

**Universitatea Politehnica București
Facultatea de Automatică și Calculatoare**



PROIECT DE DIPLOMĂ

-COMANDĂ VOCALĂ KEPHERA III-

Coordonatori Științifici:

Prof. Dr. Ing. Simona Iuliana Caramihai

Sl. Dr. Ing. Calin Aurel Munteanu

As. Prof. Drd. Ing. Mihnea Moiescu

Absolvent:
George Popescu

Bucuresti 2009

Cuprins

Introducere	4
1 Robotul Kephera III	5
1.1 Robotul Kephera III - caracteristici	5
1.1.1 Descriere generala.....	5
1.1.2 Senzorii de proximitate infrarosu	10
1.1.3 Senzorii ultrasonici	11
1.1.4 Bateria.....	12
1.1.5 Alimentarea cu energie	12
1.2 Conexiuni	12
1.2.1 Configurare pentru alimentarea bateriei.....	12
1.2.2 Configurarea comunicatiei Robot-Computer.....	13
1.2.3 Configurarea comunicatiei Robot-Computer cu KoreBotLE	13
1.2.4 Comunicatia Bluetooth	14
1.3 Modul de comunicatie seriala	16
1.3.1 Testarea legaturii seriale primare.....	16
1.3.2 Testarea legaturii seriale secundare	16
1.3.3 Protocolul de comunicatie seriala	17
1.4 Programarea KoreBotLE	18
1.5 Protocolul de Comunicatie	19
2. Comanda asistata a robotului KIII posibilitati de comunicatie	25
2.1 Interfata de instalare Kephera III	25
2.2 Comunicatia Robot – Laptop prin intermediul interfetei USB	30
2.3 Comunicatia Bluetooth	32
2.4 Comunicatie Bluetooth Kephera III – Fujitsu-Siemens Pocket LOOX T830	39
2.5 Control la distanta Kephera III	42
2.6 Interfata Activa Kephera III	50
3 Prezentarea problemei: obiectivul de conducere, mod de abordare, meniul de comenzi	58
3.1 Generalitati	58
3.2 Arhitectura	59
3.2.1 Conexiunea prin intermediul microfonului	65
3.2.2 Modulul de recunoastere a comenzii	65
3.2.3 Analiza si modelarea limbajului	66
3.2.4 Modulul de comanda (dialog) intre utilizator si Kephera III	67
3.2.5 Generarea limbajului	67
3.2.6 Sintetizator de voce	68

3.3 Dialogul dintre utilizatorul uman si Kephera III	68
3.4 Evaluarea sistemului	69
3.4.1 Capabilitati.....	69
3.4.2 Performante.....	69
3.4.3 Instalarea sistemului.....	70
3.5 Recunoastere vocala si retele neuronale. Logica Fuzzy si Algoritmi genetici	73
3.6 Definirea unui profil de utilizator pentru comanda vocala	77
3.7 Editorul audio WavePad v. 4.03 – NCH Software	80
3.8 CoolSoft SpeakToText 2.51	90
3.9 Aplicatia de comunicatie seriala COM Kephera III	95
3.10 Concluzii si directii viitoare de cercetare.....	96
4 Implementare: mediu software de comanda vocala pentru Kephera III	98
4.1 Aplicatie de comunicatie seriala COM Kephera III	98
4.2 Circular optim incet	101
4.3 Inainte optim	101
4.4 Inapoi optim	102
4.5 Stop	102
4.6 Inainte incet	103
4.7 Inainte rapid	104
4.8 Conectare automata la portul COM (6).....	104
4.9 Deconectare de la portul COM (6).....	106
4.10 Programul pentru recunoastere voce.....	107
4.11 Un traseu creat de utilizator	109
4.12 Aplicatia de control si comanda vocala Kephera III.....	111
5 Bibliografie	116

INTRODUCERE

Lucrarea își propune realizarea unei aplicații de comandă vocală care să permită interacțiunea, la un nivel de acuratețe cât mai ridicat, dintre un utilizator uman și robotul Kephera III.

Comenzile date către robot vor fi cele de mișcare pentru fiecare inițiere dorită (stânga, dreapta, înainte, înapoi, cu o anumită viteză – presetată) însă, folosind o aplicație de comandă mai „generală” utilizatorul poate afla și alte informații culese de către robot precum: valorile înregistrate de senzorii ultrasonici, de cei infraroșii, starea bateriei, poziția curentă, viteza curentă, etc.

Obiectivul lucrării este asadar comandă vocală a robotului Kephera III.

Acest obiectiv va fi realizat în prezenta lucrare considerând multiple aplicații de comandă și control vocal construite în mediul de programare Microsoft Visual Basic 2008. Pașii pentru recunoașterea comenzii vocale precum și identificarea vocii utilizatorului împreună cu arhitectura generală pentru comandă vocală vor fi descriși în capitolele următoare, precum și o scurtă caracterizare a robotului.

Comandă vocală pentru Kephera III va fi realizată prin intermediul tehnologiei Bluetooth (control la distanță) considerând o analiză detaliată a specificațiilor robotului și tipurile de comenzi admise de către acesta.

Cu mult timp în urmă (istoricii nu au oferit o dată precisă însă informațiile culese se referă la secolul al X-lea) într-o țară foarte îndepărtată (care acoperea teritoriul Danemarcei de astăzi precum și unele zone ale Norvegiei) locuia un rege Viking care este în principal amintit pentru convertirea întregii populații la o religie străină, creștinismul. Numele său era Harald Bluetooth, fiul lui Gorm cel Bătrân, iar el este acela care a unificat cele mai multe triburi Vikinge înainte ca fiul său, Sven Forkbeard să-l trimită în Valhalla și să preia fraurile puterii.

Cu mai puțin de 1000 de ani mai târziu, atentând la mandria Scandinavă, Ericsson, gigantul Suedez de telecomunicații a decis să preia numele marelui rege pentru a-și denumi noul standard wireless de comunicație după numele său: Bluetooth. Mai mult, compania a reușit să convingă și alți parteneri precum Nokia, Toshiba, IBM și Intel în crearea unui Grup de Interes Special de faptul că Bluetooth este numele potrivit și împreună au pornit spre cucerirea „aerului” (mediul wireless).

Tema de comandă vocală a fost aleasă în mod natural, ca o continuare a cercetărilor făcute în special în ultimii 2 ani și a lucrărilor prezentate la Sesiunea de Comunicări Științifice Studentești la Universitatea Politehnică București.

Lucrarea va prezenta, în ordine, o scurtă descriere a capacităților robotului Kephera III, posibilitățile de comunicație cu acesta, obiectivul de conducere, arhitectura considerată, comenzile introduse, precum și sursele (codul în limbaj de programare) utilizate pentru fiecare tip de mișcare în parte.

Am găsit această problemă deosebit de interesantă în pregătirea lucrării mele de licență în urma cercetărilor făcute și a surselor citate în special datorită progreselor realizate în ultimii ani în domeniul recunoașterii comenzii vocale și a extinderii comenzii vocale în interacțiunea dintre utilizatorul uman și computer și, ca o dezvoltare ulterioară, aceea dintre utilizatorul uman și roboti prin intermediul computerului.

As dori să mulțumesc doamnei profesoare Simona Caramihai care m-a susținut permanent și a crezut în mine în privința acestui proiect și nu numai. Sunt recunoscător pentru sprijinul venit din partea domnilor Mihnea Moisescu și Calin Aurel Munteanu pentru multele sfaturi primite de la dansii precum și link-urile continuând informații deosebit de valoroase pentru această lucrare.

1 ROBOTUL KEPHERA III

1.1 Robotul Kephera III - caracteristici

1.1.1 Descriere generala

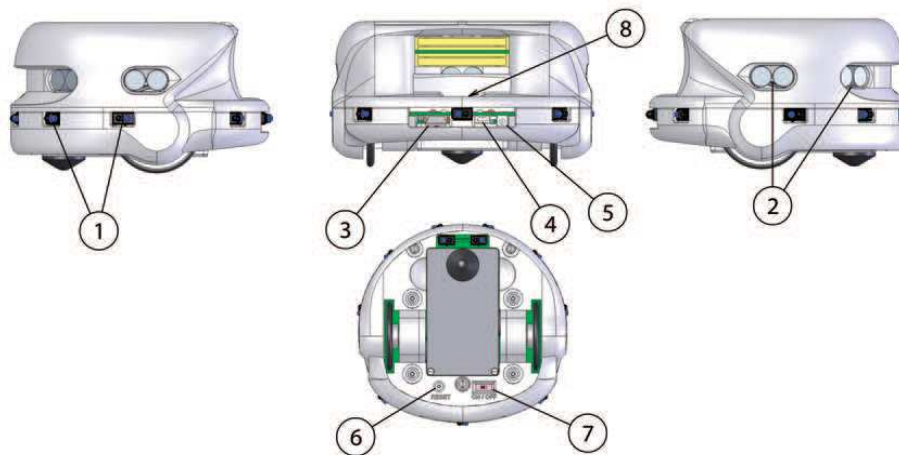


Figura 1.1: Vizualizare generala Kephera III

Robotul are in componente urmatoarele elemente:

1. Senzori infrarosii
2. Senzori ultrasonici
3. Conector serial principal
4. Conector Mini-USB AB
5. Conector jckak pentru alimentare
6. Buton de resetare
7. Buton de pornire/oprire
8. Conector cablu

1.1.1.1 Switch-ul de pornire/oprire a bateriei

Permite utilizatorului sa schimbe starea robotului: pornit / oprit. Cand este in starea "pornit", robotul este alimentat. Robotul poate fi alimentat fara pachetul de alimentare cu ajutorul conectorului de alimentare al bateriei externe.

1.1.1.2 LED-urile de indicare

Robotul Kephera III are in dotare 6 LED-uri de indicare, 2 pentru alimentarea cu energie, unul pentru "pornit", unul pentru starea controler-ului motoarelor si alte 2 programabile de catre utilizator.

George POPESCU – Comandă vocală Kephera III

Cand LED-ul controler-ului de stare al motorului este pornit, unul dintre cei 2 controleri ai motorului se afla in starea de eroare. Aceasta poate avea loc atunci cand unul dintre motoare este blocat sau o limita de curent este atinsa. In acest caz controler-ul trebuie resetat de catre dsPIC (comanda 'M' in modul de interfata seriala sau apelarea functiei 'initKH3()') in programul KoreBotLE)

Pentru incarcator, LED-ul rosu indica faptul ca bateria se incarca iar cel verde semnaleaza faptul ca incarcarea este completa. Daca nu exista un pachet al bateriei la conectarea sursei externe de energie, LED-ul verde se va aprinde.

LED-urile 5 si 6 pot fi setate de catre utilizator cu ajutorul comenzii 'L'.

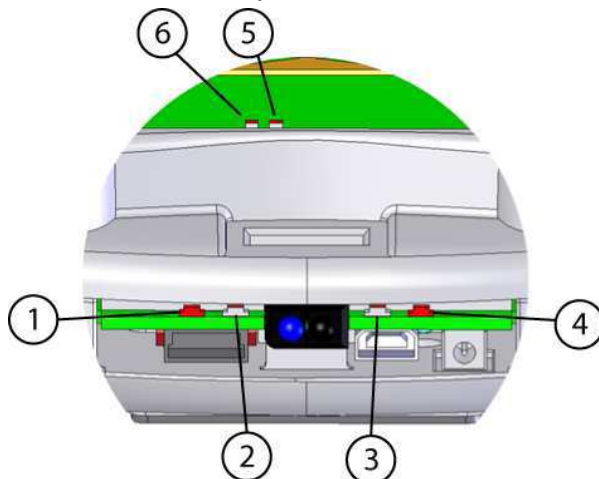


Figura 1.2: LED-urile Kephera III

1. LED-ul Controler-ului de stare pentru motor (LED-ul Rosu)
2. LED-ul pentru pornire
3. LED-ul pentru alimentare completa (verde)
4. LED-ul pentru alimentare in desfasurare (rosu)
5. LED programabil (verde)
6. LED programabil (rosu)

1.1.1.3 Conectorul Serial

Conectorul serial contine diferite linii seriale. O linie seriala care se afla la valoarea de TTL (0V/+5V) este folosita pentru conectarea unui computer personal la dsPIC-ul Kephera III (cu ajutorul unui KoreConnect). Se foloseste doar cand nu este conectat un KoreBotLE.

Exista deasemenea si RS232 (-10V/+10V) folosit pentru comunicatia cu KoreBotLE, i.e. a vedea mesajul de boot-are al KoreBotLE.

Mai este posibila si folosirea unui link USB al KoreConnect pentru comunicatia cu KoreBotLE.

Lungimea cablului serial trebuie sa fie limitata la 2 metri pentru operabilitatea cu succes.

- | | |
|----------------|-----------|
| 1. Korebot RXD | 7. USB D- |
| 2. Korebot TXD | 8. USB D+ |
| 3. dsPIC RXD | 11. GND |
| 4. dsPIC TXD | 12. GND |
| 6. VCC (+5V) | |

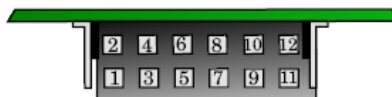


Figura 1.3: Detalii port serial Kephera III

1.1.1.4 Conectorul pentru cablu

Conectorul pentru cablu este un conector serial si de alimentare. Este posibila atat cumincatia cat si alimentarea robotului. Acest conector este util daca se doreste alimentarea robotului de la o sursa de tensiune de +9V si mentinerea mobilitatii robotului cu ajutorul unui cablu. Totusi, daca se doreste realimentarea bateriei in acelasi timp, este obligatoriu sa se foloseasca conectorul jack.

- | | |
|--------------|--------|
| 1. VCC (+9V) | 4. GND |
| 2. dsPIC RXD | 5. GND |
| 3. dsPIC TXD | 6. GND |

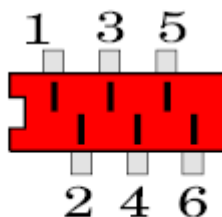


Figura 1.4: Detalii asupra cunectorului

1.1.1.5 Conectorul USB

Conectorul mini-USB este util doar atunci cand KoreBotLE este adaugat la KepheraIII. Exista posibilitatea de conectare directa la PC prin Kephera III cu ajutorul unui cablu mini-USB. Pentru comunicatia cu KoreBotLE prin acest conector USB se va consulta documentatia existenta pe www.k-team.com.

1.1.1.6 Cum se poate adauga KoreBotLE

Kephera III a fost construit pentru a fi folosit impreuna cu KoreBotLE, dar acest lucru nu este obligatoriu. In pachetul initial nu exista acest KoreBotLE. Acesta poate fi achizitionat si montat in robot. Pentru conectarea ulterioara a acestuia la robot este necesara inlaturarea carcasei initiale.

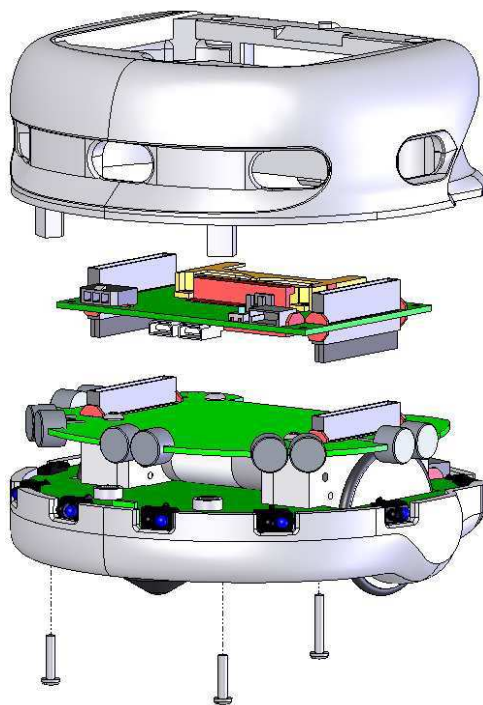


Figura 1.5: Detalii asupra ansamblului KoreBotLE

1.1.1.7 Motoarele si controlul lor

Fiecare roata este miscata de catre un motor DC cuplat la aceasta prin intermediul unei reduceri 43.2:1. Motorul insusi are o reducere 27:1 iar cutia de viteze are o reducere de 1.6:1. Motorul are propriul sau encoder incremental, aflat pe axa motorului, care ofera 16 pulsatii la o rotatie completa a motorului. Aceasta permite o rezolutie de 691.2 pe o rotatie a motorului care corespunde la 54 de pulsatii pe 10 milimetrii parcursi ai robotului (diametrul rotii = 41 mm, astfel fiecare rotatie a robotului inseamna 128.8 mm). Initial robotul se afla la o rezolutie de 4 ori mai mare a encoder-ului. Astfel, in acest caz, pentru fiecare rotatie completa a rotii controler-ul motorului face 2764 masuratori. Configuratia celuiilalt encoder este 2x (1382 masuratori / miscare).

Observatie: 1 rotatie = 128.8 mm = 2764 masuratori.

Fiecare motor este condus de catre propriul sau controler implementat in PIC18F4431. PIC-ul are control direct pentru puterea motorului prin intermediul unui bridge dublu H si poate citi pulsatiile encoder-ului incremental.

Blocul de control al motorului se comporta ca si dispozitiv "slave" pe magistrala IC2. Cand KoreBotLE nu este conectat la robotul Kephera III, CPU-ul robotului principal va fi i2c "master". Cand KoreBotLE este conectat, acesta devine i2c "master".

Fiecare controler al motorului isi schimba valoarea "pornit" / "oprit" la o anumita frecventa intr-un anumit interval de timp. In acest fel, motorul reactioneaza la timpul mediu de alimentare, care poate fi schimbat prin modificarea perioadei in care motorul este "pornit". Aceasta inseamna ca numai partea dintre perioadele de "pornire" so "oprire" este modificata, precum ilustrata in figura. Metoda de control a alimentarii este numita "modulatia pulsatiei". Valoarea acesteia este definita ca timpul in care motorul este "pornit".

