suna ciudat, in timp s-a introdus notiunea ca atat subst si articolul trebuie sa aibe acelasi gen si numar). Am facut aceasta paranteza pentru a arata dificultatea problemelor ce trebuie depasite in NLP, deci softul incearca sa faca o analiza sintactica si semantica a corpusului de cuvinte. Reamintesc ca corpusul de cuvinte reprezinta vocabularul limbajului, corpusul deocamdata este limitat dar datorita exploziei combinationale textele interpretate pot fi in numar foarte mare.

Componenta face analiza sintactica a cerintei(query-ul) si daca este corecta din punct de vedere lexical , verifica din punct de vedere semantic si eventual executa o actiune.

De exemplu "Eu as vrea ca tu sa salvezi tot " este corect sintactic, dar semantic nu, intrucat nu am specificat unde sa salveze. Sau "Sa salvezi tot in calculator" este corect dar nu are sens. Se poate scrie pentru orice formulare corecte si semantic orice formular corect semantic o actiune pe care sa o genereze soft-ul.

Sistemul expert propus de mine permite urmatoarele genuri de propozitii(template-uri, ele pot fi dezvoltate) reamintesc ca nu se rezumeaza doar la acestea:

sterge tot

sterge "A" si "b"

sterge din schematic "a" si "b"

sterge din schematic doua parturi "a" si "b"

sterge un part din schematic "a" si "b"

sterge un part "a" si "b"

sterge doua parturi "a" si "b" si "c"

salveaza tot in "C:"

salveaza in "C:"

salveaza un schematic in "C:"

salveaza un fisier in "C:"

salveaza un schematic "c:"

salveaza un schematic curent in "C:"

deschide din "C:"

deschide aplicatie

deschide fisier

deschide un schematic din "C:"

deschide un schematic "C:"

deschide un schematic nou "C:"

inchide o aplicatie

inchide un schematic.

inchide tot

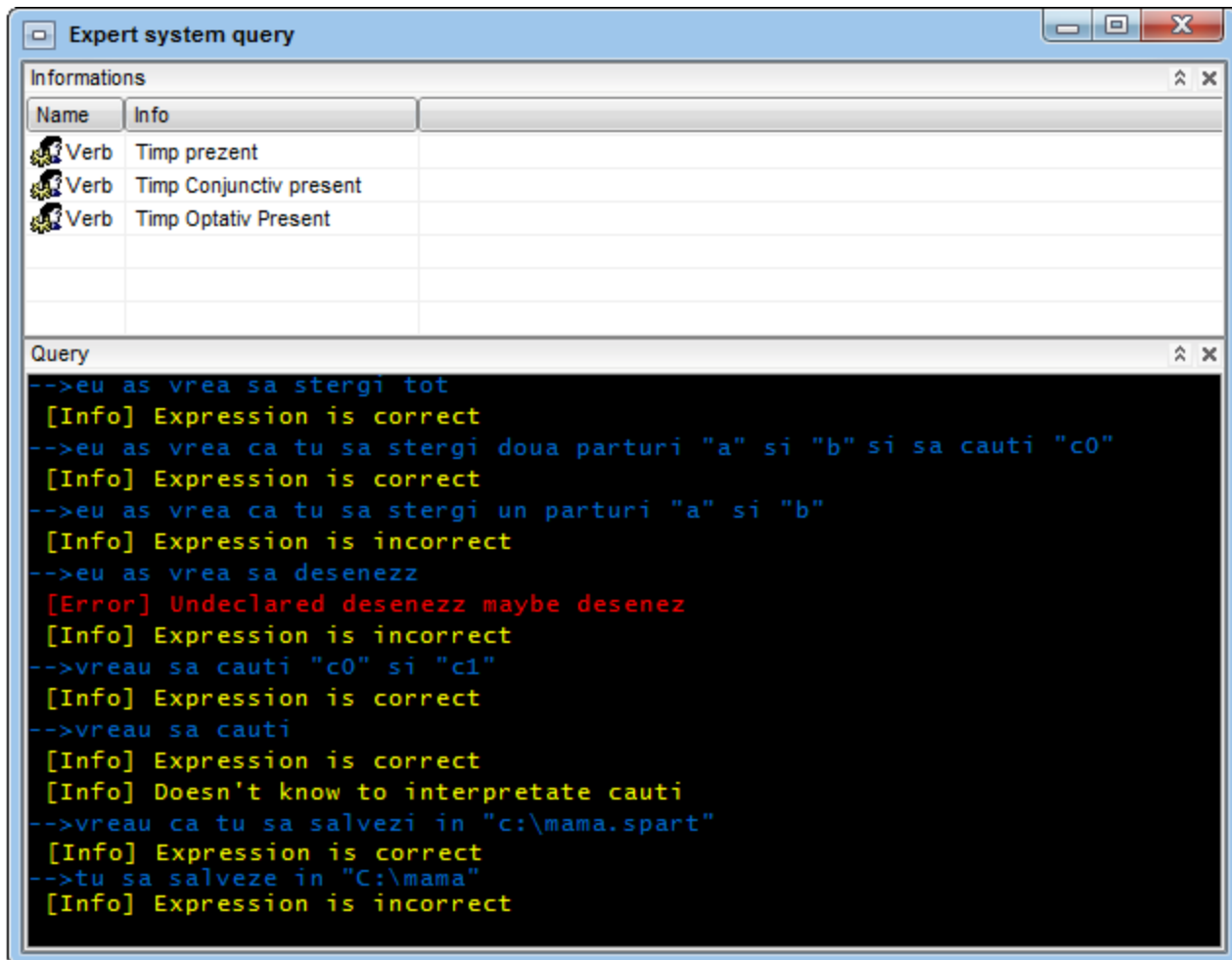
cauta tot "c"

cauta doua parturi "c0"  
cauta in schematic doua parturi "c0"  
cauta tot in schematic doua parturi "c0"

### Deasemenea permite inlantuirea a mai multora precum

Eu as vrea ca tu sa stergi doua parturi "a" si "b" si sa cauti "a"  
Eu as dori ca sa stergi tot  
Vreau sa stergi tot si tu sa inchizi o aplicatie  
Vreau ca el sa stearga tot  
Doresc sa cauti in schematic doua parturi "a" si "b" si "c"  
Eu as dori sa deschizi o aplicatie  
Eu as dori sa sterge din schematic un part "a"

Aplicatia reuseste sa-si dea seama de analiza lexicala din:



The screenshot shows a window titled "Expert system query". It has two main sections: "Informations" and "Query".

The "Informations" section contains a table with the following data:

Name	Info
Verb	Timp prezent
Verb	Timp Conjunctiv present
Verb	Timp Optativ Present

The "Query" section shows a log of interactions:

```
-->eu as vrea sa stergi tot  
[Info] Expression is correct  
-->eu as vrea ca tu sa stergi doua parturi "a" si "b" si sa cauti "c0"  
[Info] Expression is correct  
-->eu as vrea ca tu sa stergi un parturi "a" si "b"  
[Info] Expression is incorrect  
-->eu as vrea sa desenezz  
[Error] Undeclared desenezz maybe desenez  
[Info] Expression is incorrect  
-->vreau sa cauti "c0" si "c1"  
[Info] Expression is correct  
-->vreau sa cauti  
[Info] Expression is correct  
[Info] Doesn't know to interpretate cauti  
-->vreau ca tu sa salvezi in "c:\mama.spart"  
[Info] Expression is correct  
-->tu sa salveze in "C:\mama"  
[Info] Expression is incorrect
```

Intrucat corpusul lingvistic este restrans, si sunt anumite cuvinte, softul permite un intellisense doar pentru cuvinte, selecteaza doar cuvintele dupa prefix, lasand utilizatorului sa aleaga un verb, adverb sau ce element lexical util doreste, nu spune si ce parte de vorbire urmeaza(se potriveste)



## Baze de cunostinte

Ideea este ca toata teoria sa fie reprezentata sub forma de baze de cunostinte, momentan se reprezinta sub forma de clauze prolog. Apoi prin interferentiera lor se pot deduce mai multe informatii.