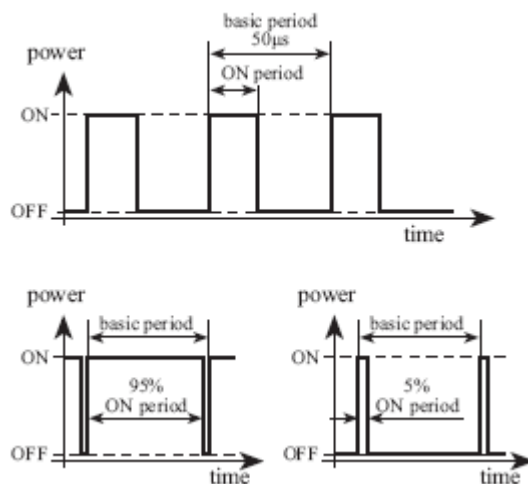


Figura 1.6: Modulatia impulsurilor

Fiecare controler al motorului poate controla viteza si pozitia fiecarui motor, stabilind valoarea de "modulatie a pulsatiei" conform cu viteza reala sau pozitia citita de catre encoder-ii incrementali.

Ambele motoare DC pot fi controlate de catre un controler PID. Fiecare termen al acestui controler (Proportional, Integral, Derivativ) este asociat unei constante, stabilind impactul fiecărei componente: Kp pentru proportional, Ki pentru integral, Kd pentru derivativ.

Controlerul motorului poate fi folosit in doua moduri de control: pentru viteza si pentru pozitie. Modul activ de control este stabilit conform cu cumanda primita. Daca controler-ul primeste o comanda pentru viteza, atunci acesta schimba viteza robotului. Daca controler-ul primeste o comanda pentru controlul pozitiei, modul de control este automat schimbat pentru modul de pozitie. Diferite valori ale parametrilor (Kp, Ki si Kd) pot fi setati pentru fiecare din modurile de control.

1.1.1.8 Viteza

Folosit in modul pentru viteza, controler-ul are ca input o valoare pentru viteza rotilor si controleaza motorul pentru a mentine aceasta valoare.

Valoarea pentru viteza reprezinta diviziunea unei valori constante la timpul intre pulsatiile encoder-ului. In modul default (rezolutia encoder-ului x4 si raport 1:4), o masuratoare este facuta de 4 ori la fiecare pulsatie. Apoi fiecare rotatie a rotii va avea 2764 masuratori. Valoarea constanta este definita de timpul maxim inmultit cu 256 ($0xFFFF * 256 = 16'776'960$). Aceasta operatie permite un calcul mai bun PID pentru o viteza mai mica.

$$VitezaMotorului = \frac{16'776'960}{Timer5value}$$

Pentru a converti aceasta valoare in timp real, se foloseste urmatorul calcul (de aceasta data este intarzierea intre cele 2 masuratori):

$$Timpul = \frac{\frac{Timer5value}{Postscaler}}{\frac{f_{osc} / 4}{Tmr5Pr escaler}}$$

, unde $f_{osc} = 20$ MHz si $Tmr5Prescaler = 8$ (initial).

Pentru a calcula viteza reala in mm/s:

$$Vitezareala = \frac{CircumferinaRotii}{Timpul * 2764}$$

, unde $CircumferintaRotii$ este 128.8 mm si 2764 corespunde numarului de masuratori pe rotatie al rotii.

Iata un exemplu cu $VitezaMotorului=20'000$, $Tmr5Prescaler=8$, $Postscaler=4$ si rezolutia encoder-ului= $x4$

$$Timer5value = \frac{16'776'960}{20'000} = 839$$

$$Timpul = \frac{\frac{839}{4}}{\frac{20'000'000 / 4}{8}} = 0.336[ms]$$

$$Vitezareala = \frac{128.8}{0.000336 * 2764} = 138.9[mm / s]$$

$$Vitezareala = \frac{VitezaMotorului}{VitezaK} [mm / s]$$

, unde $VitezaK$ este 144.01 cand este folosita configuratia initiala.

Viteza maxima de atins este 48'000 (= 333 mm/s) in bucla deschisa si 43'000 (= 298 mm/s) in modul regulat. Iar minimul este de 2'000 (= 13.9 mm/s) cu control regulat.

Folosit in modul de control al pozitiei, controler-ul are ca input pozitia rotii, o acceleratie si o viteza maximala. Folosind aceste valori, controler-ul accelereaza roata pana ce acesta atinge viteza maxima si o decelereaza pentru a atinge pozitia tinta. Aceasta miscare urmeaza un profil trapezoidal al vitezei, precum in figura de mai jos.

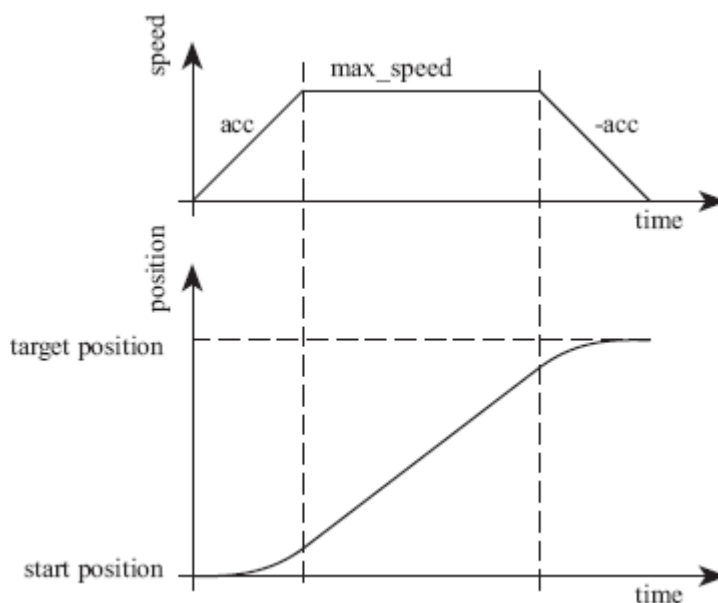


Figura 1.7: Profilul vitezei pentru a se atinge o pozitie dorita

Valorile de intrare co modul de control al acestui controler pot fi modificate in orice moment. Controler-ul va update si executa noul profil al modului de pozitie sau va controla viteza rotii urmand valoarea noua pentru modul de viteza.

Mai intai se va configura profilul de viteza (comanda 'J') cu parametrii necesari. Apoi se poate inregistra o miscare cu profilul modului de pozitie (comanda 'F').

1.1.2 Senzorii de proximitate infrarosu

Kephera III are 9 senzori aflatii in jurul robotului si 2 plasati sub acesta. Ultimii permit experimente si teste precum urmarirea unei linii.

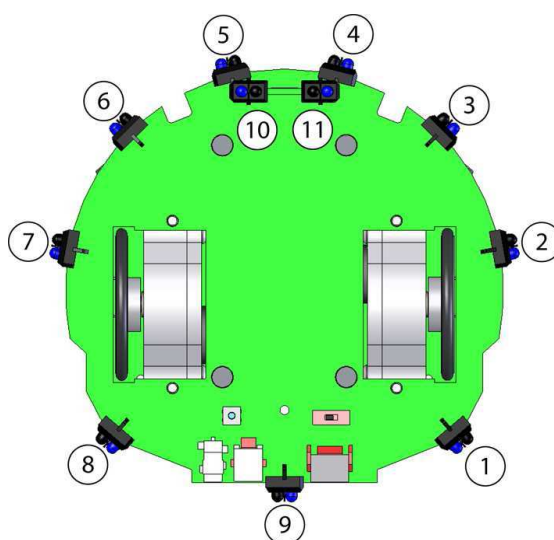


Figura 1.8: Senzorii infrarosu

Acesti senzori contin un emitor si un receptor infrarosu. Cei 8 senzori sunt TCRT5000 – senzori optici reflectivi produsii Vishay Telefunken.

Acest dispozitiv senzor permite 2 masuratori:

- al luminii normale ambientale. Aceasta masuratoare este realizata folosind doar receptorul, fara a emite lumina prin emitor. O noua masuratoare este realizata la fiecare 33ms. In timpul celor 33ms cei 11 senzori sunt cititi intr-o ordine secventiala la fiecare 3ms. Valoarea returnata la un anumit interval de timp este rezultatul ultimei masuratori efectuate.
- a luminii reflectate de catre obstacole (= proximitate). Aceasta masuratoare este realizata prin emiterea de lumina folosind partea emitoare a dispozitivului. Valoarea returnata este diferenta dintre masuratoare realizata prin emiterea de lumina si lumina masurata fara emisie de lumina (lumina ambientală). In timpul celor 33 de ms cei 11 senzori sunt cititi intr-o ordine secventiala la fiecare 3 ms. Valoarea returnata la un anumit interval de timp este rezultatul ultimei masuratori efectuate.

lesirile fiecarei masuratori sunt date de valoarea analogica convertita la o valoare pe 12 biti. Urmatoarele 2 sectiuni ilustreaza insemnatate acestei valori pe 12 biti.

1.1.2.1 Masuratori asupra luminii ambientale

Masuratorile asupra luminii ambientale sunt puternic influentate de catre mediul in care se afla robotul. Depinzand de tipul sursei de lumina, culoare si distanta, profilul masuratorilor luminii ambientale poate varia. Nu este recomandata folosirea unei surse de lumina cu o emisie mare intr-o gama infrarosu larga deoarece acesta poate induce in confuzie senzorii infrarosu.

1.1.2.2 Masuratori asupra luminii reflectate (proximitate)

Senzorii sunt in principal folositi cu scopul de a detecta obstacolele din jurul robotului KepheraIII.

Masuratorile asupra luminii reflectate depinde de reflexivitatea obiectului si de conditiile ambientale de lumina. Culoarea, materialul suprafetelor obiectului **au o influenta** asupra raspunsului senzorial. Mai mult, ca orice senzor, senzorii infrarosii sunt sensibili la zgomotul mediului inconjurator. Pentru toate aceste motive, graficul urmat este oferit cu scopul informarii si nu ar trebui considerate ca si referinte.

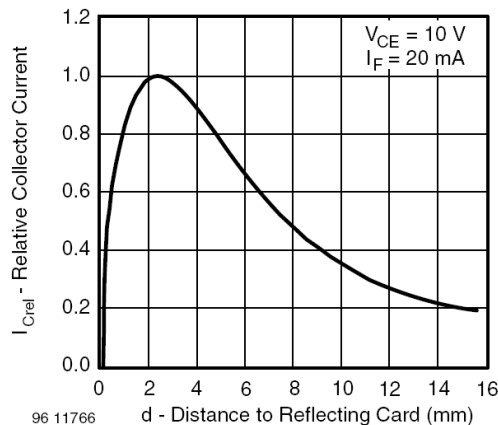


Figura 1.9: Curentul colector in comparatie cu distanta

1.1.3 Senzorii ultrasonici

In setul de baza, 5 senzori sunt plasati in jurul robotului si sunt pozitionati si numerotati precum in figura urmatoare.

5 senzori sunt, de fapt, 5 perechi de dispozitive ultrasonice unde fiecare pereche este compusa dintr-un emitor si un receptor.

Senzorul ultrasonic este alimentat de la o sursa dc de 20V. Frecventa nominala a acestor traductoare este de 40kHz +/- 1kHz.

Parametrii precum numarul maxim de ecouri, timpul mort si senzorii activi sunt **parametrizabili prin intermediul unui api de configurare** (comanda 'C'). Initial numarul maxim de ecouri este de 3, senzorul 3 (cel din fata) este activ, timpul mort este setat pentru a vedea intre 20cm si 4 metri.

Ultimul parametru este daca exista montat corpul superior sau nu (initial montat). Camp corpul superior este montat exista un oarecare zgomot datorat ecoului interior, insa acest zgomot este sters prin aplicatii software. De fapt, se poate imbunatati detectia celor mai apropiate obstacole (20cm pana la 40cm) prin detasarea corpului superior.

Fiecare masuratoare returneaza numarul de ecouri gasite, distanta in cm a fiecarui ecou, amplitudinea fiecaruia si timpul (stampila temporală) de cand ecoul a fost auzit.

Valoarea returnata de catre comanda este zgomotul alb masurat inainte ca un puls ultrasonic sa fie trimis. Apoi amplitudinea reala a fiecarui ecou va fi amplitudinea acestuia minus zgomotul alb (comanda 'G').

Pentru mai multe detalii despre senzorul ultrasonic exista referinte pentru componenta Midas, receptorul este 400ST100 iar receptorul este 400SR100.



Figura 1.10: Pozitia celor 5 perechi de senzori ultrasonici

1.1.4 Bateria

Kephera III este echipat cu o baterie compusa din doua elemente polimerice de tip Li-Ion. Acestea produc o tensiune de 7.4V cu o capacitatea a curentului de 1400 mAh. Folosind puterea sa inmagazinata, robotul este capabil sa functioneze complet autonom pe perioada a mai mult de 4 ore, folosind o configuratie de baza. Cand se adauga un echipament auxiliar, autonomia este redusa dupa cum extensiile Kephera III precum KoreBotLE se bazeaza pe bateriile robotului ca sursa de alimentare.

Nu exista un sistem de management al alimentarii pentru Kephera III. Cand voltajul a 2 baterii scade sub 6V bateria se deconecteaza automat pentru a evita o descarcarea prea mare a celulei. Utilizatorii pot implementa propriul lor program software prin care sa realizeze un sistem de management de alimentare pentru ca KoreBotLE sa se inchida corespunzator in acest caz (comanda 'V').

1.1.5 Alimentarea cu energie

Daca este dorita o sursa externa de energie sau se doreste alimentarea cu energie in timp ce bateriile se incarca, alimentarea poate fi asigurata prin intermediul conectorului jack sau prin intermediul conectorului de cablu microMatch.

Este recomandat sa se foloseasca numai incarcatorul oferit de K-Team. Daca se doreste alimentarea prin intermediul unui alt dispozitiv trebuie sa se asigure ca voltajul este de +9V si ca sursa de alimentare poate oferi 2A.

Precautie: Incarcatorul trebuie sa fie conectat la priza dupa ce toate celelalte conexiuni sunt deja realizate¹.

1.2 Conexiuni

1.2.1 Configurare pentru alimentarea bateriei

Pentru incarcarea bateriilor robotului trebuie sa se asigure ca urmatoarele sunt realizate corect:

- Alimentatorul cu energie trebuie conectat la robot.

¹ *Kephera III User Manual*, version 2.2, K-Team, Switzerland, March 2008

- Observatie: bateria robotului poate fi incata atat cu robotul in starea “pornit” cat si in starea “oprit”. Atunci cand robotul se afla in starea “pornit” timpul de incarcare va fi mai lung.
- Alimentatorul trebuie sa fie conectat la o sursa de energie.

Exista 3 LED-uri. Un LED verde marcheaza daca robotul este alimentat sau nu. Si 2 alte LED-uri sunt prezente pentru a arata ca robotul se incarca (LED-ul rosu “aprins”) sau ca incarcarea este completa (LED-ul verde este “aprins”).

In timpul incarcarii LED-ul rosu este “pornit” iar cel verde este “oprit”. Procesul este reversat la sfarsitul procesului de incarcare. Timpul de incarcare pentru o baterie goala este de aproximativ 180 de minute. La acest moment alimentatorul cu energie poate fi deconectat. In timpul incarcarii bateria poate ajunge la valoarea de 40 de grade Celsius.

1.2.2 Configurarea comunicatiei Robot-Computer

Aceasta configuratie permite comunicatia intre robot si un computer prin intermediul unei legaturi seriale. Computerul este legat la modulul de interfata prin linia standard RS232, in timp ce modulul de interfata converteste semnalul RS232 la un semnal de nivel TTL pentru a comunica cu robotul.

Pentru a folosi modul de comunicatie seriala utilizatorul trebuie sa se asigure ca urmatoarele sunt indeplinite corect:

- KoreBotLE nu este conectat la Kephera III.
- Bateria robotului este incarcata ceea ce inseamna ca LED-ul pentru alimentare este “aprins”.
- Robotul trebuie sa fie conectat la un modul KoreConnect folosind cablul serial.
- KoreConnect trebuie conectat la computer folosind cablul serial RS232, in acest fel, cablul trebuie conectat la **conectorul DB9 numarul 2** (figura).
- **Portul Serial trebuie configurat dupa cum urmeaza: 115200bps, 8 Data bits, 1 stop bit, no parity, no hardware control.**

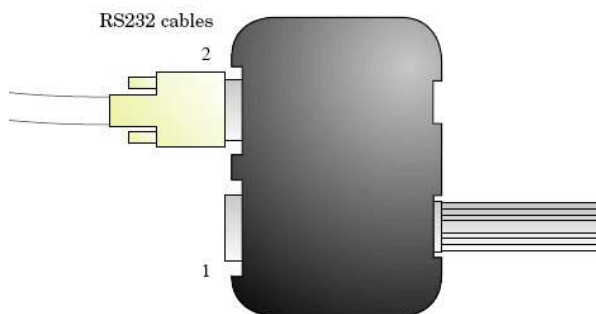


Figura 1.11: Conexiune atunci cand robotul este folosit fara KoreBotLE

1.2.3 Configurarea comunicatiei Robot-Computer cu KoreBotLE

1.2.3.1 Legatura seriala

Aceasta configuratie permite comunicatia intre KoreBotLE montat pe robot si computer prin intermediul unei legaturi seriale. In acest configuratie nu este posibila comunicatia cu robotul baza insusi precum a fost explicat in capitolele precedente. Computerul este legat la modulul de interfata prin intermediul unei linii standard RS 232. Adaptarea RS232/TTL este realizata pe KoreBotLE.

Pentru a folosi modul de comunicatie seriala trebuie sa se asigure ca urmatoarele sunt indeplinite corect:

- KoreBotLE este conectat pe Kephera III.
- Bateria sau un alimentator este montat iar robotul este “pornit”.

- Robotul trebuie conectat la un modul KoreConnect folosind cablul Serial.
- KoreConnect trebuie conectat la computer folosind cablul serial standard RS232. In acest mod, cablul trebuie sa fie conectat la **conectorul DB9 numarul 1** (figura).
- **Portul Serial trebuie configurat dupa cum urmeaza: 115200bps, 8 Data bits**, 1 stop bit, no parity, no hardware control.

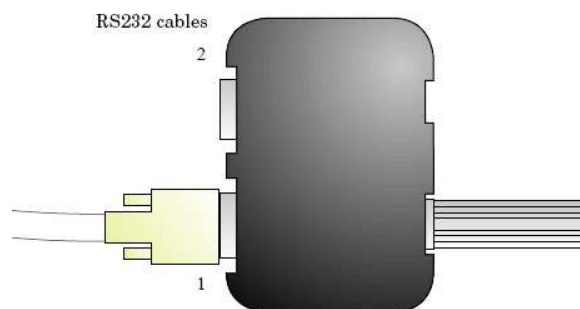


Figura 1.12: Conexiune atunci cand robotul este folosit cu KoreBotLE

1.2.3.2 Legatura USB

Aceasta configuratie permite comunicatia intre KoreBotLE montat pe robot si computer prin intermediul unei legaturi USB. Exista 2 moduri de a conecta USB KoreBotLE. Prin KoreConnect si cablul standard USB sau doar cu ajutorul unui cablu mini-USB.

Pentru a folosi modul de comunicatie USB trebuie sa se asigure ca urmatoarele sunt indeplinite corect:

- KoreBotLE este conectat pe Kephera III.
- Bateria sau un alimentator este montat iar robotul este "pornit".
- Robotul este conectat la modulul KoreConnect si apoi un cablu USB sau un cablu mini-USB este legat direct la conectorul USB miniAB al robotului (figura).
- Cablul USB (standard sau mini) trebuie sa fie conectat la computer.

1.2.4 Comunicatia Bluetooth

Un dispozitiv Bluetooth este montat pe fiecare robot Kephera III cu firmware 2.0 sau mai mare. Dispozitivul este un WT12 de la compania BlueGiga, are licenta I-Wrap si este configurat de comanda AT. Dispozitivul este complet configurat ca inlocuitor de cablu pentru Kephera III². Dar poate fi folosit cu KoreBotLE deasemenea montat pe Kephera III. Apoi conexiunea Bluetooth va fi disponibila via portul serial /dev/tts/1 al KoreBotLE³. Configuratia este urmatoarea:

- 115200bps data rate
- 8 data bits
- no parity
- one stop bit
- **no hardware flow control**
- connection securised (security code is: 0000).

² Referinta pentru comunicatia Bluetooth: Che Fai Yeong, Amin, S.H.M., Faisal, N., Bakar, J.A., „Bluetooth enabled mobile robot”, IEEE International Conference, 2007.

³ Alte porturi de comunicatie seriala Bluetooth: Sidenmark, M., „Blue Id Access System using Bluetooth”, University of Karlskrona, 2006.

1.2.4.1 Comunicatia Bluetooth cu Kephera III

Vom descrie o procedura pentru a stabili comunicatia Bluetooth intre Kephera III si PC precum si validarea conexiunii intr-un terminal (de exemplu pentru Windows XP). Nu este necesara conectarea unui KoreBotLE pe Kephera III in acest caz.

- In primul rand se conecteaza un dispozitiv Bluetooth la PC (daca este cazul).
- Se verifica daca sistemul recunoaste dispozitivul si apoi se instaleaza driver-ul. Daca exista Bluetooth integrat in laptop atunci este necesara doar activarea lui.
- Se verifica daca Kephera III este corect "pornit" (LED-ul verde "aprins").
- Se porneste cautarea dispozitivului prin: "Gasirea dispozitivelor in raza de actiune".
- Kephera III trebuie sa apara cu logo-ul PDA iar numele KHIII Numar Serial. Numarul Serial este acelasi cu numarul afisat pe spatele robotului.
- Crearea unei conexiuni cu Kephera III; programul va cere codul de securitate si va trebui introdus 0000.
- Programul va asigna un port COM dispozitivului.
- Se va deschide un terminal cu portul COM asignat si configuratia corecta de mai sus.
- Se va incerca sa se citeasca valoarea senzorilor infrarosu (sau o alta comanda) pentru a valida comunicatia.
- Apoi se poate folosi orice alt program care foloseste portul COM (precum Matlab) pentru controlul la distanta al robotului.

1.2.4.2 Comunicatia Bluetooth cu KoreBotLE⁴

Procedura va fi aceeasi ca mai sus cu un KoreBotLE montat pe Kephera III.

- Creaza o conexiune Bluetooth cu Kephera III precum a fost explicat mai devreme.
- Pentru a valida comunicatia, conecteaza un KoreBotLE la KoreConnect.
- Logeaza KoreBotLE (login: root, parola: rootme).
- Deschide un minicom pe KoreBotLE (minicom -s /dev/tts/1).
- Configureaza terminalul precum s-a descris mai sus si selecteaza dispozitivul /dev/tts/1.
- Acum KoreBotLE poate trimite/primi informatie la/de la PC via Bluetooth.

1.2.4.3 Logheaza KoreBotLE la o conexiune Bluetooth

Exista o modalitate de a folosi /dev/tts/1 (Portul Serial Bluetooth) pentru a loga KoreBotLE. **Pentru aceasta este nevoie sa se modifice fisierul initial** in directorul /etc al KoreBotLE dupa cum urmeaza:

- Logheaza KoreBotLE cu KoreConnect (login: root, parola: rootme).
- **Deschide fisierele /etc/initab (via /etc/initab).**
- **La sfarsitul fisierelor se va gasi linia: :T0:23:respawn:/sbin/getty-L ttyCo 115200 vt100.**
- **Salveaza si inchide fisierul.** Apoi, **la urmatoarea pornire va fi posibila logarea KoreBotLE prin Bluetooth.**

⁴ Abordari bazate pe "token-uri" sunt studiate in: Sidenmark, M., „Blue Id Access System using Bluetooth”, University of Karlskrona, 2006.

1.3 Modul de comunicare seriala

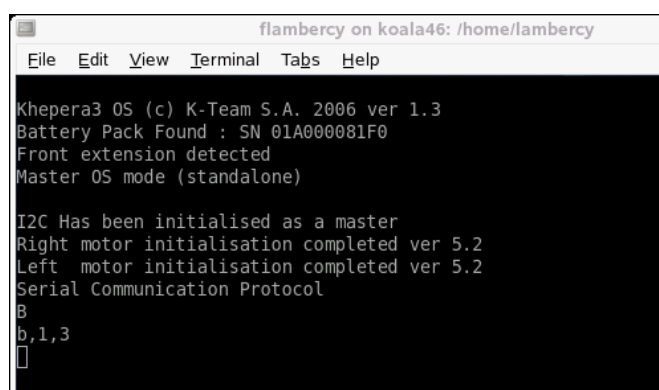
1.3.1 Testarea legaturii seriale primare

Înainte de orice operație de conectare a cablului serial între computer și robot trebuie testată. Pentru a testa comunicarea serială este nevoie de:

- Să se asigure că toate conexiunile sunt corecte.
- Robotul trebuie plasat pe o suprafață plană și sigură. Switch-ul bateriei trebuie să fie "oprit".
- **Un terminal emulator trebuie să pornească pe computer.** Trebuie să se asigure că terminalul este conectat la portul serial corect. **Configurația terminalului trebuie să fie setată la 115200 Baud, 8 bit, 1 start bit, 1 stop bit, no parity și no hardware control.**

Când este pornit, robotul trebuie să trimită un mesaj (figura) către terminalul emulator⁵.

Pentru a testa faptul că Kephera III este capabil să primească comanda se va încerca să se trimită comanda B pentru a cere către Kephera III versiunea și revizia sistemului de operare.



```
flamercy on koala46: /home/flamercy
File Edit View Terminal Tabs Help
Khepera3 OS (c) K-Team S.A. 2006 ver 1.3
Battery Pack Found : SN 01A000081F0
Front extension detected
Master OS mode (standalone)

I2C Has been initialised as a master
Right motor initialisation completed ver 5.2
Left motor initialisation completed ver 5.2
Serial Communication Protocol
B
b,1,3
█
```

Figura 1.13: Mesaj de boot de la Kephera III

1.3.2 Testarea legaturii seriale secundare

Cum robotul este potrivit pentru a fi folosit cu KoreBotLE este important să se testeze legătura serială între computer și KoreBotLE însuși.

- Să se asigure că toate conexiunile sunt corecte.
- Robotul trebuie plasat pe o suprafață plană și sigură. Switch-ul bateriei trebuie să fie "oprit".
- Un KoreBotLE trebuie să fie montat pe socket-ul corespunzător.
- **Un terminal emulator trebuie să pornească pe computer.** Trebuie să se asigure că terminalul este conectat la portul serial corect. **Configurația terminalului trebuie să fie setată la 115200 Baud, 8 bit, 1 start bit, 1 stop bit, no parity și no hardware control.**

Când este pornit, KoreBotLE va trimite mesajul normal linux de boot-are către terminalul emulator.

Dacă oricare dintre legăturile seriale nu funcționează corect este atunci important să se asigure că setările comunicării seriale au fost configurate corect.

⁵ *Kephera III User Manual*, version 2.2, K-Team, Switzerland, March 2008

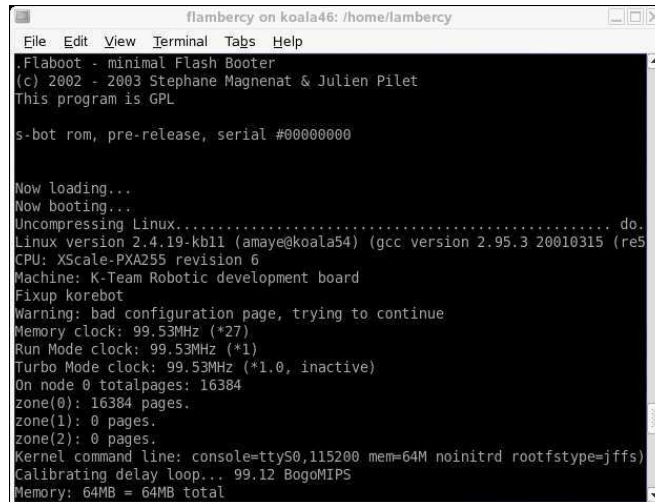


Figura 1.14: Mesaj de boot de la KoreBotLE

1.3.3 Protocolul de comunicatie seriala

Cand robotul este folosit ca singur (fara KoreBotLE), protocolul comunicatiei seriale este construit pentru a controla toate functiile Kephera III folosind linia seriala RS232. Configuratia liniei seriale (rata de baud ca si data, start, stop si bitii de paritate) pentru computer trebuie sa se potriveasca cu configuratia robotului.

Computerul si Kephera III comunica prin mesaje ASCII. Fiecare interactiune este compusa din:

- O comanda, incepand cu o litera ASCII mare si urmata, daca este necesar, de parametrii numerici sau literali separati de virgula si terminat printr-un carriage return sau o linie de feed, trimise de computer catre robot.
- Un raspuns, incepand cu aceeasi litera ASCII a comenzii dar mica si urmata, daca este necesar, de parametrii numerici sau literali separati prin virgula si terminandu-se cu un carriage return sau o linie de feed, trimise de catre Kephera III computerului.

In timpul intregii comunicatii, computerul se comporta ca master iar Kephera III ca slave. Toate comunicatiile sunt initiate de catre master.

Doua tipuri diferite de interactiuni sunt posibile. Primul set de interactiuni este folosit pentru a seta configuratia robotului prin computer (setarea liniei seriale, modificarea configuratiei controler-ilor, ...), cel de-al doilea set de interactiuni este folosit pentru a controla robotul (controlul motoarelor, valoare citita de senzori, ...).

1.3.3.1 Testarea unei simple interactiuni

Testarea unor comenzi de baza este cea mai buna metoda pentru intelegerea protocolului de comunicatie seriala si a functiilor Kephera III. Folosind o legatura seriala configurata corect intre robot si computer trebuie urmati pasii de mai jos:

- Se scrie litera **B** mare urmata de carriage return sau o linie de feed.
- Robotul trebuie sa raspunda cu **b** urmat de o indicatie a versiunii software care ruleaza pe robot and linia de feed finala.
- Se scrie litera **N** mare urmata de carriage return sau o linie de feed.
- Robotul trebuie sa raspunda cu **n** urmat de 11 numere separate de o virgula si linia de feed finala. Aceste numere sunt valorile senzorilor de proximitate ai robotului prezentati in cealalta sectiune.
- Se reincearca aceeasi comanda (**N**) punand obstacole in fata robotului. Raspunsul trebuie sa se modifice.
- Se scrie comanda de protocol **D,I10000,I-10000** urmata de carriage return sau o linie de feed.

- Robotul trebuie sa porneasca si sa raspunda cu **d** urmat de o linie de feed.
- Pentru a opri robotul se scrie comanda de protocol **D,I0,I0** urmata de carriage return sau o linie de feed.
- Se incearca si alte comenzi urmand descriere lor.

1.4 Programarea KoreBotLE

Aceasta sectiune explica modul in care se scrie un program simplu pentru KoreBotLE cu scopul de a controla robotul de baza Kephera III. Kephera III se va comporta precum un periferic la KoreBotLE.

Primul si cel mai simplu program executabil KoreBotLE/Kephera III va fi construit asumandu-se:

- Computerul are instalat un toolchain arm-linux
- Legatura NFS este activa iar directorul share-uit (*mySharedDirectory*) este pe */mnt/nfs*.

Un fisier sursa trebuie mai intai creat si editat ca un program simplu C precum urmatorul. Controler-ele au fost montate in robot dar actioneaza ca si I2C slave⁶.

```
/* kh3test.c /
#include <korebot/korebot.h>

static knet_dev_t * mot1;
static knet_dev_t * mot2;

int main()
{
    kh3_init();

    /* open various socket and store the handle in their respective pointers */

    dsPIC = knet_open( "Khepera3:dsPic" , KNET_BUS_I2C , 0 , NULL );
    mot1 = knet_open( "Khepera3:mot1" , KNET_BUS_I2C , 0 , NULL );
    mot2 = knet_open( "Khepera3:mot2" , KNET_BUS_I2C , 0 , NULL );

    /* initialize the motor controller 1 */
    kmot_SetMode( mot1 , kMotModelIdle );
    kmot_SetSampleTime( mot1 , 1550 );
    kmot_SetMargin( mot1 , 6 );
    kmot_SetOptions( mot1 , 0x0 , kMotSW0ptWindup | kMotSW0ptStopMotorBlk |
kMotSW0ptDirectionInv );
    kmot_ResetError( mot1 );
    kmot_SetBlockedTime( mot1 , 10);
    kmot_ConfigurePID( mot1 , kMotRegSpeed , 400 , 0 , 10 );
    kmot_ConfigurePID( mot1 , kMotPos , 620 , 3 , 10 );
    kmot_SetSpeedProfile(mot1, 30,3);

    /* initialize the motor controller 2 */
    kmot_SetMode( mot2 , kMotModelIdle );
    kmot_SetSampleTime( mot2 , 1550 );
    kmot_SetMargin( mot1 , 6 );
    kmot_SetOptions( mot1 , 0x0 , kMotSW0ptWindup | kMotSW0ptStopMotorBlk |
kMotSW0ptDirectionInv );
    kmot_ResetError( mot2 );
    kmot_SetBlockedTime( mot2 , 10);
    kmot_ConfigurePID( mot2 , kMotRegSpeed , 400 , 0 , 10 );
```

⁶ *Kephera III User Manual*, version 2.2, K-Team, Switzerland, March 2008

```
kmot_ConfigurePID( mot2 , kMotPos , 620 , 3 , 10 );
kmot_SetSpeedProfile(mot2, 30,3);

/* For ever loop /
while(1)
{
/* wait 5 seconds */
sleep(5);

/* Tell to the motor controller to move the Kephera III forward */
kmot_SetPoint( mot1 , kMotRegSpeed , -motspeed );
kmot_SetPoint( mot2 , kMotRegSpeed , motspeed );

/* Wait 5 seconds */
sleep(5);

/* Tell to the motor controller to stop the Kephera III forward */
kmot_SetPoint( mot1 , kMotRegSpeed , 0 );
kmot_SetPoint( mot2 , kMotRegSpeed , 0 );
}
}
```

Acest cod sursa poate fi compilat folosind *arm-linux-gcc* care suporta cele mai multe dintre optiunile gcc:

```
Arm-linux-gcc kh3test.c -o k3test
```

Un fisier executabil kh3test poate fi creat si executat pe KoreBotLE sau, eventual, o alta masina arm-linux.

Pentru a executa programul se copiaza kh3test in directorul share-uit nfs, apoi se executa dintr-un terminal KoreBotLE. Daca directorul share-uit se afla la /mnt/nfs, trebuie mai intai schimbat directorul de lucru:

```
Cd /mnt/nfs
```

Apoi se verifica faptul ca kh3test este prezent:

```
Ls -l
```

Si se executa acesta:

```
./kh3test
```

1.5 Protocolul de Comunicatie

Acest protocol de comunicatie permite controlul complet al functiilor robotului prin intermediul liniei serial RS232. Configuratia dorita este prezenta intr-un din sectiunile precedente. Linia seriala stabilita pe computer trebuie sa se potriveasca cu cea a robotului in concordanta cu modul de functionare ales⁷.

Protocolul este realizat din comenzi si raspunsuri, toate in standardul codului ASCII. O comanda este trimisa de la computer catre robot: incepe cu un caracter mare si urmat, daca este necesar, de parametrii literali sau numerici separati prin virgula si terminal de o linie de feed. Raspunsul este trimis de catre robot catre computer: incepe cu acelasi caracter mic care a fost trimis initial ca si comanda si urmat, daca este nevoie, de parametrii numerici sau literali separati prin virgula si terminal de o linie de feed⁸.