Exemple de propozitii care pot fi scrise sub forma de text

Omul este inteligent  
parturile sunt obiecte si masinile deseneaza  
parturile sunt si obiecte  
parturile sunt obiecte si masini  
parturile sunt obiectele  
Omul este inteligent  
Electronistul este un om  
Edison este electronist  
Partul este un obiect  
Este edison inteligent?  
Sunt electronistii oameni?  
Obiectele se pot sterge daca nu sunt fire  
obiectele se pot sterge si desena daca nu sunt fire  
Electronistul este un om inteligent si cunoaste daca firul se poate sterge si desena si  
daca masina se poate sterge  
Electronistul este si cunoaste

Edison descopera becul electric  
este edison inteligent  
Toti electronistii sunt oameni

## Iar reprezentarile lor aferente

```
inteligent(om).  
obiect(part).  
a_desena(masina).  
obiect(part).  
obiect(part).  
masina(part).  
inteligent(om).  
om(electronist).  
electronist(edison).  
obiect(part).  
?- nl,print('q10 '), inteligent(edison),print('true').  
a_sterge(obiect) :- not(a_fi(fir)).  
a_sterge(obiect) :- not(a_fi(fir)).  
a_desena(obiect) :- not(a_fi(fir)).
```

```

om(electronist) :- a_sterge(fir), a_desena(fir), a_sterge(masina).
a_cunoaste(electronist) :- a_sterge(fir), a_desena(fir), a_sterge(masina).
a_cunoaste(electronist).
a_descoperi(edison, bec).
intelligent(edison).
?- nl, print('q19 '), a_descoperi(edison, A), print(A), print(' '), print('true').
?- nl, print('q20 '), a_sterge(fir), print('true').
?- nl, print('q21 '), om(electronist), print('true').
?- nl, print('q22 '), a_descoperi(edison, A), print(A), print(' '), print('true').

```

## 1. Speech recognition

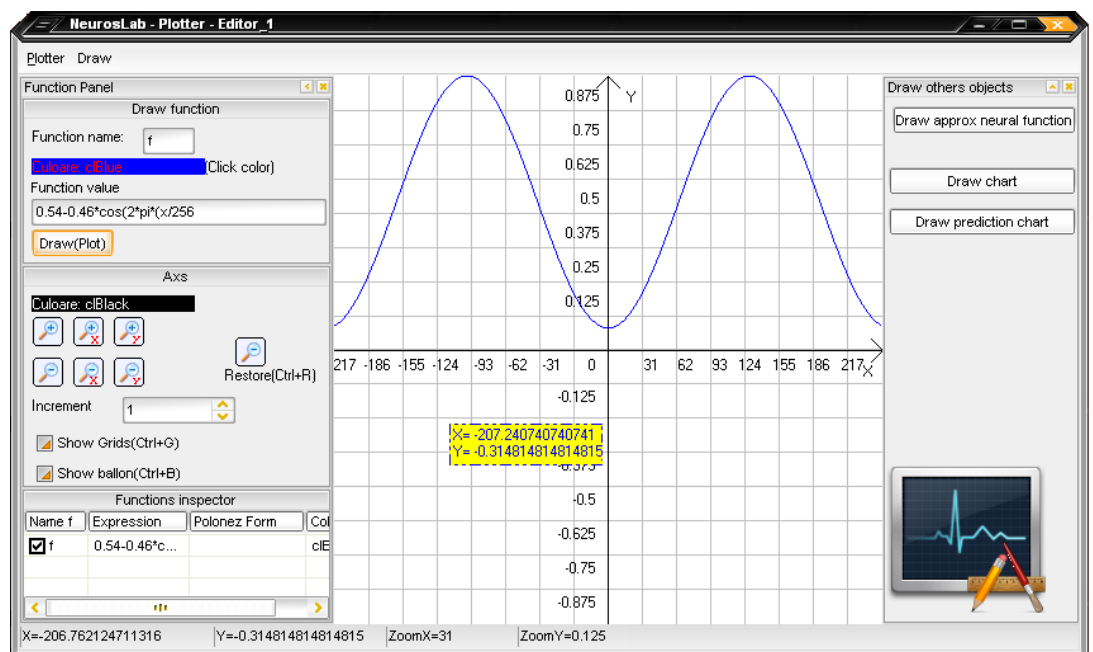
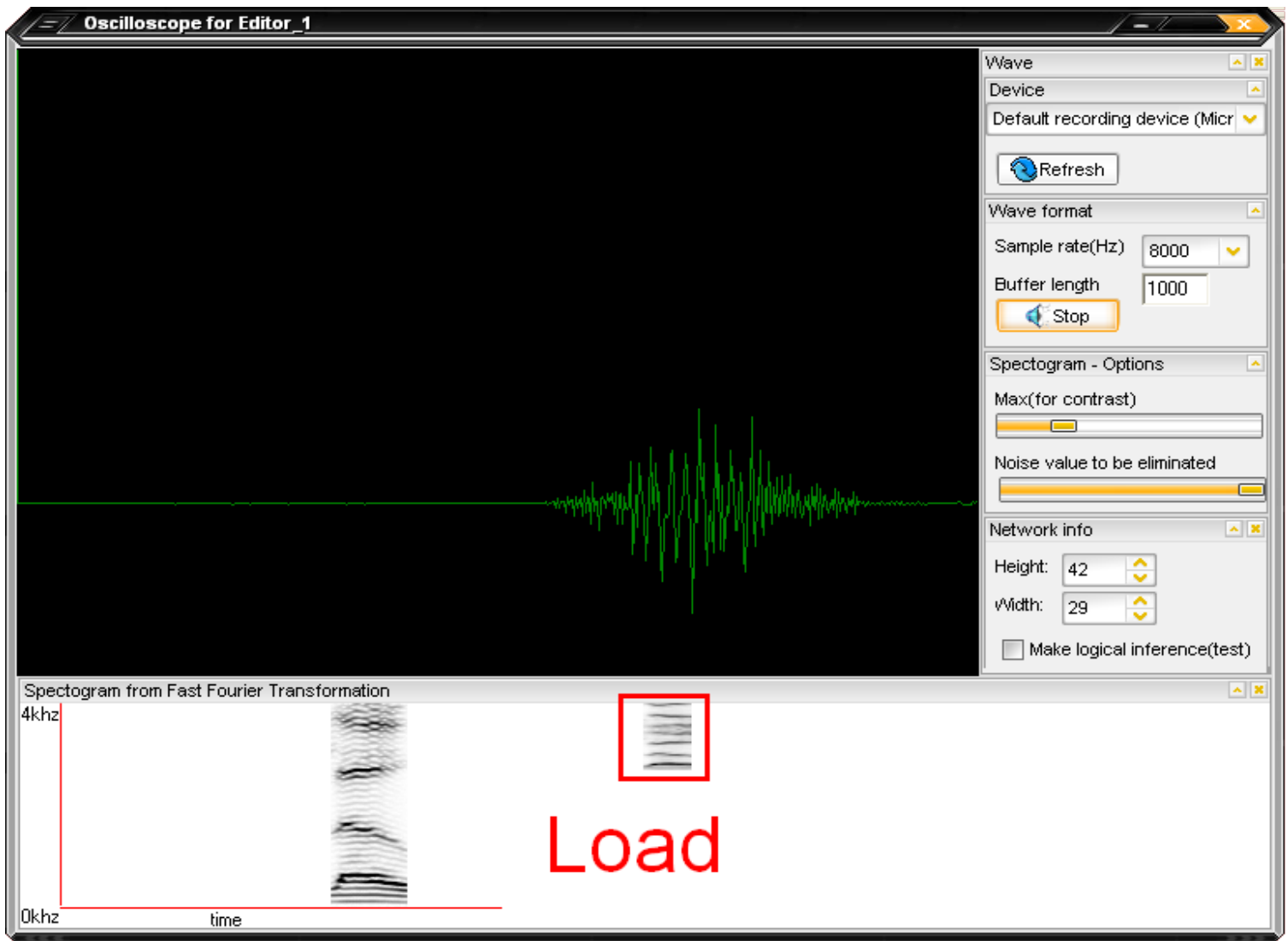
S-a incercat deasemenea in care softul ElectroTools sa recunoasca diferite cuvinte vorbite la microfon. Momentan softul reuseste sa recunoasca doar cateva cuvinte precum: search, delete, new, save, load, stop. Ideea este ce a crea noi modalitati de interfete om calculator pentru elevi si de a le putea folosi.