Cand o comanda solicita o valoare mai mare de 8 biti (sau mai mare de 256) urmatoarele prefixe sunt adaugate:

- d : pentru o valoare pe 16 biti
- l: pentru o valoare pe 32 biti

⁷ Descriere completa in: Sidenmark, M., „*Blue Id Access System using Bluetooth*”, University of Karlskrona, 2006.

⁸ *Kephera III User Manual*, version 2.2, K-Team, Switzerland, March 2008

George POPESCU – Comandă vocală Kephera III

- f : pentru o valoare reala

Exemplu pentru a seta viteza: D,I20000,I20000.

Pentru a intelege mai bine acest protocol se poate face referire la urmatorul exemplu de test:

- Seteaza configuratia de conexiune
- Porneste un terminal emulator pe computer cu linia seriala setata la 115200 Baud, 8 bit data, 1 start bit, 1 stop bit, no parity.
- Scrie litera majuscula **B** urmata de carriage return sau o linie de feed.
- Robotul trebuie sa raspunda cu **b** urmat de un indicator al versiunii soft-ului care ruleaza pe robot si terminat de o linie de feed.
- Scrie litera majuscula **N** urmata de carriage return sau o linie de feed.
- Robotul trebuie sa raspunda cu **n** urmat de 11 numere separate prin virgula si terminat prin o linie de feed. Aceste numere sunt valorile senzorilor de proximitate prezenti pe robot.
- Reințearca aceeasi comanda **N** punand obstacole in fata robotului. Raspunsul trebuie sa se modifice.
- Ințearca alte comenzi...

A Porneste modul Braintenberg

Comanda: A, mod (b biti)

Raspuns: a

Efect: Ințepe modul Braintenberg cu senzorul infrarosu (mod = 0) sau cu senzorul ultrasonic (mod = 1).
Pentru a opri modul Braintenberg trimite comanda A,2.

Sintaxa: A,0

B Citeste versiunea software

Comanda: B

Raspuns: b, versiune_BIOS, revizie_BIOS

Efect: Obtine versiunea software stocata in memoria EEPROM a robotului.

Index	Valoare default	Descriere
0	0b00000100	Decide cati senzori ultrasonici sunt activi.
1	3	Decide numarul maxim de ecouri
2	0	Nu este setat.
3	0	Nu este setat.
4	0xFFFF	Masca senzorului infrarosu, gestioneaza senzorul infrarosu (bit0=IR=, bit1=IR1, etc.).
5	12	Castig infrarosu pentru Braintenberg
6	1	Corpul superior este montat. Daca corpul superior nu este montat schimba acesta valoare la 0 pentru a obtine o masura a ultrasunetelor.

Configuratia senzorilor ultrasonici suportata este listata in urmatorul tabel:

George POPESCU – Comandă vocală Kephera III

Masca	Binar	Zecimal	Descriere
Niciuna	0b00000000	0	Niciunul dintre senzorii ultrasonici
US1ONLY	0b00000001	1	Doar senzorul US 1
US2ONLY	0b00000010	2	Doar senzorul US 2
US3ONLY	0b00000100	4	Doar senzorul US 3
US4ONLY	0b00001000	8	Doar senzorul US 4
US5ONLY	0b00010000	16	Doar senzorul US 5
US1TO2	0b00000011	3	Senzorii 1 si 2
US2TO3	0b00000110	6	Senzorii 2 si 3
US3TO4	0b00001100	12	Senzorii 3 si 4
US4TO5	0b00011000	24	Senzorii 4 si 5
US1TO3	0b00000111	7	Senzorii 1, 2 si 3
US2TO4	0b00001110	14	Senzorii 2, 3 si 4
US3TO5	0b00011100	28	Senzorii 3, 4 si 5
US1TO4	0b00001111	15	Senzorii 1, 2, 3 si 4
US2TO5	0b00011110	30	Senzorii 2, 3, 4 si 5
USALL	0b00011111	31	Toti senzorii

D Seteaza viteza

Comanda: D, viteza_motor_stang (32 biti), viteza_motor_drept (32 biti)

Raspuns: d

Efect: Seteaza viteza celor 2 motoare.

Sintaxa: D,l20000,l200000

E Citeste viteza

Comanda: E

Raspuns: e, viteza_motor_stanga, viteza_motor_dreapta

Efect: Citeste viteza instantanee a celor 2 motoare.

Exemplu: e,20000,200000

F Seteaza profilul tinta

Comanda: F, pozitia_tinta_motor_stanga (32 biti), pozitia_tinta_motor_dreapta (32 biti)

Raspuns: f

Efect: Seteaza o pozitie sa fie atinsa. Miscarea va fi realizata in 3 faze, o acceleratie pentru a atinge viteza maxima, o viteza constanta si o faza de decelerare intainte de pozitia terminala. Unitate este pulsatia, fiecare corespunzand la 0,047 mm (cu rezolutia encoder-ului x4 (initial)).

Sintaxa: F,l1000,l10000

N Citeste senzorii de proximitate

Comanda: N

Raspuns: n, val_senz_spate_stanga, val_senz_stanga_90, val_senz_stanga_45, val_senz_fata_stanga, val_senz_fata_dreapta, val_senz_dreapta_45, val_senz_dreapta_90, val_senz_spate_dreapta, val_senz_spate, val_senz_jos_dreapta, val_senz_jos_stanga, timp

Efect: Citeste valorile pe 10 biti ai celor 11 senzori de proximitate. Ultimul argument este timpul relativ de masura, de la primul senzor situat la stanga robotului si pana la cel din spate-stanga in sens anti-trigonometric.

O Citeste valorile luminii ambientale

Comanda: O

Raspuns: n, val_senz_spate_stanga, val_senz_stanga_90, val_senz_stanga_45, val_senz_fata_stanga, val_senz_fata_dreapta, val_senz_dreapta_45, val_senz_dreapta_90, val_senz_spate_dreapta, val_senz_spate, val_senz_jos_dreapta, val_senz_jos_stanga, timp

Efect: Citeste valorile pe 10 biti ai celor 11 senzori de proximitate. Ultimul argument este timpul relativ de masura, de la primul senzor situat la stanga robotului si pana la cel din spate-stanga in sens anti-trigonometric.

P Seteaza pozitia tinta

Comanda: P, pozitia_tinta_motor_stanga (32 biti), pozitia_tinta_motor_dreapta (32 biti)

Raspuns: p

Efect: Seteaza pozitia destinatie. Miscarea va fi inregistrata cu acceleratie si deceleratie. Unitatea este pulsatie, fiecare corespunde la 0,047 mm (cu rezolutia encoder-ului x4 (initial)).

Sintaxa: P,I1000,I1000

Q nefolosit

R Citeste pozitia

Comanda: R

Raspuns: r, pozitie_motor_stanga, pozitie_motor_dreapta

Efect: Citeste counter-ul pe 32 de biti al celor 2 motoare. Unitatea este pulsatie, fiecare corespunde la 0,047 mm (cu rezolutia encoder-ului x4 (initial)).

S nefolosit

T nefolosit

U Seteaza pozitia tinta

Comanda: U

Raspuns: u

Efect: Schimba pentru modul de firmware upgrade. Se va avea grija sa nu se incarca un fisier hexazecimal gresit in spatiul de memorie al dsPic-ului. Daca memoria dsPic-ului este corupta, un program extern va fi necesar pentru a reface memoria. Nu se va inchide sau reseta robotul cand programul ruleaza. Acesta va corupe memoria dsPic-ului. Daca s-a pornit acest mod din greseala, se va astepta timp de 5-7 secunde.

V Obtine starea bateriei

Comanda: V, index (8 biti)

Raspuns: v, unitatea_de_masura_a_bateriei, masura_zecimala_a_bateriei

Efect: 0: Citeste voltajul bateriei (Unitati: 1V, 0.1mV).

1: Citeste curentul prin baterie (Unitati: 1A, 0.1mA).

2: Citeste curentul mediu al bateriei (Unitati: 1A, 0.1mA).

George POPESCU – Comandă vocală Kephera III

3: Citeste capacitatea absoluta ramas a bateriei (Unitati: 1Ah).

4: Citeste temperatura bateriei (Unitati: 1 C, 0.1).

5: Citeste capacitatea relativa ramasa a bateriei (Unitati:%).

Sintaxa: P,I1000,I1000

T nefolosit

X nefolosit

Y nefolosit

Z Timpul relativ zero

Comanda: Z

Raspuns: z

Efect: Reseteaza counter-ul pentru timpul relativ.

2. COMANDA ASISTATA A ROBOTULUI KIII POSIBILITATI DE COMUNICATIE

2.1 Interfata de instalare Kephera III

Vom incerca sa realizam conexiunea dintre Kephera III si laptop-ul DELL D630 prin intermediul interfeței grafice de care dispune robotul. Pentru acesta este nevoie de setarea porturilor de comunicare între cele 2 dispozitive.

Pe langa componentele fizice de care dispune KHIII, in CD-ul de instalare, pe langa manualele si cartea tehnica exista si programul pentru realizarea interfatarii la un computer normal.

In colectia de fisiere ale robotului exista si setup.exe, fisier pentru instalarea interfeței Kephera III pe PC-ul gazda pentru realizarea comunicatiei cu robotul.

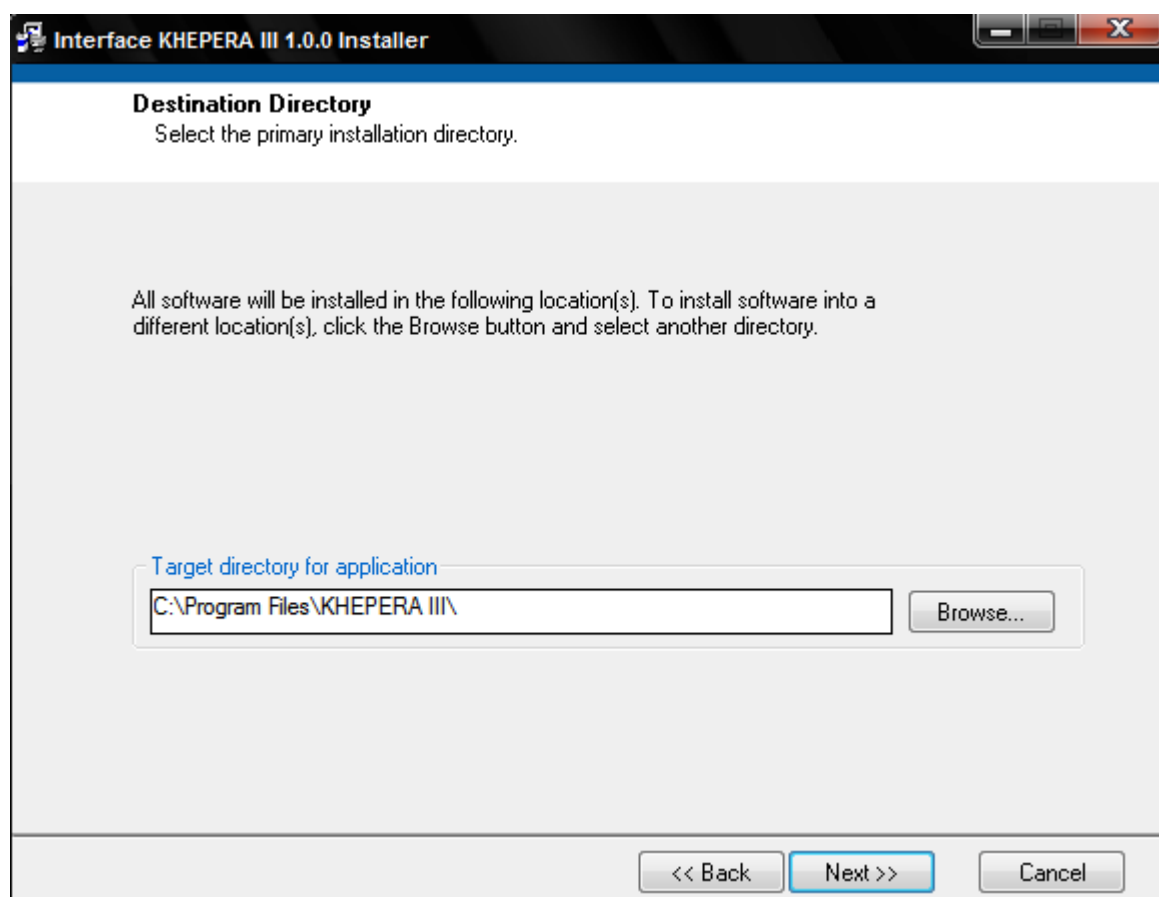


Figura 2.1: Rularea soft-ului de instalare al interfeței – alegerea locatiei pentru instalare

In cele ce urmeaza vom instala acest program si vom deschide conexiunea pentru a realiza controlul la distanta al robotului.

In fereastra imediat urmatoare va fi afisat un sumar al optiunilor selectate pentru instalare.

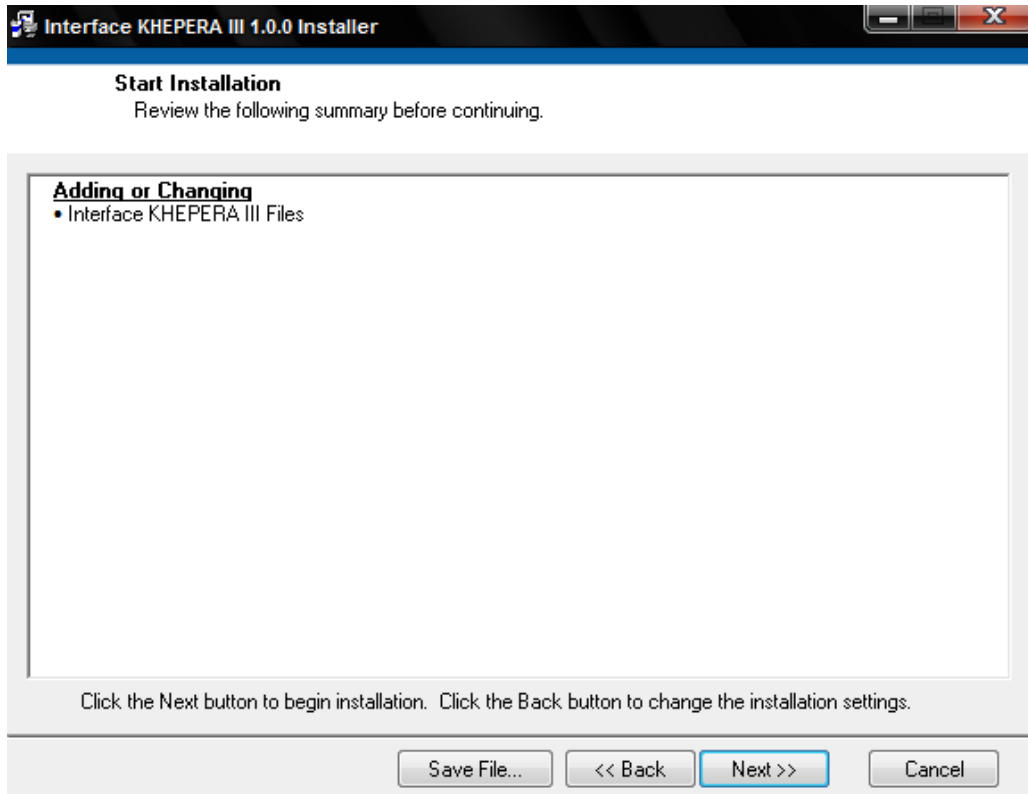


Figura 2.2: Sumar al instalarii

Programul va instala fisierele corespunzatoare.

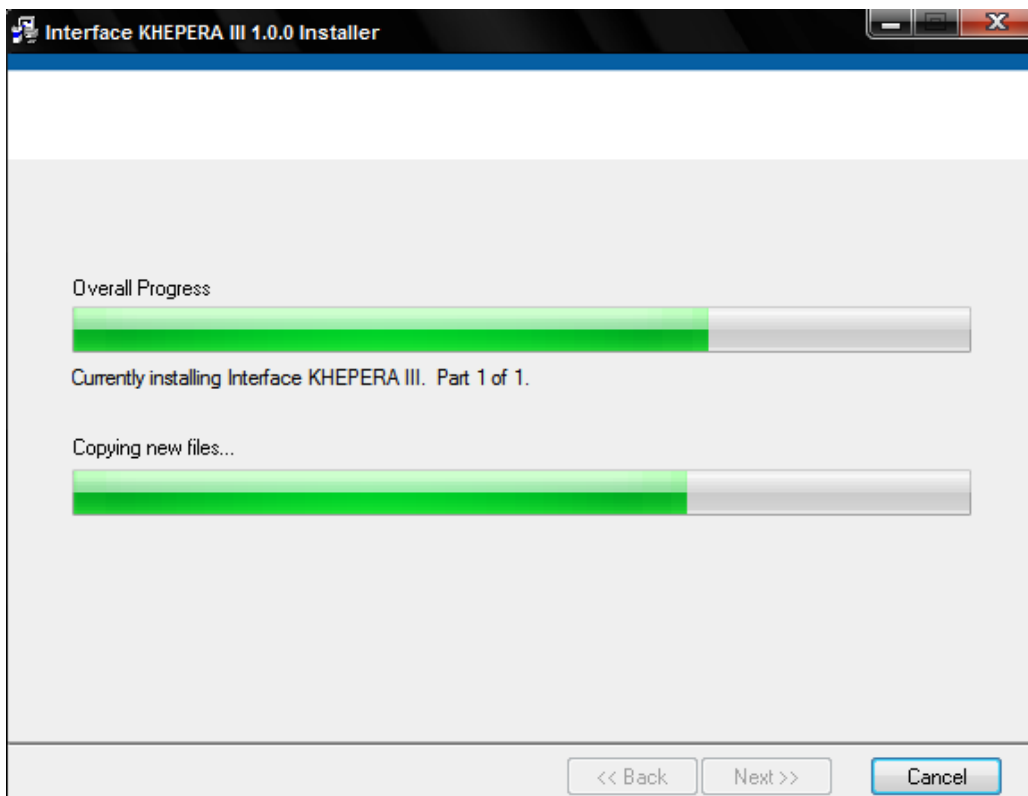


Figura 2.3: Instalare interfata Kephera III

Instalarea s-a realizat cu succes.