Intrucat ANN nu stiu sa faca decat clasificari, trebuie sa transform problema mea intr-o problema de clasificare. Sunetul trebuie transformat intr-o poza, pe care retea neuroanala sa incerce sa o classifice. Crearea unei poze in domeniul amplitudine-timp nu este utila. Trebuie lucrat in domeniul timp-frecventa.

Dupa aceea, am vrut sa vad daca o retea neuronală MLP poate sa classifice sunete. Astfel s-a folosit clasa TMicrophone, (oscilloscop-ul in domeniul timp amplitudine), am setat frecventa de esantionare pe 8000. Cand am apelat functia de desenat esantioanele, nu am scos componenta continua (DC) [Figura 3.1]. Dupa aceea am sczut DC (componenta continua) si inmultit cu fereastra Hamming, pentru a imbunatati calitatea sunetului inregistrat de la microphone.

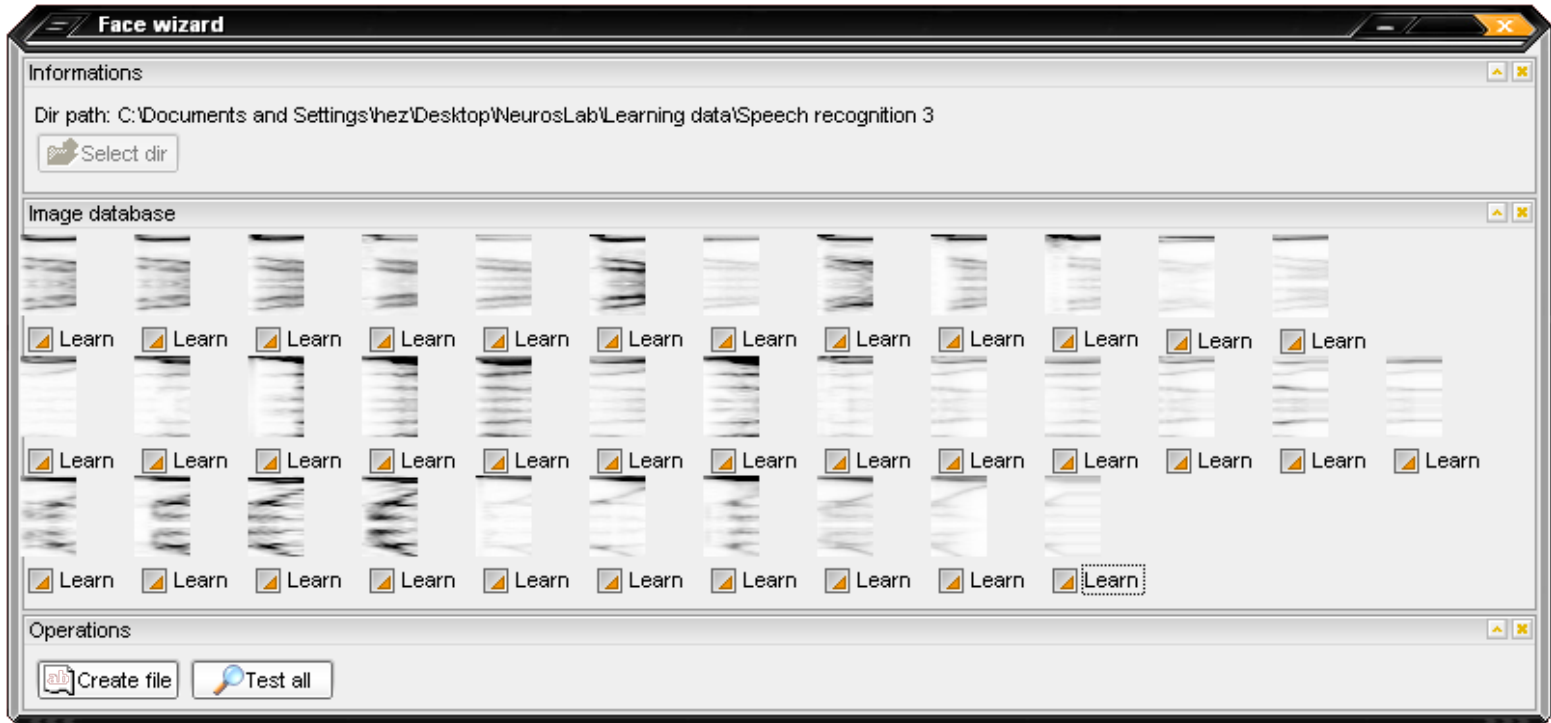
Fereastra hamming este:  $\text{hamming}(x) = 0,54 - 0,46 \cdot \cos(2 \cdot \pi \cdot (x/256))$  [Figura 3.2]. Aceasta functie este clasica.

Algoritmul cu fereastra hamming: Iau primele 256 [1..256] de esantioane le introduc intr-un buffer si il inmultesc cu valorile hamming. Dupa ce calculez prima fereastra o introduc la analiza Transformata Rapida Fourier, din rezultat fac modulul numerelor complexe dupa transformata, si obtin cate o coloana, din imaginea gray-scale, a spectrogramei sunetului. Deci o sa obtin  $8000 / 30$  (numarul de esantioane, luate intr-o secunda) / 30, unde 30 este deplasamentul ferestrei hamming [Figura 3.1]. Dupa aceea, reiau procedeul, iau intervalul [30, 286] si aplic aceeasi inmultire. Dupa aceea [60, 316]





Valorile pe care le obtin, in spectograma, le normez, si le inmultesc cu 255, pentru a le transforma in valori de la [0..255]. Aplic diverse metode, pentru a elimina zgomotul prin mediere, componenta continua(DC), si a separa, cuvintele de zonele de pauza. Am aplicat diverse nivele de separare a zgomotului, si a intervalului de pauza.