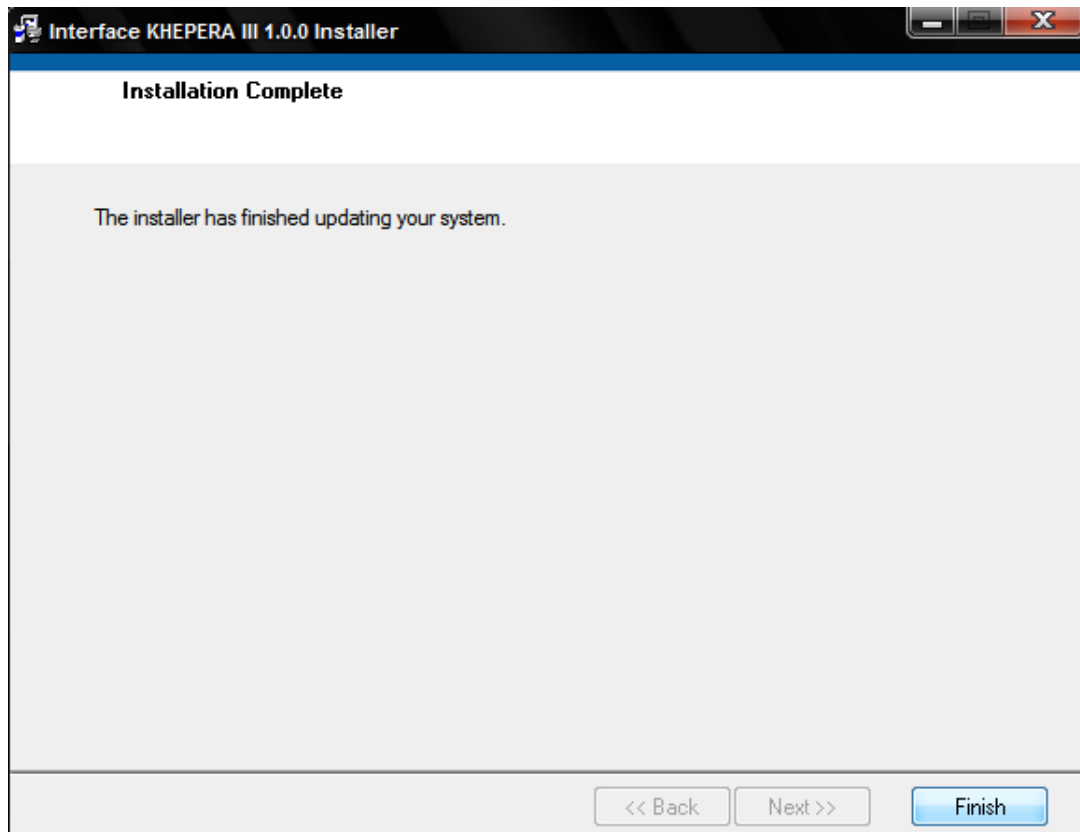


Figura 2.4: Instalare realizata cu succes

Programul a fost instalat iar la deschiderea aplicatiei, utilizatorul va fi intampinat de urmatoarea interfata.



Figura 2.5: Interfata Kephera III

George POPESCU – Comandă vocală Khepera III

Se poate observa pe fundal diferite imagini din diverse unghiuri cu robotul precum și un meniul în partea dreapta-sus. Conexiunea propriu-zisă va fi inițiată prin switch-ul din bara de sus, în partea stanga.

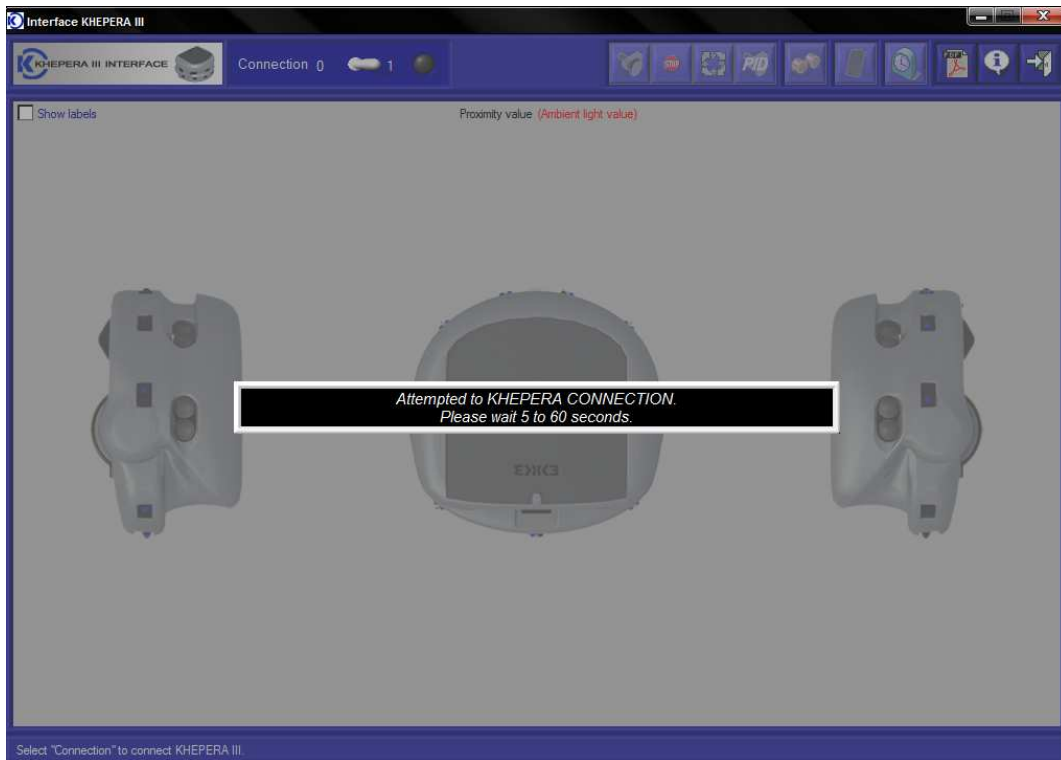


Figura 2.6: Incercare de conexiune

Conexiunea a eșuat pe portul inițial de conectare. Motivul prezentat este legat de faptul că robotul nu a răspuns conexiunii.

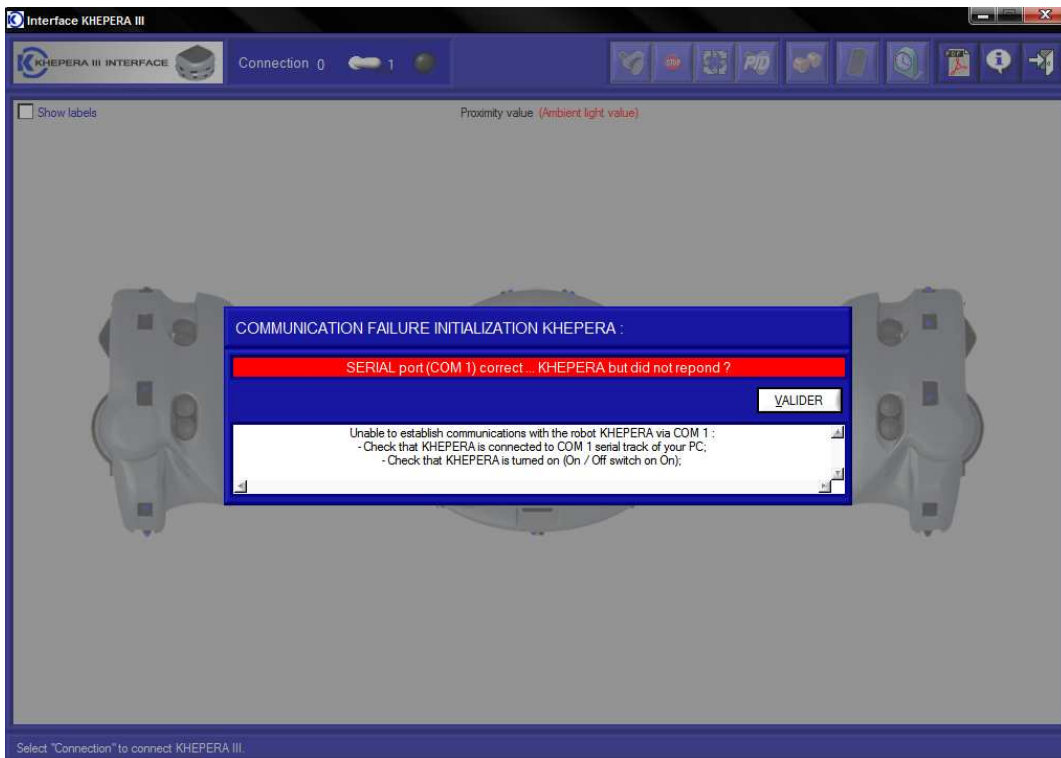


Figura 2.7: Conexiune eșuată COM1

Alta eroare de conexiune este legata de adresarea incorecta a portului.



Figura 2.8: Adresarea incorecta a portului

Pentru a verifica pe ce port robotul realizeaza conexiunea cu PC-ul se va deschide urmatoarea fereastra:

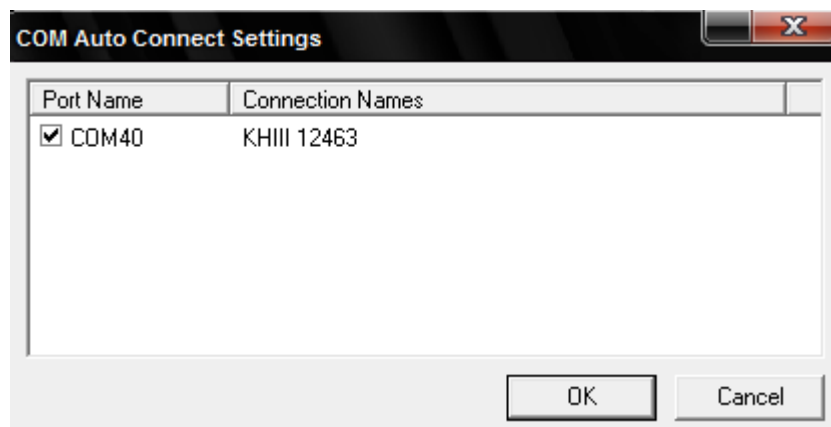


Figura 2.9: Portul de comunicatie pentru Kephera III

Din lista de porturi COM disponibile se poate alege o valoare cuprinsa intre COM1 si COM10.

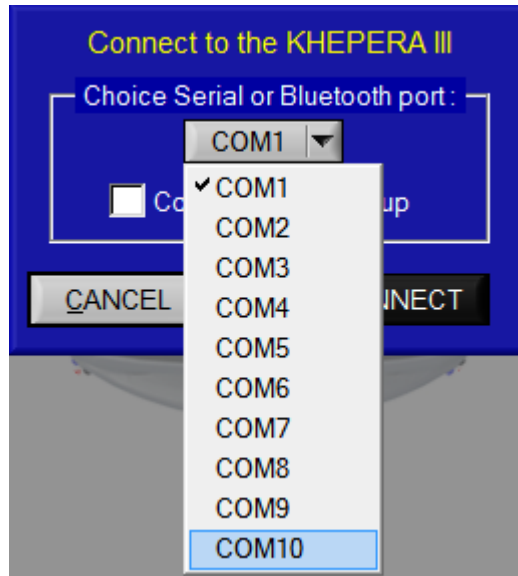


Figura 2.10: Porturi COM disponibile pentru conectare

2.2 Comunicatia Robot – Laptop prin intermediul interfetei USB

Primul pas este reprezentat de conectarea cablului USB la Kephera III precum in imaginea de mai jos:



Figura 2.11: Folosirea conexiunii ABmini-USB a robotului

Observatie: Robotul poate fi mentinut in starea de alimentare.



Figura 2.12: Conexiune USB2.0 - Abmini-USB

Folosind un alt cablu de conectare USB (precum cel pentru imprimante), prin intermediul conectorului KoreConnector este posibila conectarea robotului Kephera III la laptop (si in timpul alimentarii).



Figura 2.13: Conexiune USB (Laptop) – Serial (Robot)

Dupa cum se observa in imagine KoreConnect are aprinse cele 2 LED-uri: portocaliu si rosu ale conexiunii:

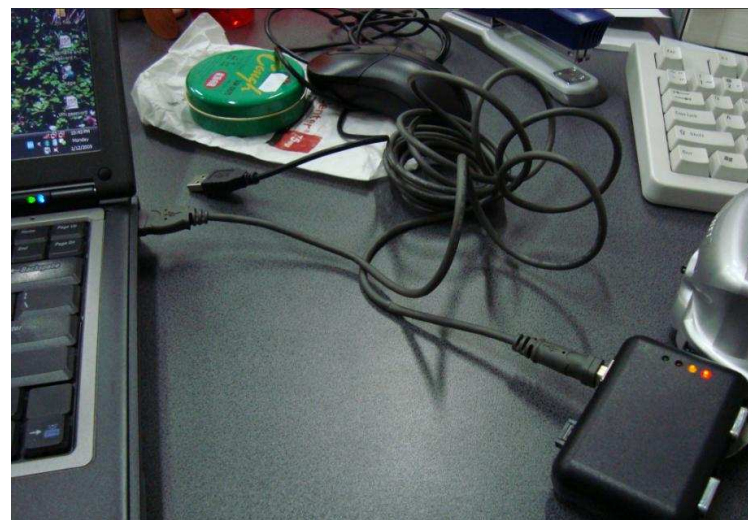


Figura 2.14: Conexiune Laptop – KoreConnect

Kephera III si KoreConnect in timpul alimentarii fara o conexiune cu laptop-ul:



Figura 2.15: a) Kephera III + KoreConnect

Kephera III si KoreConnect = alimentare; doar LED-ul este aprins.



Figura 2.16: b) Kephera III + KoreConnect

2.3 Comunicatia Bluetooth

Dupa ce a fost activata conexiunea Bluetooth pe laptop, aplicatia va incerca sa detecteze toate dispozitivele active in raza de actiune:



Figura 2.17: Cautarea dispozitivelor Bluetooth

Computerul a găsit conectorul Bluetooth al robotului, identificat prin KHIII 12463.

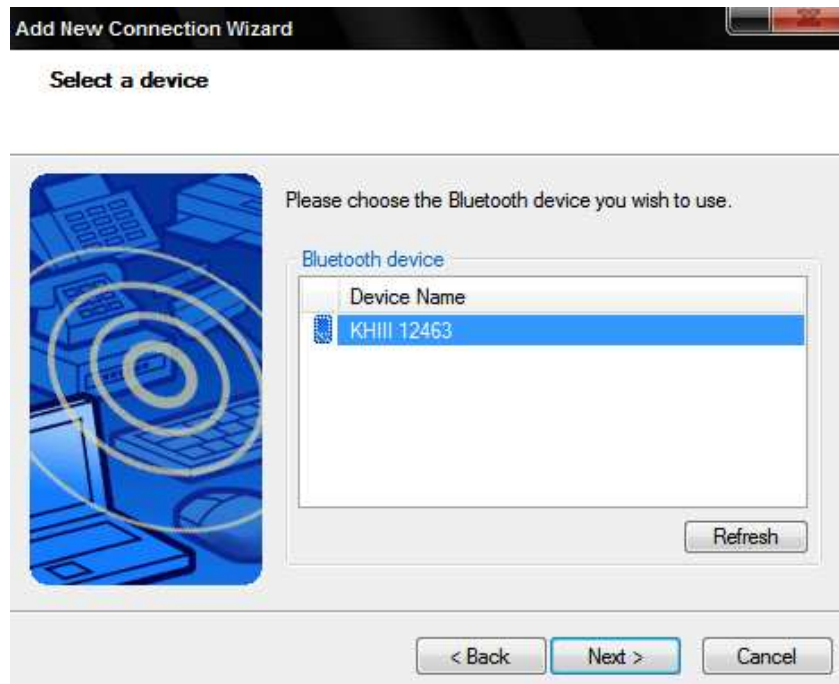


Figura 2.18: Kephera III a fost descoperit de catre Computer

In continuare se va realiza conexiunea efectiva cu acesta⁹.

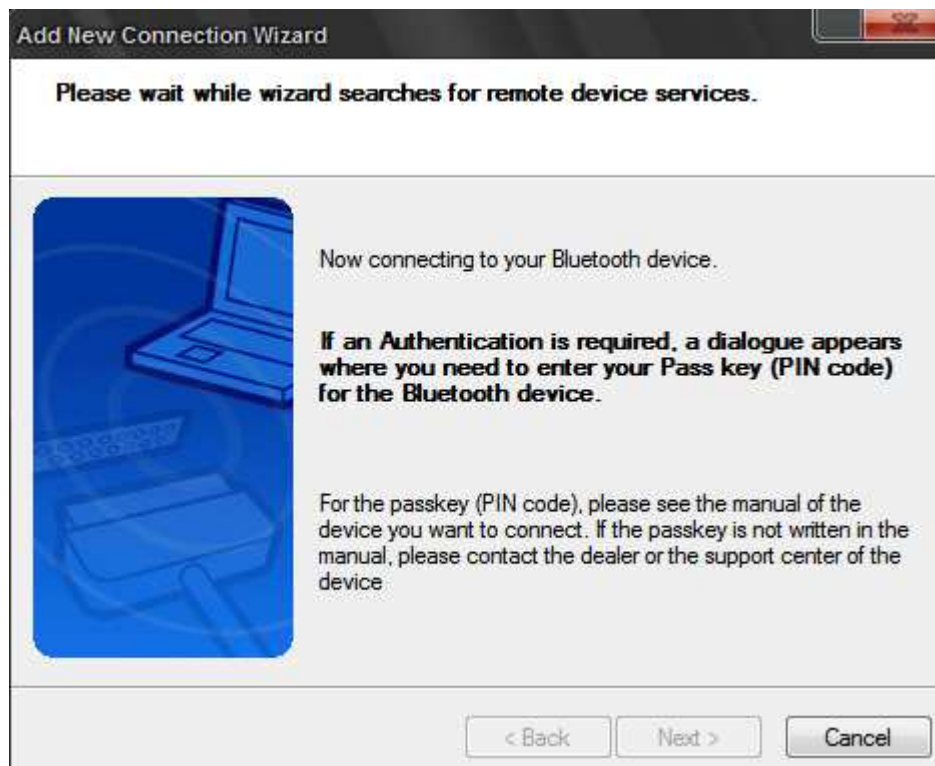


Figura 2.19: Conectarea via Bluetooth intre cele dispozitive

⁹ Abordari similare: Mayer H., Schilling K. Mayer H. and Harmo P. Halme A, „*Mobile robot control via Bluetooth technology*”, Zurcher Hochschule Winterthur, IEEE, 2008.

În etapa următoare, întrucât conexiunea este una securizată, utilizatorului îi este solicitat PIN-ul:

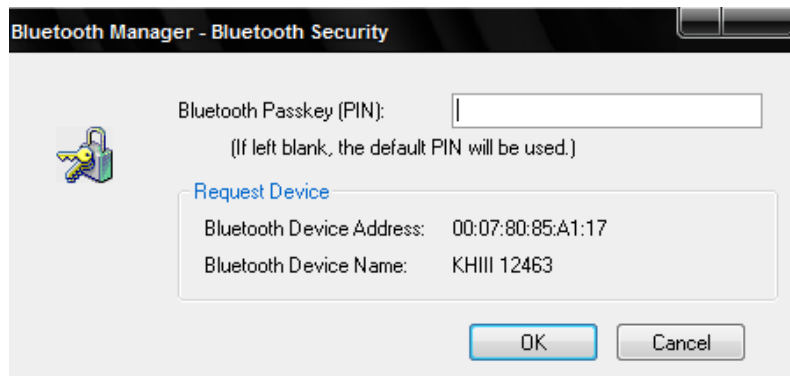


Figura 2.20: Introducere PIN de către utilizator – conexiune securizată

Urmărește setarea portului de conexiune.

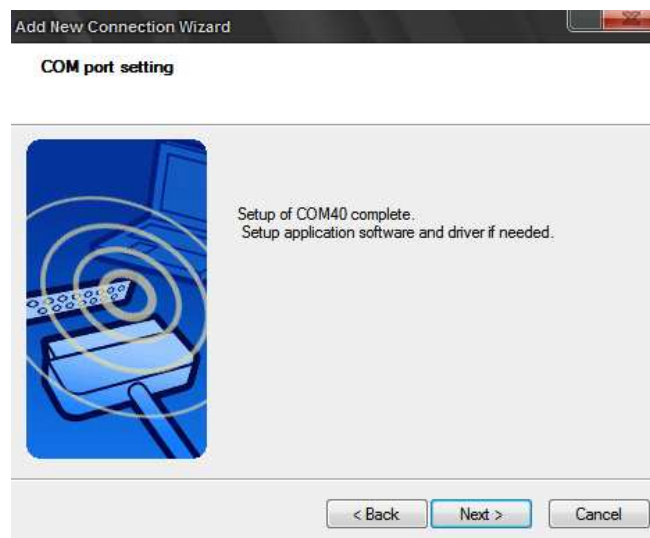


Figura 2.21: Setarea portului de comunicare COM40

În fereastra care deschide conexiunea realizată robotul va fi identificat printr-o pictogramă.

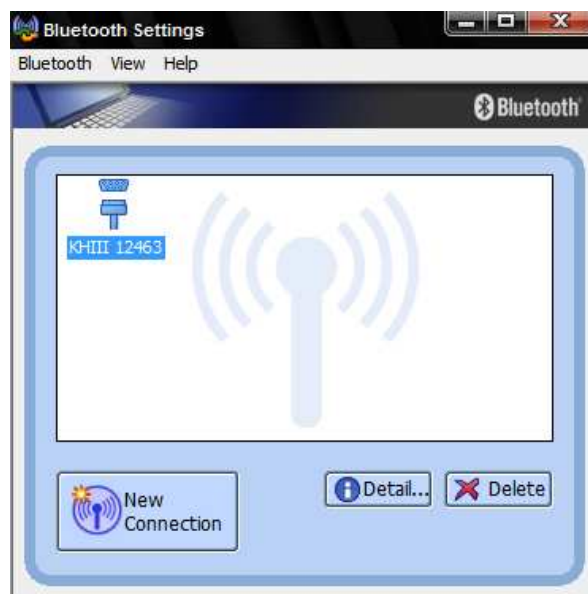


Figura 2.22: Dispozitivele Bluetooth conectate la Laptop

Ulterior pot fi vizualizate atat proprietati cat si optiuni/setari pentru conexiunea realizata.

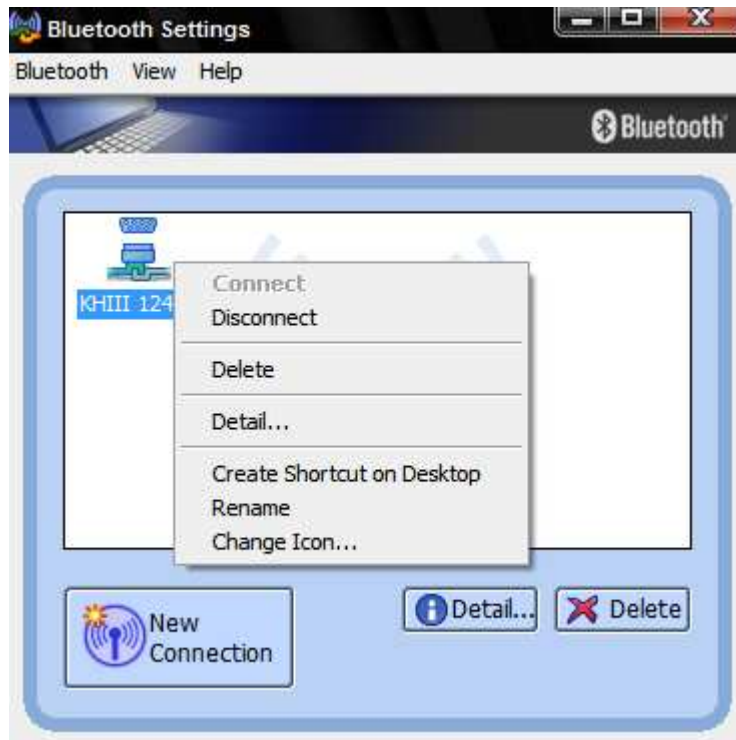


Figura 2.23: Meniul de conexiune Bluetooth

Prin alegerea butonului "Details" utilizatorul va avea acces la urmatoarele informatii:

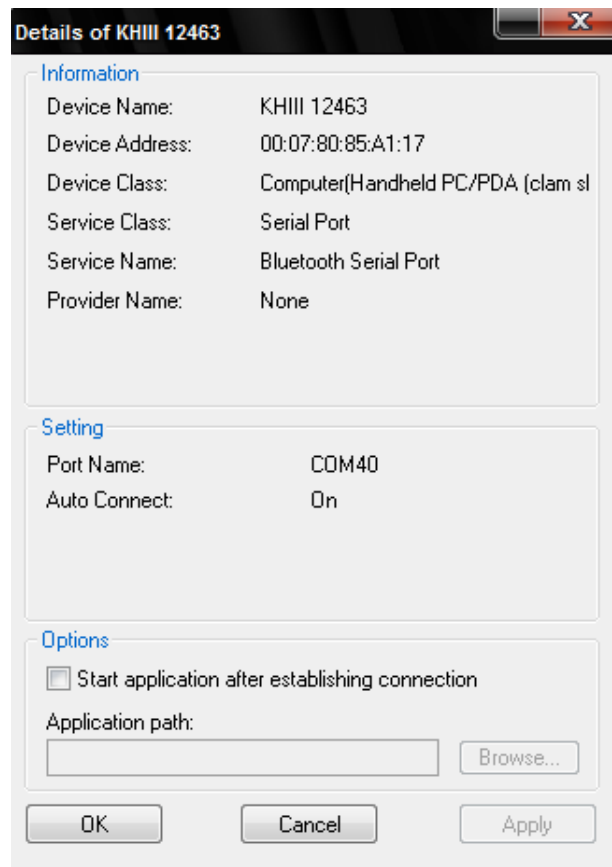


Figura 2.24: Informatii despre KHIII – 12463

Prin intermediul aceleiasi conexiuni am incercat si conectarea PDA-ului:



Figura 2.25: Cele 2 dispozitive conectate via Bluetooth: KHIII si PDA Fujitsu

Informatiile legate de cea de-a doua conexiune sunt afisate in mediul de "detalii"¹⁰:

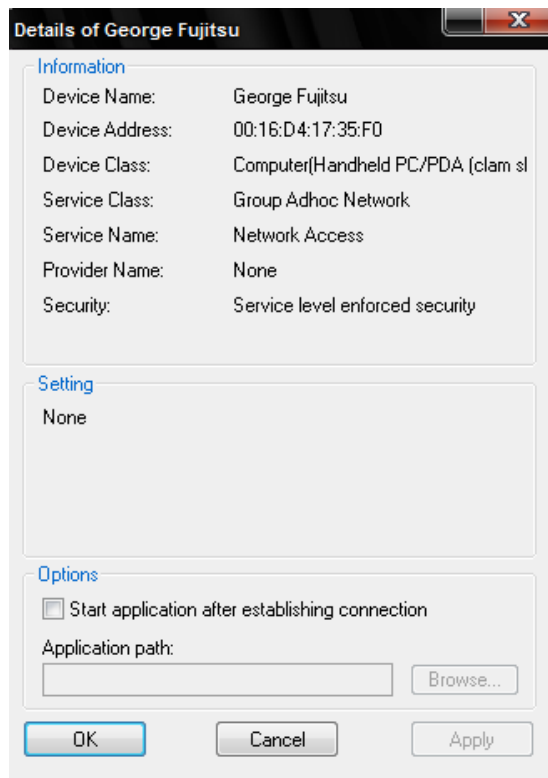


Figura 2.26: Detaliile conexiunii Laptop-PDA

¹⁰ Controlul la distanta: Mayer H., Schilling K. Mayer H. and Harmo P. Halme A, „Mobile robot control via Bluetooth technology”, Zurcher Hochschule Winterthur, IEEE, 2008.

În paleta “Opțiuni”, la sub-meniul “General” vom găsi următoarele informații:

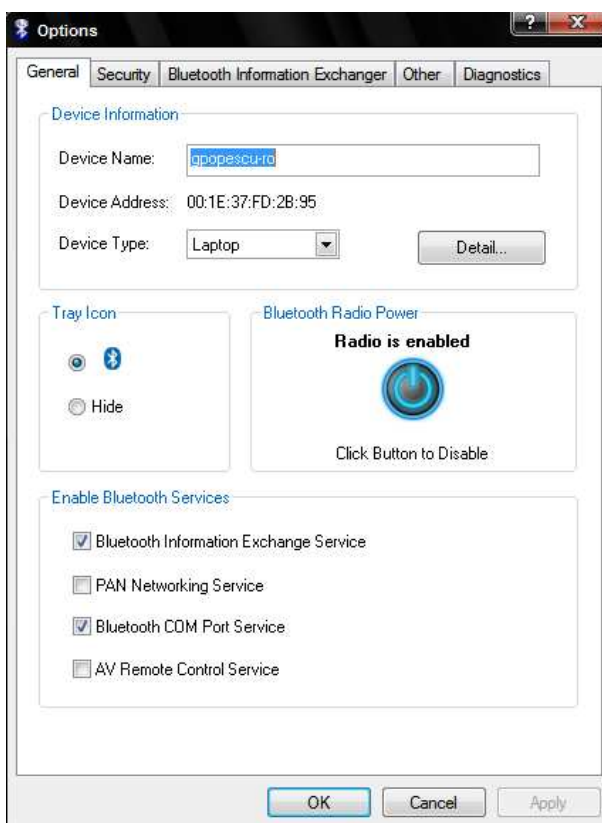


Figura 2.27: Informații din tab-ul “General”

Putem vizualiza și informațiile legate de securitatea conexiunii:

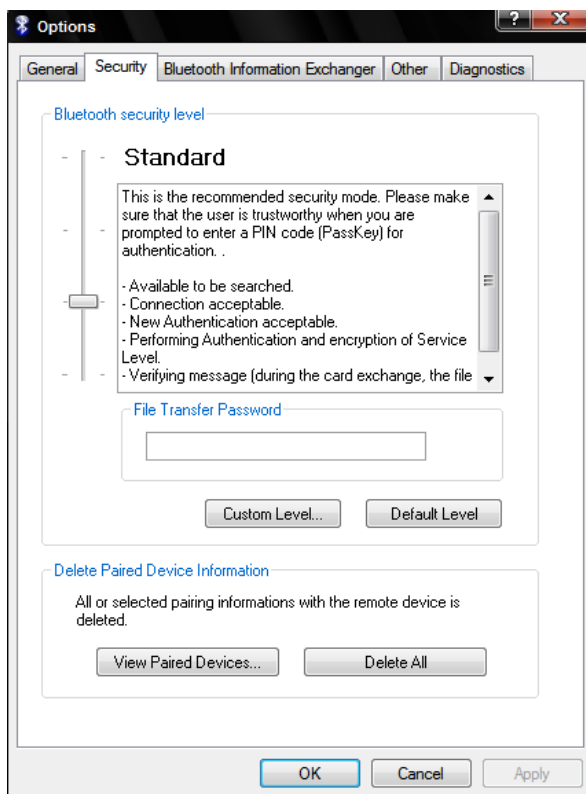


Figura 2.28: Informații din tab-ul de securitate

In meniul imediat urmator apar informatii cu privire la schimbul de informatii via Bluetooth:

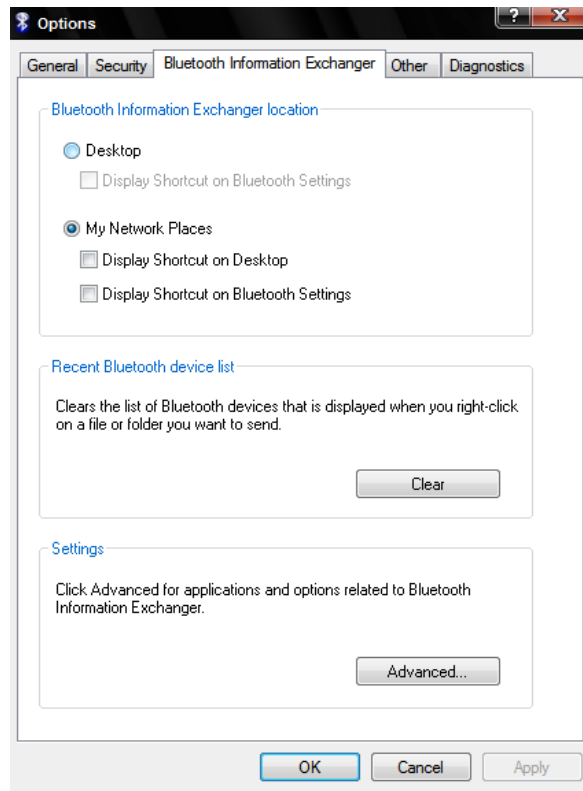


Figura 2.29: Schimb de pachete de informatii via Bluetooth

Sub-meniul "Other" ofera alte informatii cu privire la setarile conexiunii pentru portul COM, selectia si configuratia player-ului AV:

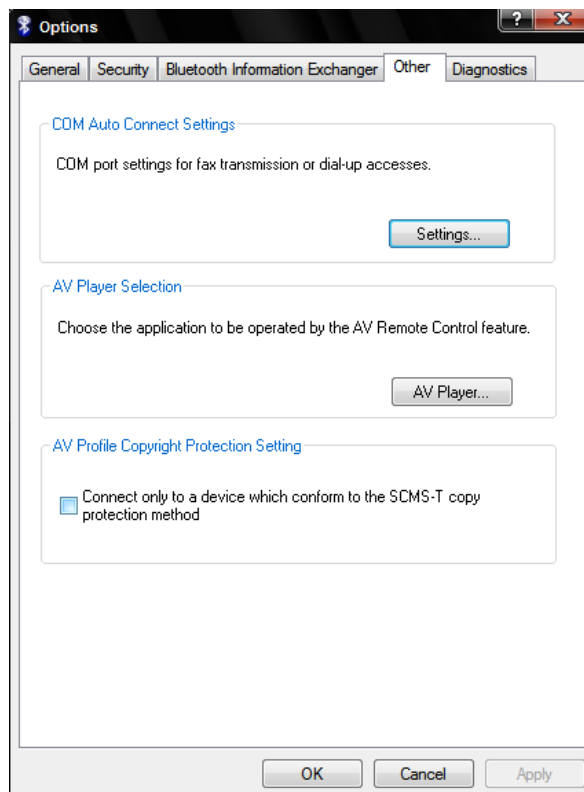


Figura 2.30: Alte informatii

Ultimul tab indica si testele care pot fi realizate:

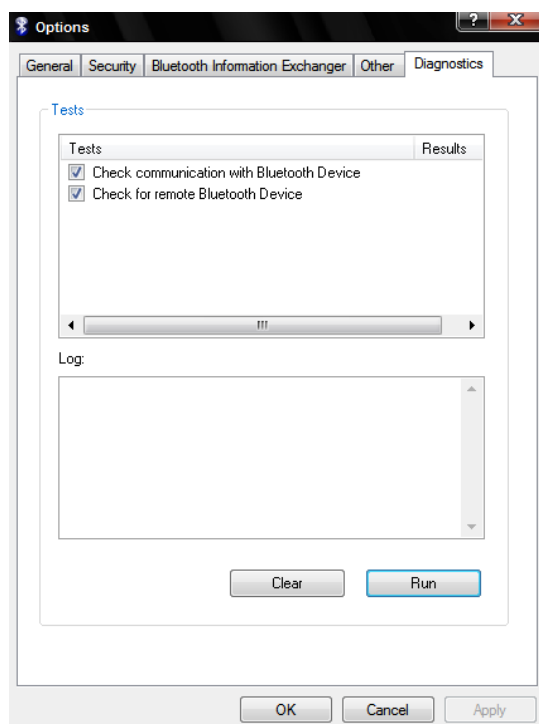


Figura 2.31: Diagnostic al conexiunii: teste

2.4 Comunicatie Bluetooth Kephera III – Fujitsu-Siemens Pocket LOOX T830

Pentru a testa comunicatia Bluetooth pentru Kephera III am dispus de un PDA¹¹ produs de compania Fujitsu-Siemens, PC Pocket LOOX T 830 care dispune de toate capacitatile unei astfel de comunicatii¹².