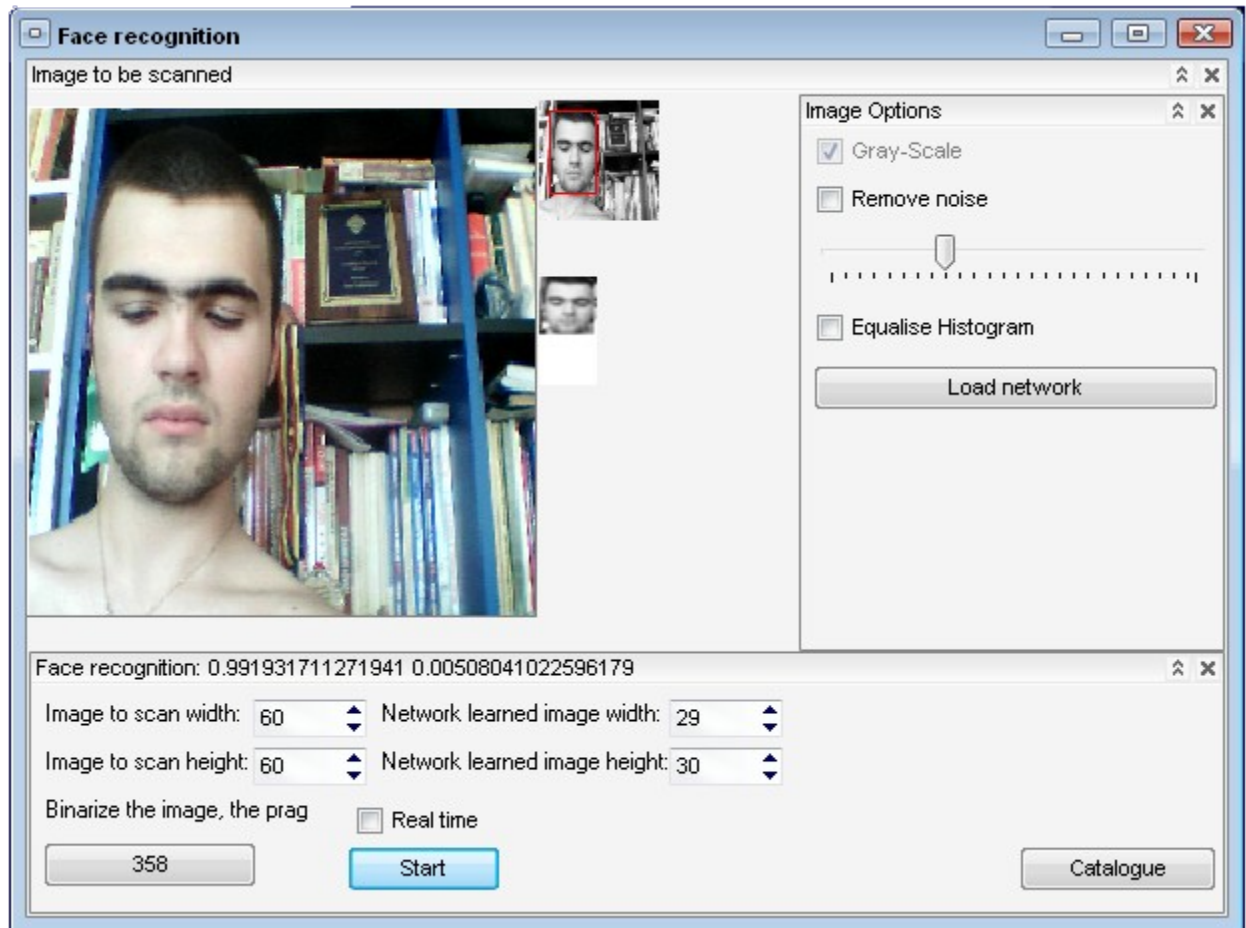
**Figura 3.3** aici sunt spectogramele a 3 cuvinte(save, load, new), dupa aceea se obtine fisierul de antrenament, si dupa aceea se invata retea neuronală

Concluzie la recunoasterea cuvintelor.

Separarea sunetelor in silabe, si apoi clasificarea lor, nu ar duce la o imbunatatire a recunoasterii vocale, intrucat silabele sunt foarte apropiate ca spectre, si trebuie o clasificare mai puternica, admitand ca am reusi sa facem clasificarea silabelor, ne trebuie o inferenta puternica, pentru a recrea cuvintele, posibil un arbore de decizie. Metoda adoptata, cu clasificarea spectrelor cuvintelor, ar fi dificil de implementat, pt un nr mare de cuvinte, dar se bazeaza pe faptul ca distanta intre spectre e foarte mare. O alta directie de dezvoltare ar putea fi crearea unor inferente logice ce separa cuvintele dupa lungime.

## Face recognition

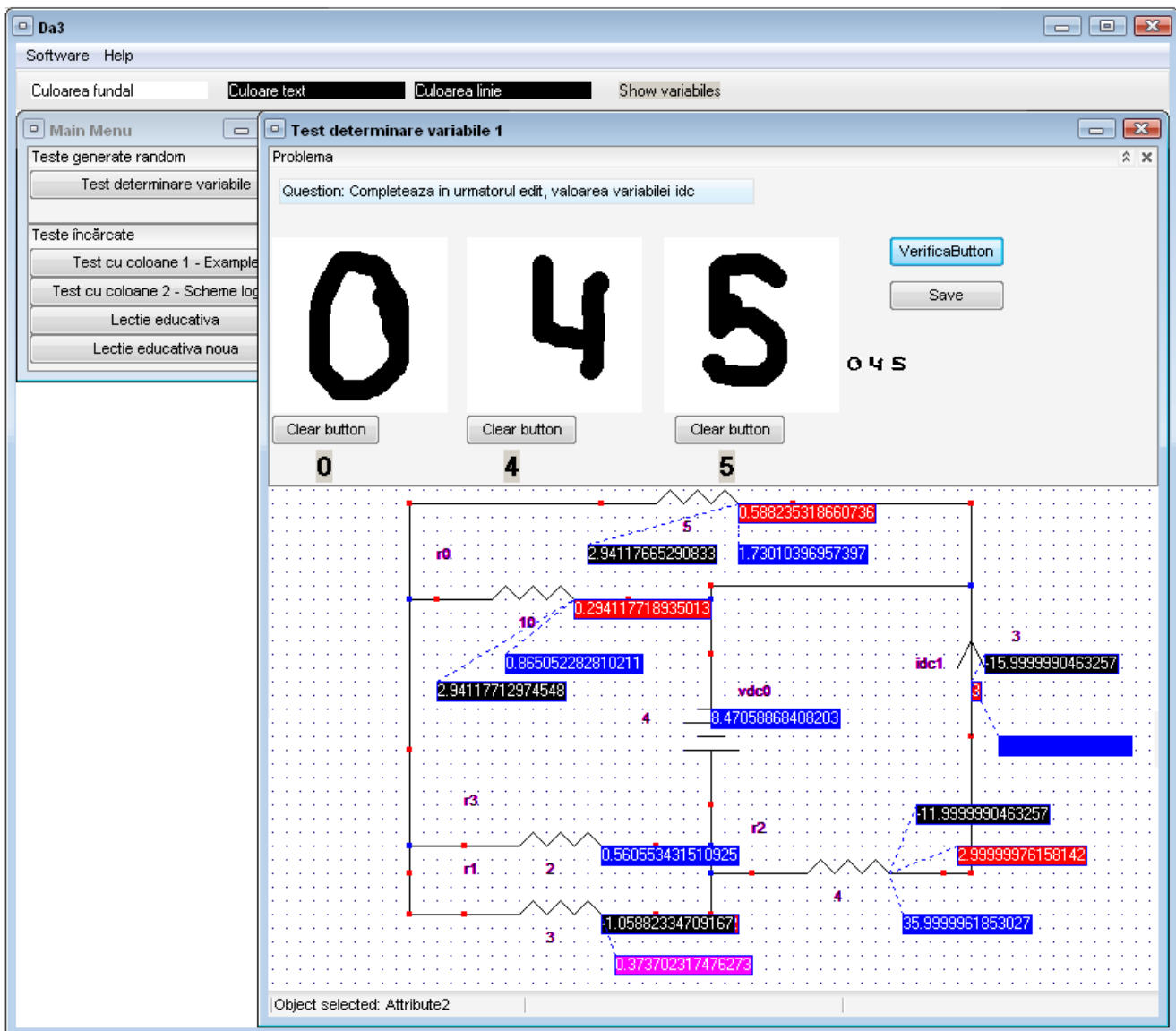
Softul permite logarea in zona de administrare dupa ce are loc o recunoastere faciala + o clasificare faciala. Atat partea de recunoaste facila,a cat si partea de clasificare, faciala este perfect functionala, inasa recunoasterea real-time mai are probleme. Totusi scopul dorit a fost atins, implementarea a astor concepte in softurile educationale. Mentionez ca partea de recunoasterea faciala, cat si partea de clasificare recunoastere, este scrisa completa de mine, si prezentata in concursuri. Nu am folosit un nici un SDK.