Figura 2.32: PDA Fujitsu Siemens Pocket LOOX T830

PDA-ul are preinstalat sistemul de operare Windows Mobile 5.1 impreuna cu suita de programe standard. Pentru stabilirea conexiunii cu robotul Kephera III va trebui sa activam functia Bluetooth pe fiecare dintre acestia (Kephera III are in mod automat acesta setare)¹³.

¹¹ PDAdriver: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.1234>

¹² Comunicatia cu dispozitive similare este descrisa in: Hoptroff, R., „Bluetooth remote robot control”, Elektor Electronics, 2008.

¹³ Alte tipuri de dispozitive PDA testate pentru comunicatia Bluetooth: Terrence W. Fong, Chuck Thorpe, Betty Glass, „PDA driver – A handheld system for remote driving”, IEEE International Conference on Advanced Robotics, July 2006.

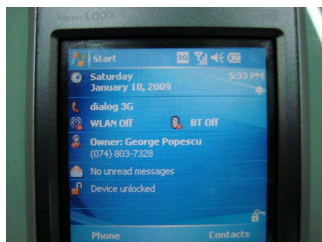


Figura 2.33: Ecranul de start al PDA-ului

Setarile pentru Bluetooth se gasesc in meniul principal al sistemului de operare in categoria "Settings".

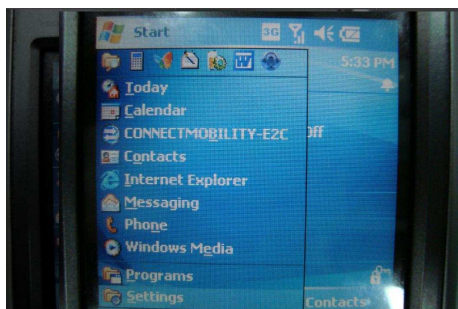


Figura 2.34: Meniul principal

Vom naviga catre functia Bluetooth din submeniul "Connections"¹⁴.



Figura 2.35: Selectarea din meniul de conexiuni a functiei Bluetooth

Profilul general este cel standard; functia Bluetooth a fost activata.

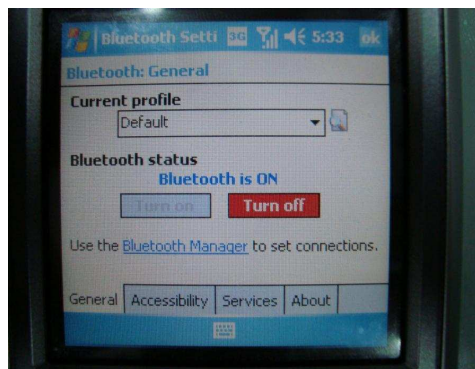


Figura 2.36: Activarea functiei Bluetooth

Folosind "Bluetooth Manager" vom descoperi toate dispozitivele aflate in raza de actiune. In interval de 30 de secunde PDA-ul a descoperit robotul Kephera III avand numarul de identificare 12070.

¹⁴ Alte dispozitive PDA folosite: Zijng Lin, Meng, M.Q.-H., Wanming Chen, Huawei Liang, Xin Liu, „*Design of a PDA-based telerobotic system*”, ROBIO, IEEE International Conference on Robotics and Biomimetics, 2007.



Figura 2.37: Gasire robot Kephera III

In continuare vom stabili conexiunea seriala Bluetooth cu robotul.



Figura 2.38: Conectare seriala Bluetooth

Conectarea a fost realizata cu succes! Din cate se poate observa in figura de mai jos puterea semnalului este potrivita pentru aceasta conexiune. Un semnal prea slab ar fi insemnat pierderea conexiunii in orice moment¹⁵. Conform cartii tehnice si manualului de utilizare al robotului configuratia Bluetooth a robotului presupune¹⁶:

- 115200bps data rate; 8 data bits;
- no parity; one stop bit;
- **no hardware flow control**
- **codul de securitate: 0000**

Alte observatii:

- Durata conexiunii este de 12 secunde iar portul folosit este cel Serial Bluetooth.
- Semnalul este stabil
- Inca nu au fost trimise pachete de informatie intre cele 2 dispozitive¹⁷

¹⁵ Comunicatia wireless prin Bluetooth: Espinosa Horacio, Macedonia David, Same Adam, Storch David, Wroblewski Norbert, „*Robotic Control with Bluetooth Wireless Communication*”, Northwestern University, 2007.

¹⁶ Alte dispozitive PDA functioneaza similar: Terrence W. Fong, Chuck Thorpe, Betty Glass, „*PDA driver – A handheld system for remote driving*”, IEEE International Conference on Advanced Robotics, July 2006.

¹⁷ PDAdriver: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.1234>

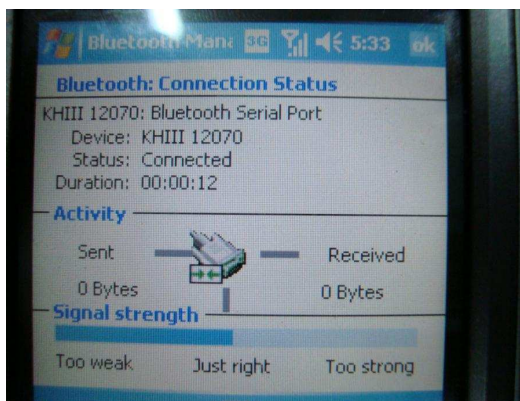


Figura 2.39: Informatii legate de conexiunea seriala Bluetooth [reusita]¹⁸

2.5 Control la distanta Kephera III

Controlul la distanta al robotului Kephera III se va realiza prin intermediul unui dispozitiv Bluetooth (USB stick). Dupa instalarea driver-ului, "My Bluetooth places" va contine pictograma pentru identificarea portului de comunicatie Bluetooth seriala.

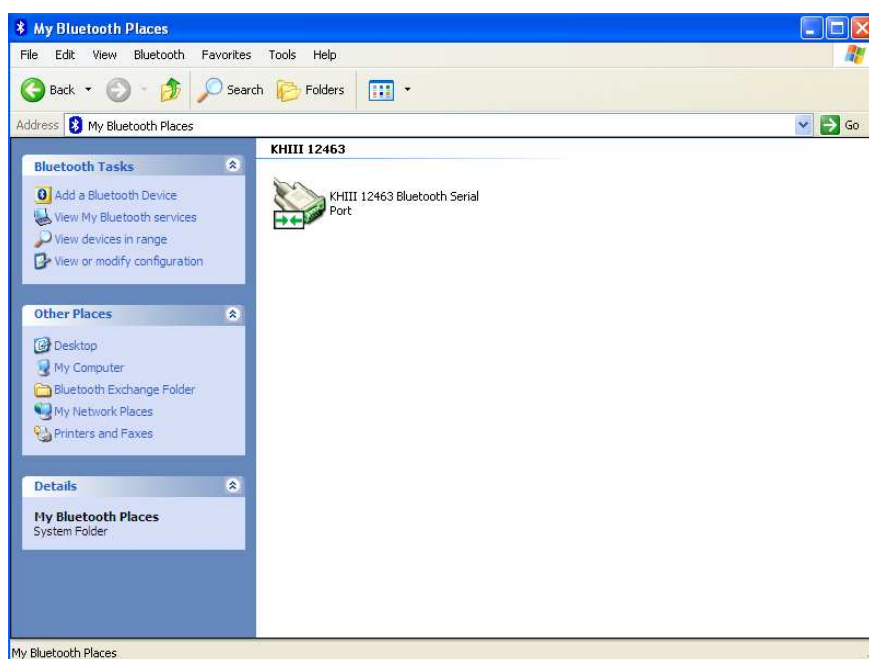


Figura 2.40: Descoperirea Bluetooth a KHIII 12463 prin interfața Bluetooth Seriala Computer-ului va afisa seria robotului la conectare. Sagetile verzi indica posibilitatea de conectare.

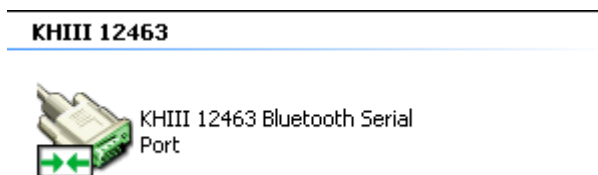


Figura 2.41: Kephera III 12463: Activare port Bluetooth serial

¹⁸ Exemple de conexiuni Bluetooth similare: Zijng Lin, Meng, M.Q.-H., Wanming Chen, Huawei Liang, Xin Liu, „Design of a PDA-based telerobotic system”, ROBIO, IEEE International Conference on Robotics and Biomimetics, 2007.

Semnalul Bluetooth, după 1 minut și 27 secunde de la pornirea conexiunii are intensitatea potrivită¹⁹.

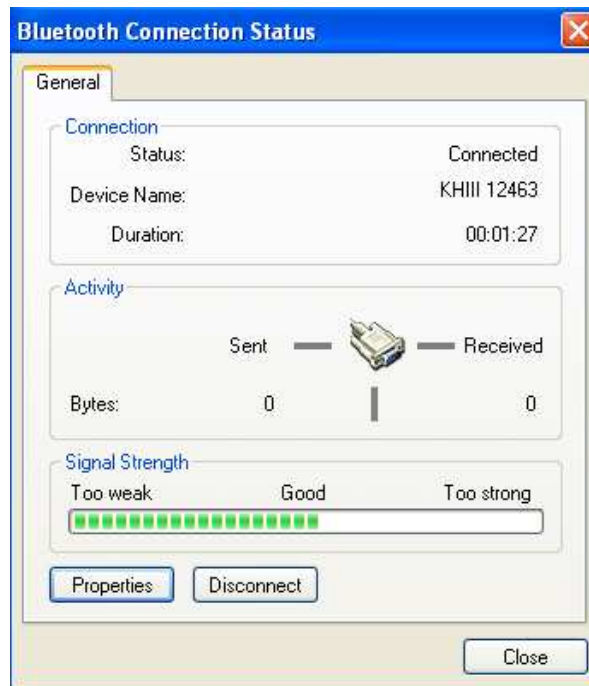


Figura 2.42: Valoarea intensității semnalului Bluetooth

În „Tab”-ul „General” al conexiunii se va observa faptul că pentru conexiunea Bluetooth, conexiunea se va realiza prin intermediul portului de comunicare cu numărul **5**.

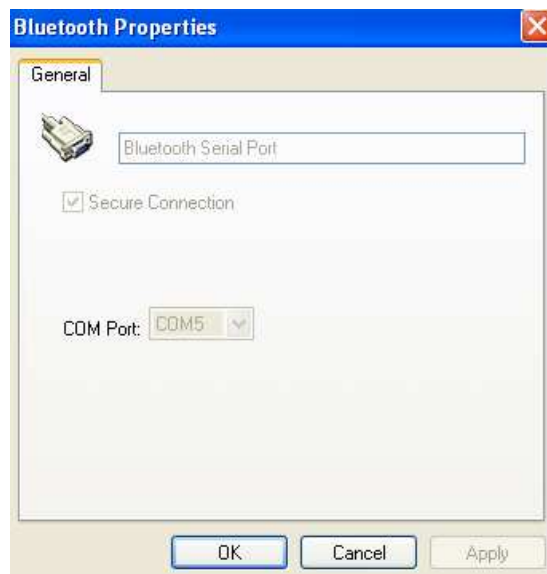


Figura 2.43: Portul de comunicare cu numărul 5 selectat implicit

În continuare vom realiza conexiunea Bluetooth (în „My Bluetooth places”, click pe pictograma conexiunii)

¹⁹ Controlul la distanță prin intermediul conexiunii Bluetooth: Mayer H., Schilling K. Mayer H. and Harjo P. Halme A, „Mobile robot control via Bluetooth technology”, Zurcher Hochschule Winterthur, IEEE, 2008.



Figura 2.44: Incercare de conexiune Bluetooth

Dupa realizarea conexiunii si detectia robotului va fi afisata o noua pictograma cu acesta, precum si alte dispozitive aflate in raza de actiune Bluetooth.



Figura 2.45: Dispozitivele Bluetooth conectate la Laptop

In continuare se va activa "icon"-ul Bluetooth in coltul din dreapta-jos al monitorului.

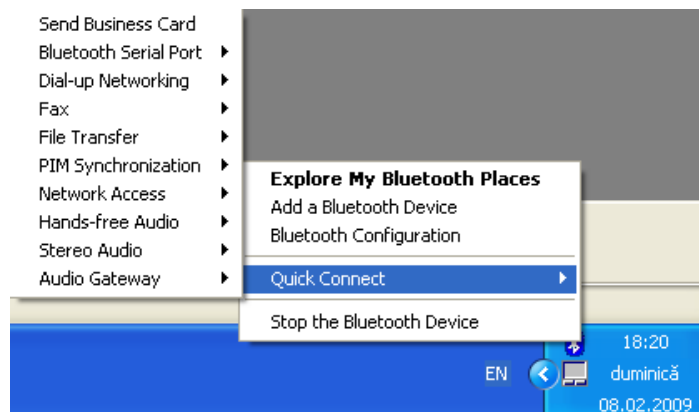


Figura 2.46: Meniul de comunicatie Bluetooth

Gasirea robotului poate fi imediata prin rpin click asupra pictogramei Bluetooth din dreapta monitorului.

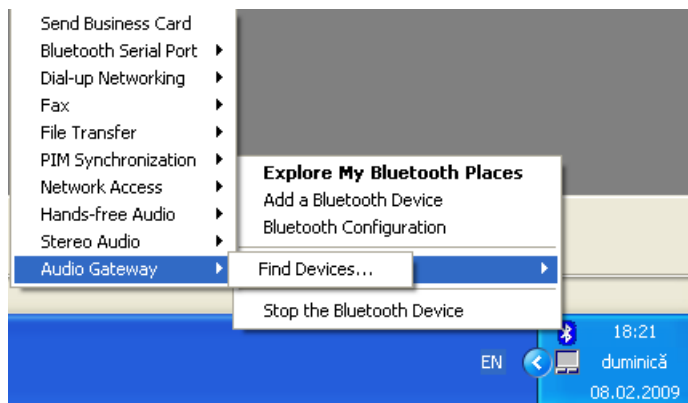


Figura 2.47: Meniul Bluetooth – gasirea dispozitivelor

Toate dispozitivele descoperite prin Bluetooth.

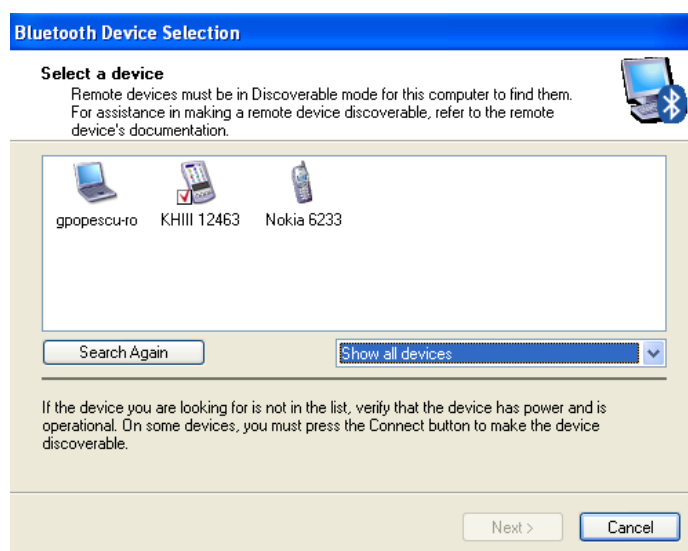


Figura 2.48: Toate dispozitivele descoperite prin Bluetooth

Nu toate serviciile vor fi active: diferentiere fata de tipul dispozitivului si posibilitatile de transmisie de date²⁰.

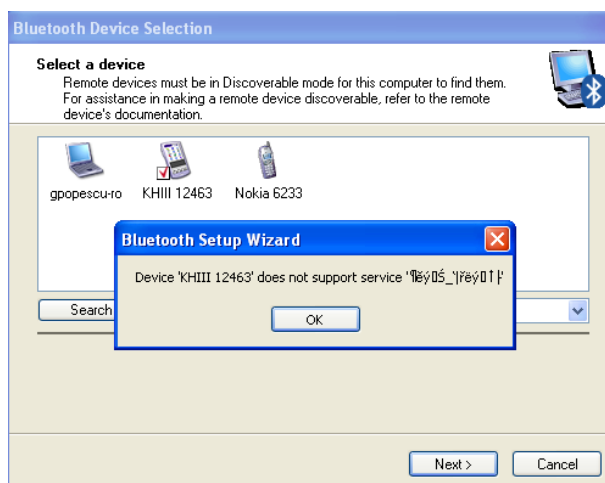


Figura 2.49: Limitari ale conexiunii

²⁰ Pentru un robot similar: YaMoR referinta este data de: Jaquier Cyril, Drapel Kévin, Ijspeert Auke, Crespi Alessandro, Andres Upegui, „Using Bluetooth to Control a YaMoR Modular Robot”, 2008.