## Partea educationala:

Ideea este sa se genereze teste random de fiecare data. Exista 2 tipuri de teste A si B. In A o schema este creata aleator iar apoi elevul trebuia sa raspunda „In curent continu ce valoare are” si dupa aceea variabila se genereaza random poate sa fie curentul care trece prin nodul respectiv, tensiunea fata de masa, puterea.

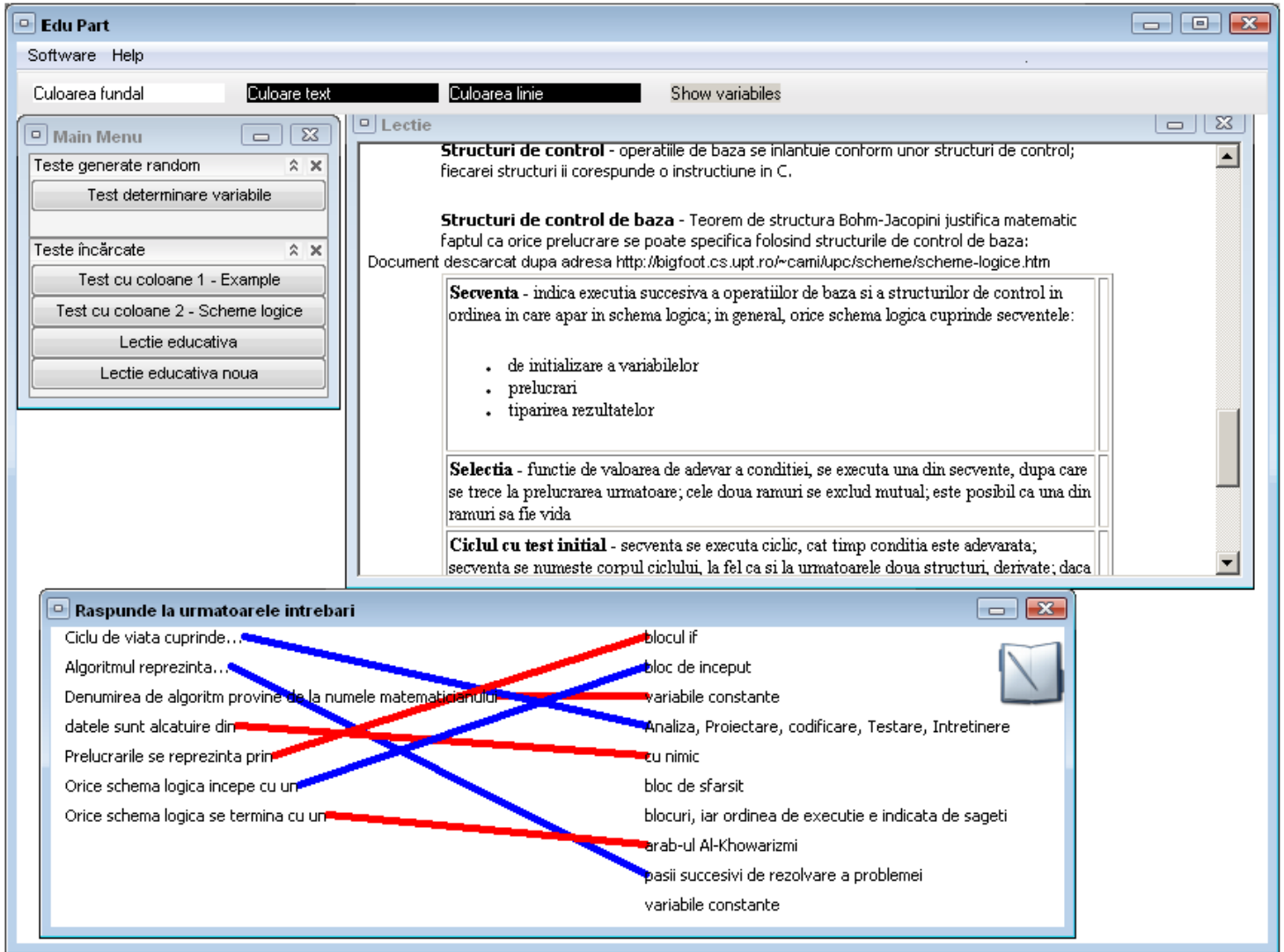
Al doilea test , se genereaza un schematic random iar apoi elevul trebuie sa il modifice astfel inca valoarea curentului/tensiuni/puteri intr-un anumit nod sa fie egala cu o valoare. Part-urile sunt asezate aleator, iar elevul trebuie sa le repositioneze.



Softul genereaza un tip de test educational in care se genereaza o schema electrica aleator, iar elevul trebuie sa o reaseze, si este obligat sa raspunda la intrebari despre comportarea circuitului. Elevul completeaza scriind de mana, rezultatul, softul foloseste un OCR(scris de mine) pentru detectia cifrelor, iar rezultatul este vorbit.

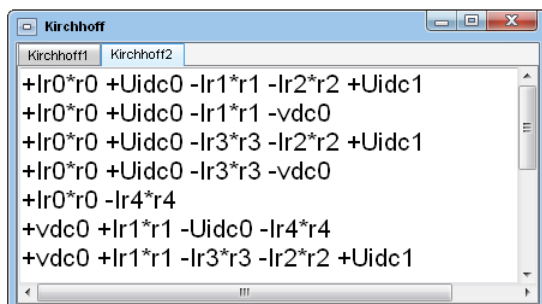
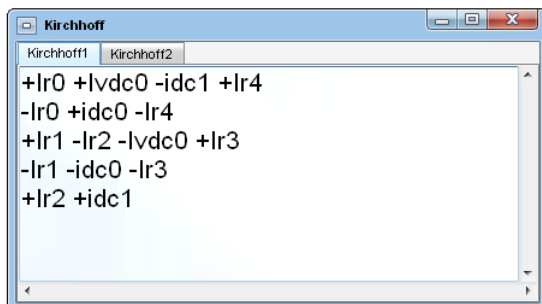
Un alt exemplu utilizand Multi-Layer-Perceptron, a fost un OCR mai "real". Se doreste ca intr-o suprafata de dimensiuni de  $128 \times 128$ . Daca noi aveam o imagine de  $128 \times 128 = 16384$  de pixeli, si ar trebui ca stratul de intrare sa fie format din toti pixeli. Sunt foarte multi pixeli, si s-a optat pe o reducere a imaginii. Informatia pentru calculatoare este aproape aceeasi(informatie redundanta) atat in matricea de  $128 \times 128$  cat si intr-o matrice de  $10 \times 10$ (pentru calculator se pierde foarte putina informatie din imaginea mare in imaginea de  $10 \times 10$ ). Din suprafata desena este face un Regional of Interes(ROI) si dupa aceea se scaleaza la dim de  $10 \times 10$

Profesorul poate crea teste(raspunsuri cu coloana) si lectii cu fisier(fisiere suportate de browser IE, deci flv, html, pdf, doc, xml). Profesorul scrie un fisier XML, care softul il parseaza, si creeaza lectia. Deasemenea, se pot descarca lectii cu update-ul aplicatiei. Logicus reprezinta o viziune noua, asupra softurilor educationale.

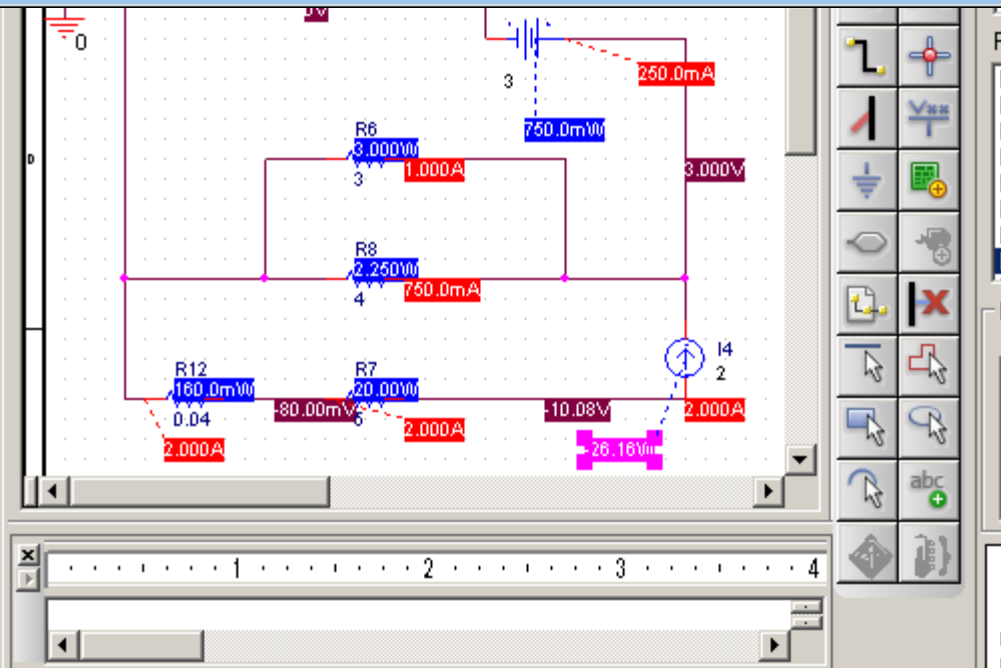
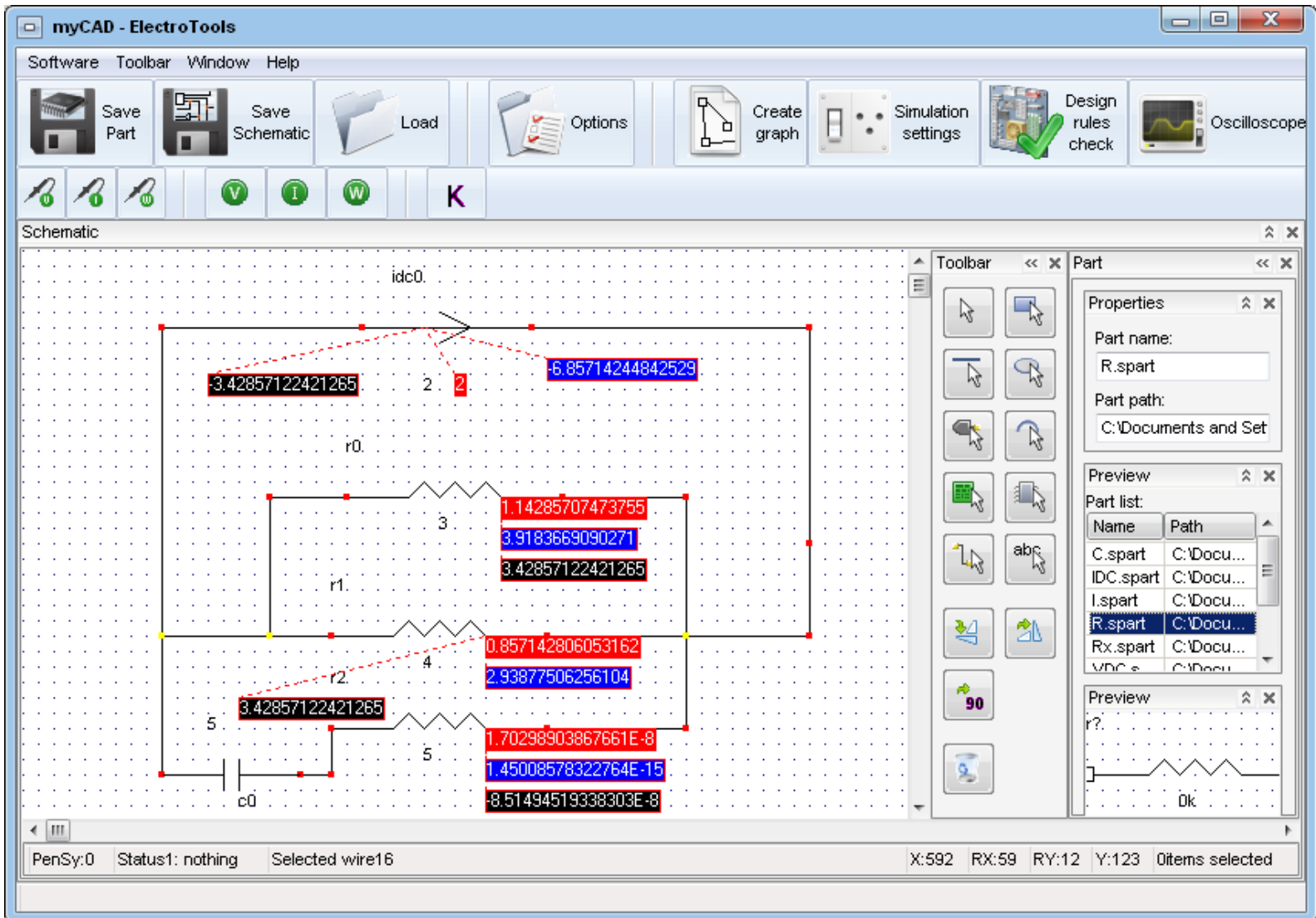


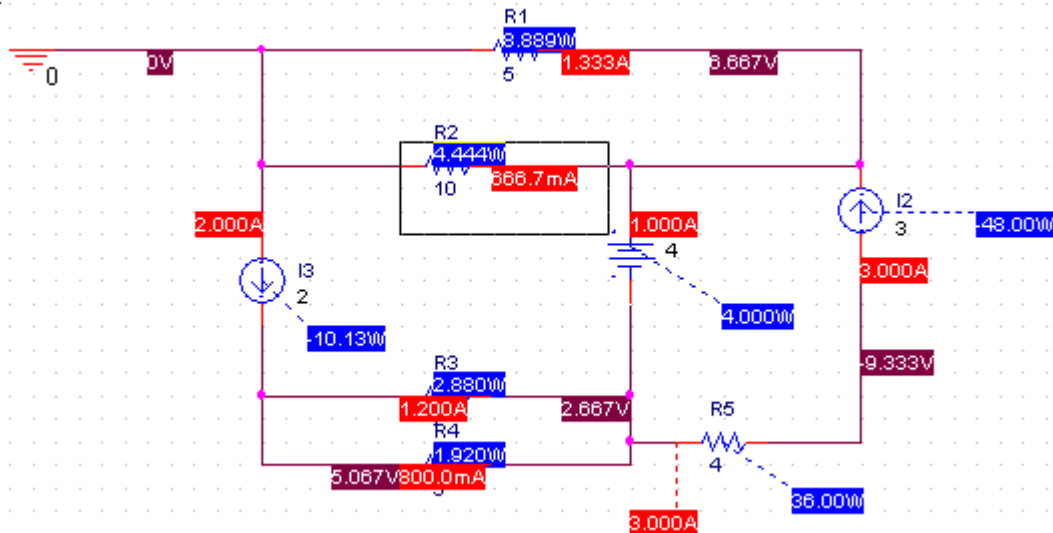
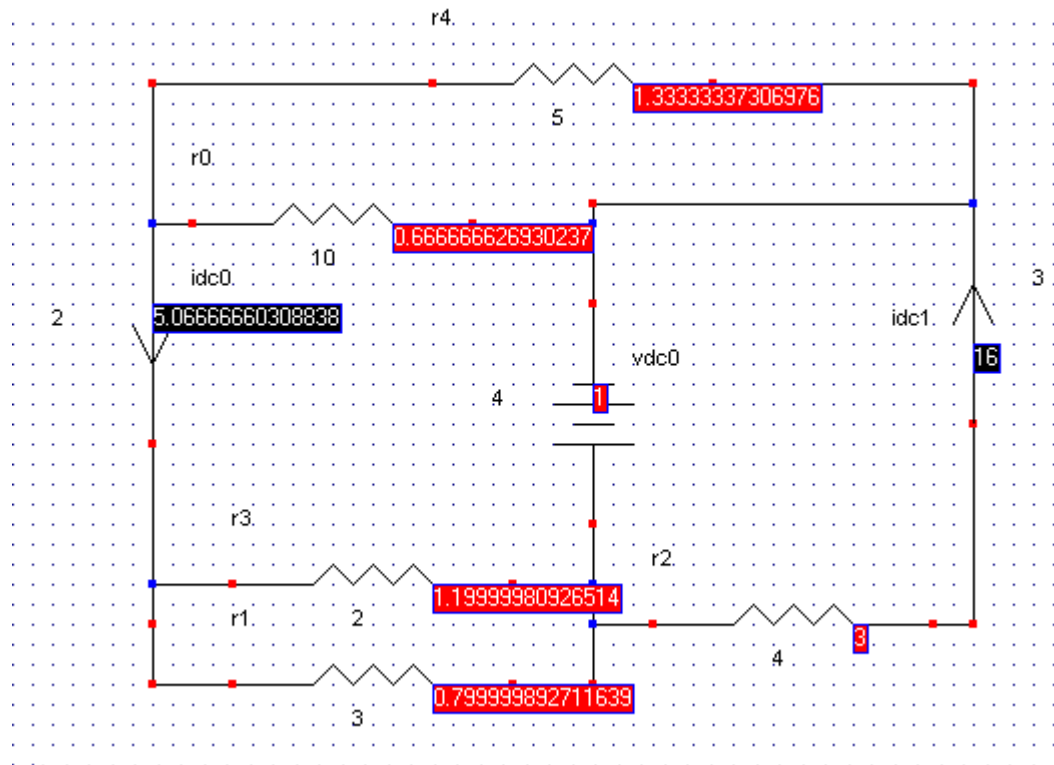
Deasemeni softul permite vizualizarea sistemelor de ecuatii Kirchhoff1 si Kirchhoff2 Pentru orice schma editata.

K



Pentru testarea softului am simulat in PSpice din Cadence OrCAD 16.01 toate schemele create, si au fost diferite de raspuns doar la 5-a zecimala, ceea ce sugerata ca rezolvarile mele sunt corecte, pentru testare am creat in jur de 10 scheme, de utilizeaza intre 5 si 10 parturi. Au fost simulate atat in AC cat si DC





Cadence OrCAD

Soft-ul genereaza pentru utilizator in mod text si teoremele lui Kirchhoff1 si Kirchhoff2 pentru a puea fi studiate de catre elev aceasta interpreteaza ecuatiile literal cat si solutiile ecuatiile, putand experimenta teorema 1 pentru toate nodurile circuitului avand si valorile numerice ale teoremelor deasemenea poate studia si teorema lui Kirchhoff2 avand generate pentru toate ochiurile teoremei. Aceaste facilitati creeaza marea putere educationala a softului, prin generarea si rezolvarea oricarei instante a problemei.

Softul este in totalitate scris in C++, folosind Borland C++ Builder este prima mea lucrare in c++ eu find programator de Delphi-pascal. S-au importat 2 clase care sunt scrise in Pascal si importate in C++. Initial am folosit Prolog, dar pe parcurs am scos. Pentru interfata se foloseste BusinessSkin.

Pentru inteligenta artificiala s-a scris in limbajul orientat spre AI, AILab care este deasemenea scris de mine, si anume Retelele neuronale, bazele de cunostinte.

Interpretarea limbajului uman este scris manual in C++, nefolosind AILab

### Pluginuri

Soft-ul permite adaugare de pluginuri in format DLL. Ele se instaleaza si se dezintaleaza. Exista un mic API pentru a putea lucra cu ele, pentru a putea transmite date atat de la aplicatie la plugin, cat si invers. Momentan este doar un simplu plugin, care sterge schematicul.

### Formatul XML

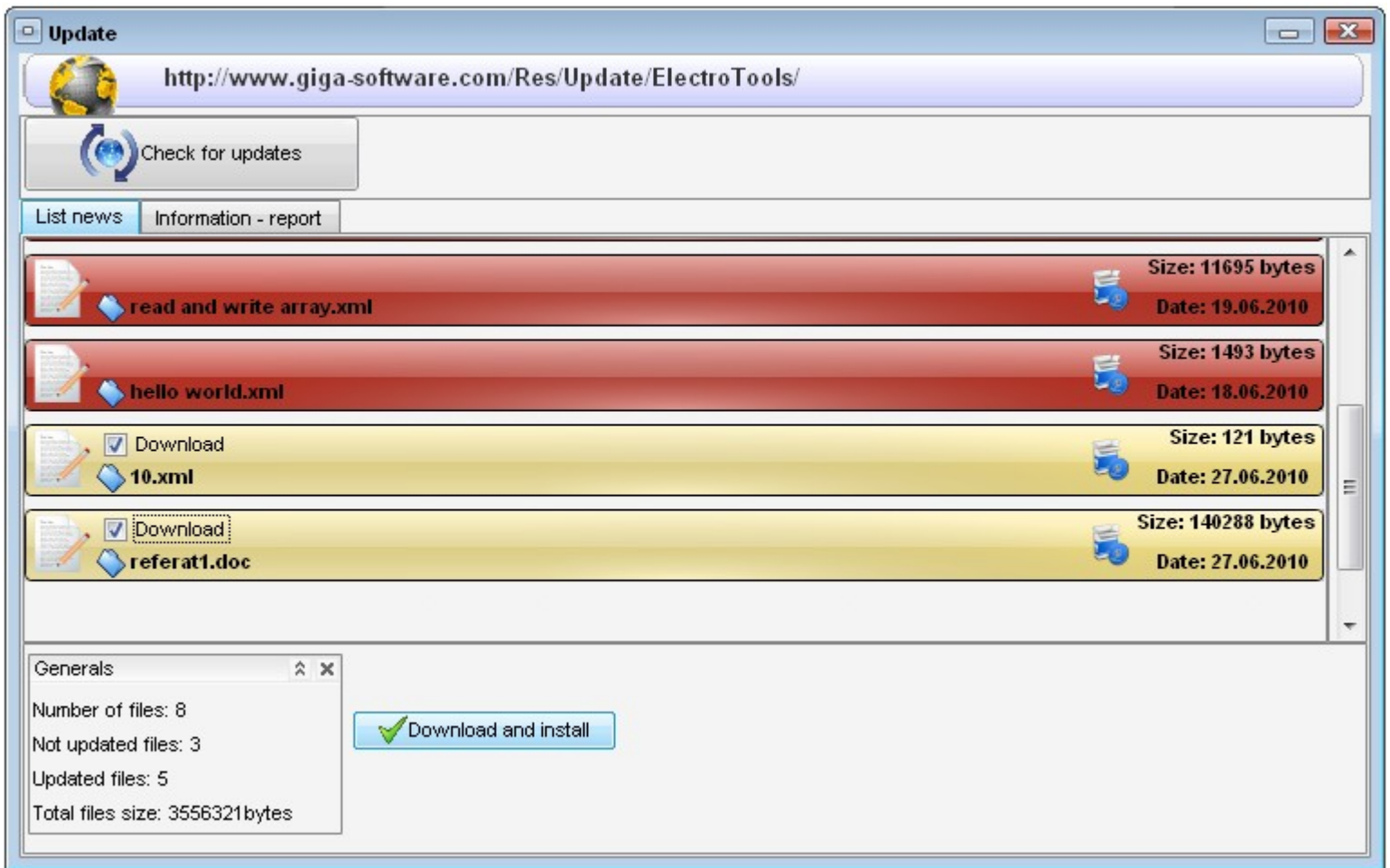
Schemele sunt salvate si importate din format XML. Parsele au fost scrise manual de catre mine.

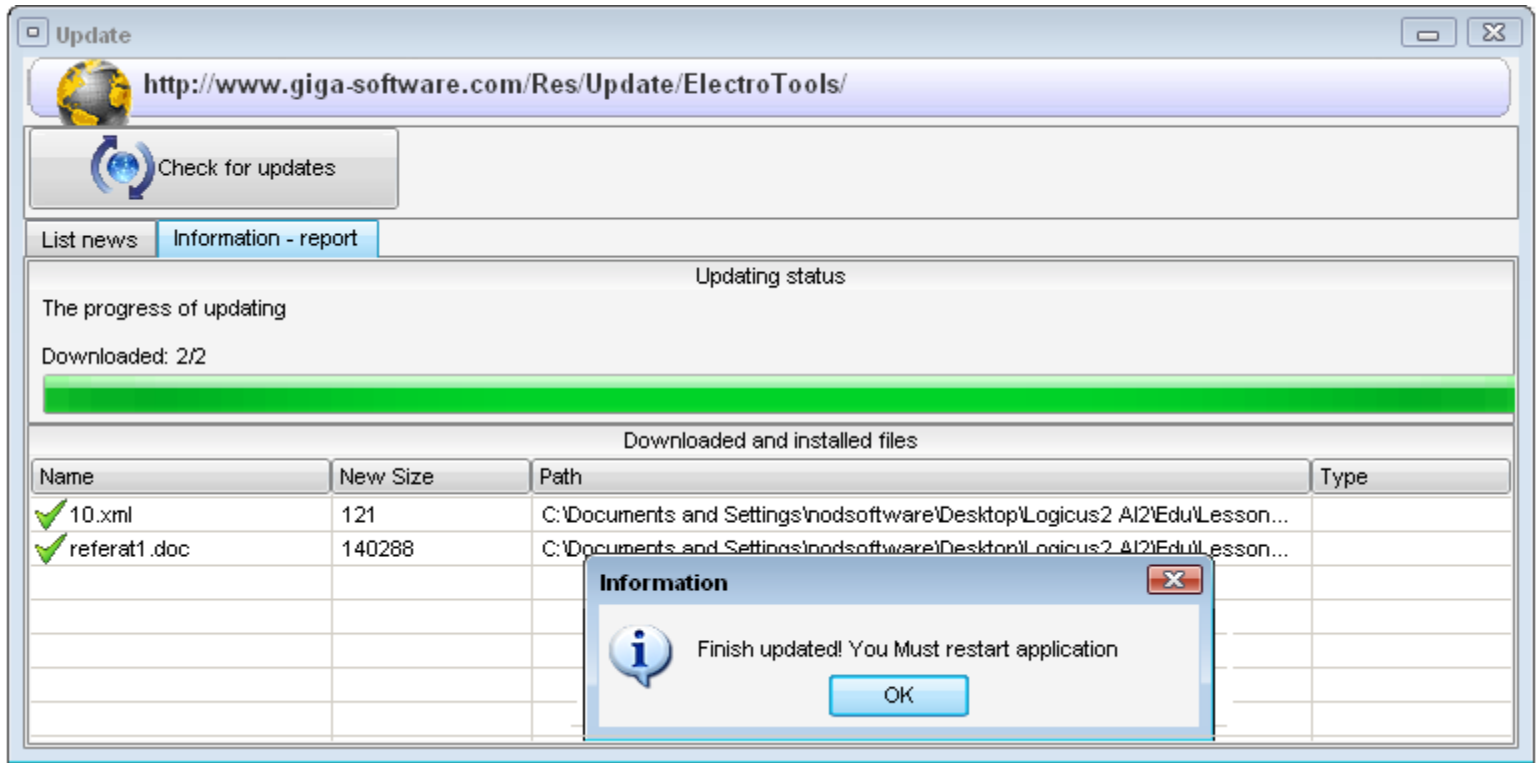
### Clipboard

Se pot pune part-uri in clipboard, data din clipboard ese deasemenea in format XML in ideea de a putea fii importata in alte aplicatii in dezvoltarea viitoare.

### Update

Deoarece, visul oricarui dezvoltator, a unei aplicatii sa pastreze legatura cu utilizatorul , am introdus, un mic update, nu numai la executabile, cat si la examples. Astfel, incat, sa pot publica utilizatorului, o gama cat mai larga de exemple, pentru utilizarea softului.





Structuri de date folosite si tehnici de programare:

1. Recursivitate
2. Teoria Grafurilor
3. Divide et impera
4. Graphics Device Interface
5. Buffere duble
6. Retele neuronale
7. Prolog
8. Pluginuri
9. XML
10. Reprezentarea bazelor de cunostinte bazate pe mostenire

Conform teoriei ingineriei programarii, softurile gen OrCAD fac parte din categoria softuri imposibile sunt scrise de echipe de programatori pluridisciplinari, programatori, automatisti, electronisti, si o realizare individuala la nivelul lui PSpice este practic o problema nerealizabila. Pentru mine realizarea acestui software m-a facut sa inteleg dificultatea conceperii unui soft dintr-un alt domeniu de activitate, practic de aceeași problema se loveste și acela care scrie un soft de medicina: are 2 posibilitati. 1) sa devina medic iar poi sa scrie softul, 2)sa colaboreze cu un medic care il va face sa inteleaga ceea ce doreste si el sa scrie efectiv solicitarile utilizatorului. Ca sa scriem un soft gen OrCAD trebuie sa fii fizician, automatist, electronist, si nu in ultimul rand informatician.



Bibliografie:

- **Cadence PSpice Help.**
- Fizica manual clasa a X-a.
- Rețele neuronale. Aplicații de Catalin-Daniel Căleanu și Virgil Tiponut.
- Rețele neuronale - Arhitecturi și algoritmi de Ștefan Holban
- Învățarea rețelelor neuronale pe bază de exemple de Calin Enăchescu
- Nicoale Tandăreanu, Reprezentarea Bazelor de cunoștințe și interpretarea limbajelor umane, Craiova.
- Nicoale Tandăreanu, Sisteme Expert, Craiova
- Sisteme expert cu Prolog – Constantin Sambotin.
- Sisteme expert Dorin Zaharie UVT
- ACM Digital Library - Natural Language Processing
- ACM Digital Library – Knowledge Base

For further queries related to this project contact us through the Giga-Software website at [www.giga-software.com](http://www.giga-software.com) or you can e-mail us at: [ibudisteanu@acm.org](mailto:ibudisteanu@acm.org) , [ibudisteanu@giga-software.com](mailto:ibudisteanu@giga-software.com)

2010-2011 **Ionut Alexandru Budisteanu**  
Grup Scolar Oltchim

[www.giga-software.com](http://www.giga-software.com)  
[www.seewithtongue.info](http://www.seewithtongue.info)

[ibudisteanu@acm.org](mailto:ibudisteanu@acm.org)

Ferdinand Street no 22, Ramnicu Valcea, 240156, Valcea, Romania