Meniul de conexiune Bluetooth va oferi cateva caracteristici. Fiecare dintre acestea va fi analizata in continuare, pe scurt.

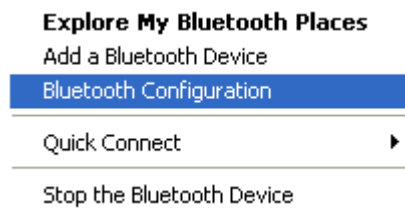


Figura 2.50: Configuratia Bluetooth

“Tab”-ul **general** al configuratiei Bluetooth va afisa, in partea de sus, datele de tip si versiune ale dispozitivului de conexiune Bluetooth USB.

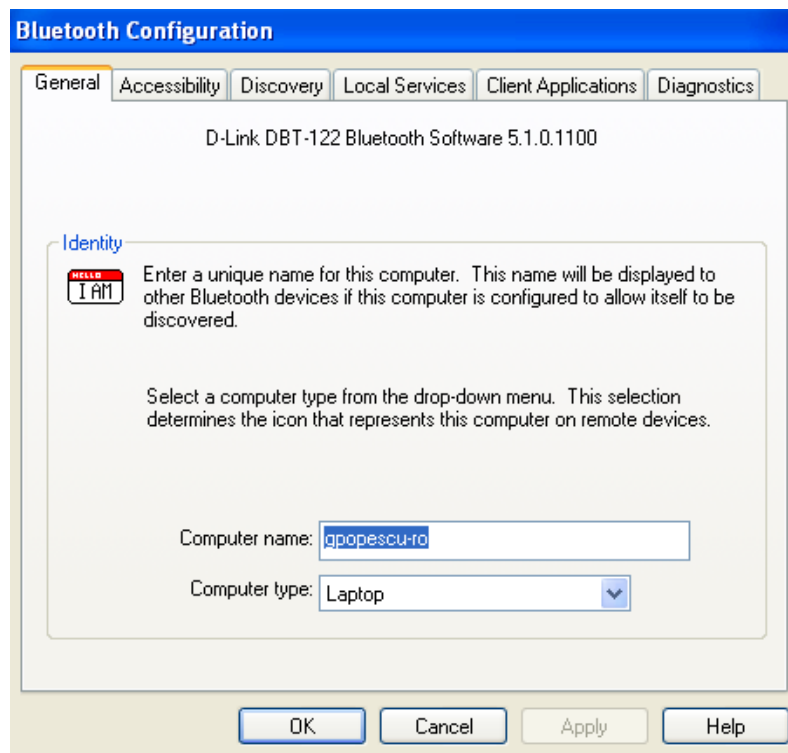


Figura 2.51: Informatii din “tab”-ul “General”

In “tab”-ul de **accesibilitate** va putea fi bifata optiunea ca alte dispozitive Bluetooth sa descopere computer-ul²¹.

²¹ Controlul la distanta folosind un dispozitiv PDA diferit va crea alti parametrii de initializare conexiune: Terrence W. Fong, Chuck Thorpe, Betty Glass, „PDA driver – A handheld system for remote driving”, IEEE International Conference on Advanced Robotics, July 2006.

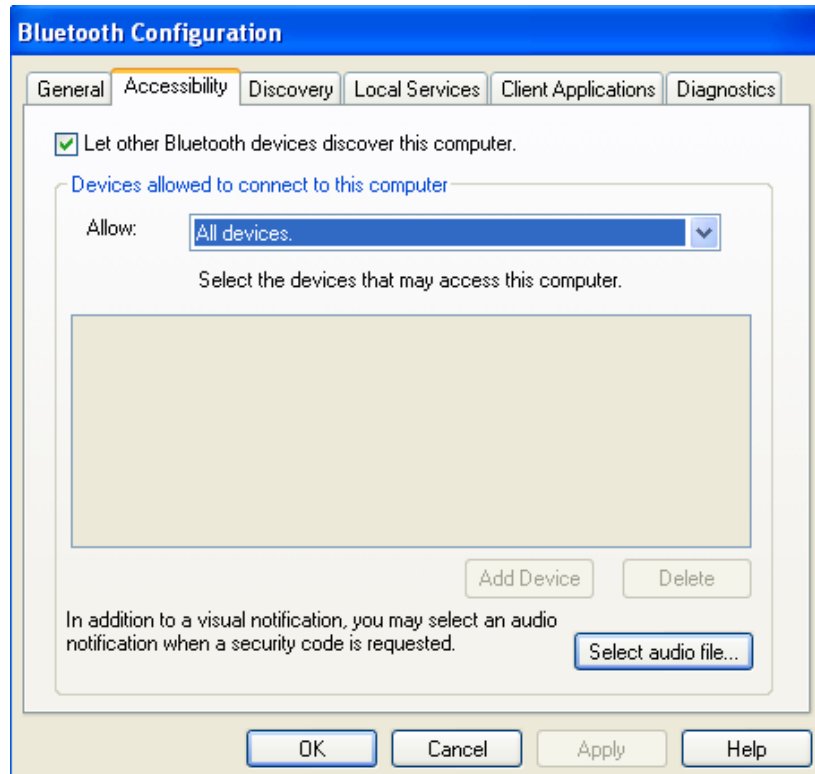


Figura 2.52: Informatii din “tab”-ul “Accesibilitate”

In “tab”-ul urmator vor putea fi gasite dispozitivele Bluetooth²².

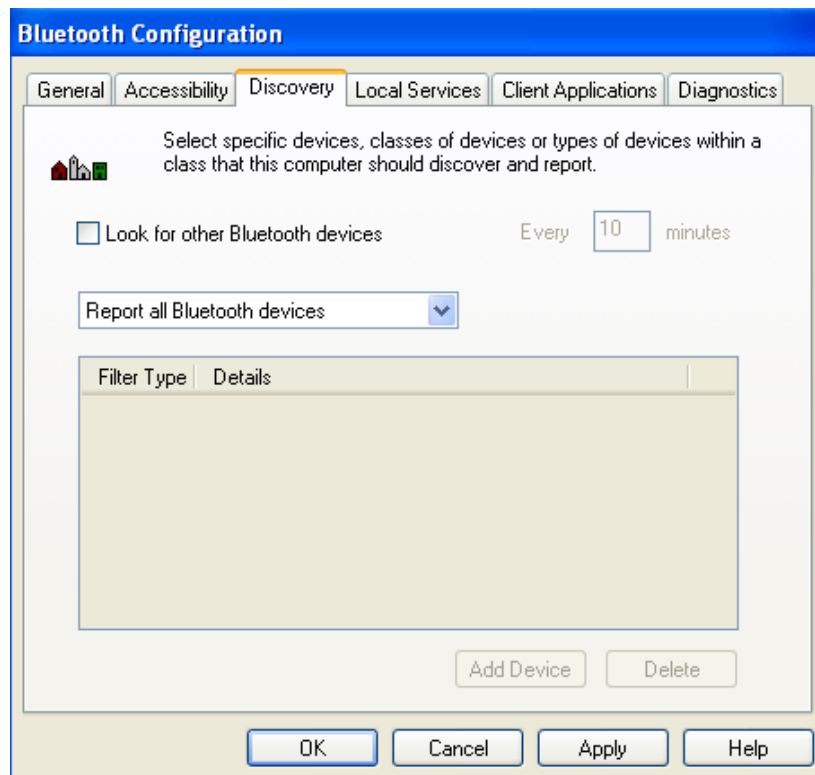


Figura 2.53: Informatii din “tab”-ul “Discovery”

²² Alte incercari: Mayer H., Schilling K. Mayer H. and Harmo P. Halme A, „Mobile robot control via Bluetooth technology”, Zurcher Hochschule Winterthur, IEEE, 2008.

În “tab”-ul de **servicii locale** vor fi afișate numele fiecărui serviciu, modalitatea de start, dacă este necesară o conexiune securizată și, în cazul sincronizării – “Active Sync” – portul de comunicație (**COM4**).

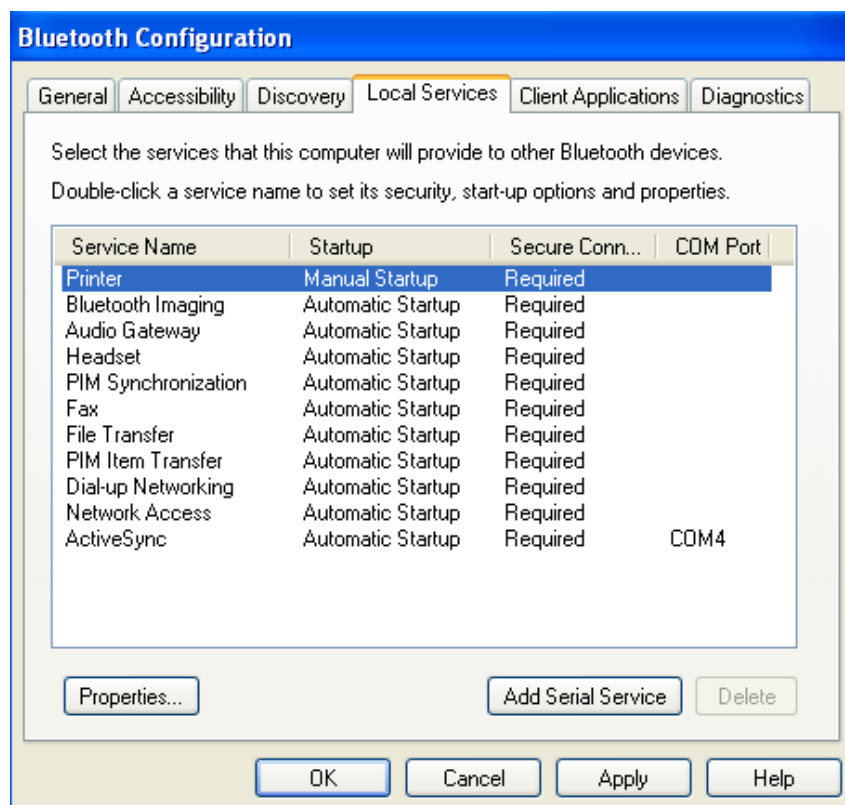


Figura 2.54: Informații din “tab”-ul “Servicii locale”

Accesând sub-“tab”-ul de **proprietăți** pentru ultima înregistrare de mai sus vom observa că singurul port de comunicație deschis este COM4.

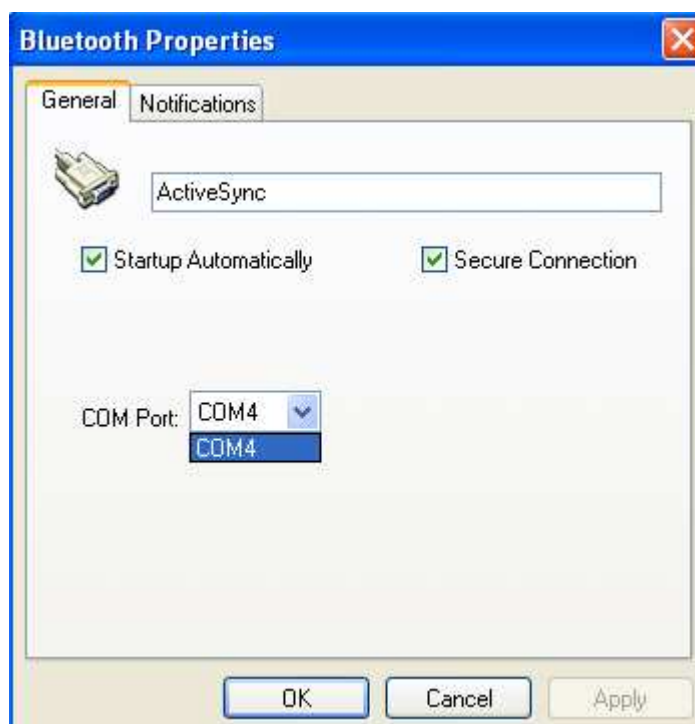


Figura 2.55: Informații din sub-“tab”-ul “Proprietăți”

În “tab”-ul de **aplicatii client** ultima înregistrare este cea care ne interesează: portul serial de comunicație Bluetooth care este activ prin **COM5**.

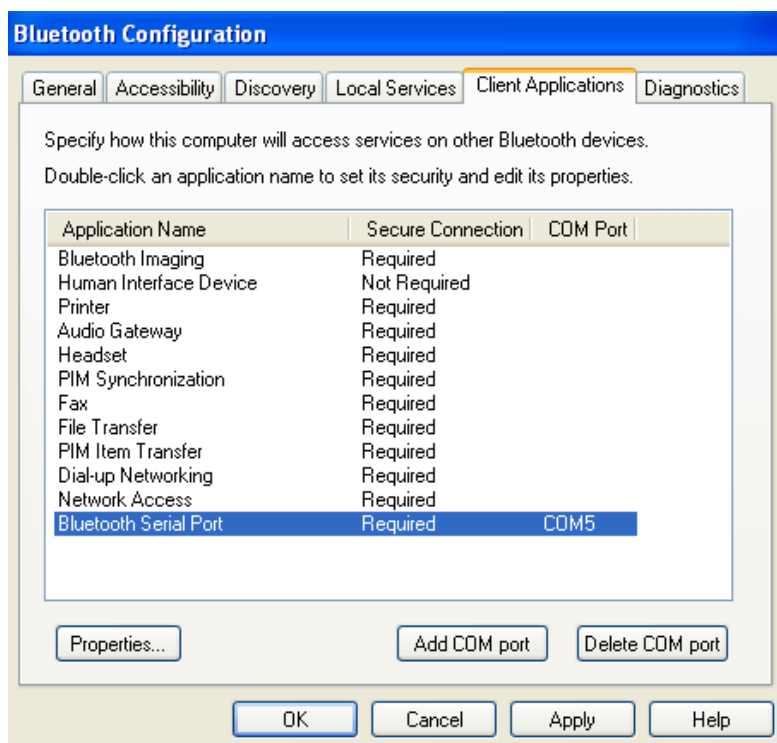


Figura 2.56: Informații din “tab”-ul “Aplicații client”

Ultimul “tab” este cel de **diagnostic**. Aici vor fi afișate informații despre dispozitivul Bluetooth USB (D-Link DBT-122): status, producător, proprietăți, adresa acestuia etc.

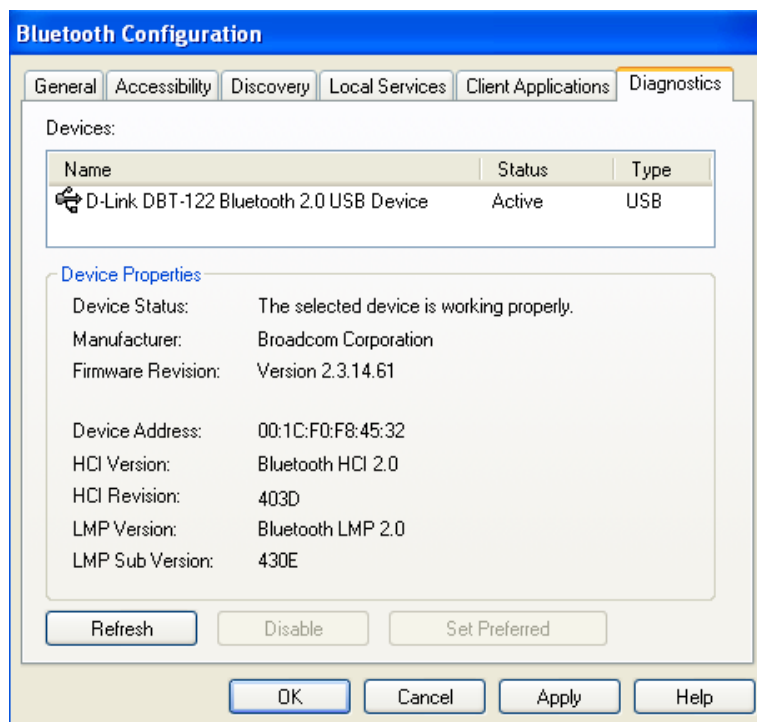


Figura 2.57: Informații din “tab”-ul “Diagnostic”

Accesand acum interfata robotului vom putea seta portul de comunicatie din lista initiala, setand asadar valoarea pentru **COM5**²³.

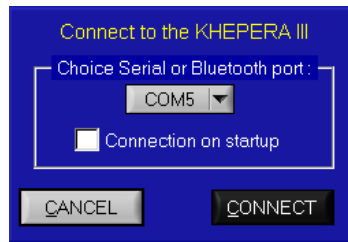


Figura 2.58: Portul de comunicatie ales: **COM5**

Una dintre erorile de comunicatie dintre robot si computer care pot aparea este "I/O operation timed out".

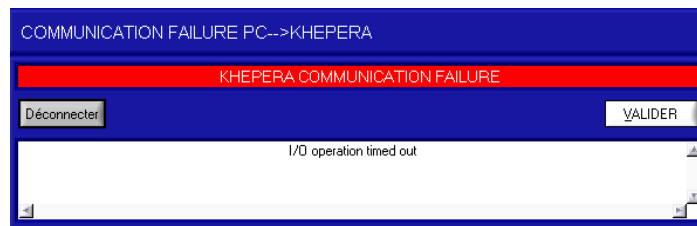


Figura 2.59: Eroare de comunicatie: "I/O operation timed out"

2.6 Interfata Activa Kephera III

Dupa setarea tuturor parametrilor de instalare si configurare Bluetooth a robotului in meniul principal se vor observa diferite valori pentru senzorii infrarosu, ultrasonici, precum si valorile pentru viteza si pozitie pentru ambele roti.

In partea din dreapta sus a interfetei grafice se observa butonul de activare a robotului. La trecerea din 0 in 1 rotile robotului vor intra in functiune.



Figura 2.60: Modul "Demo" inactiv; robotul nu se misca – stare "pasiva"

²³ Referinta generala folosita pentru descrierea comunicatiei Bluetooth intre robot si computer: Choo, S. H., Shamsudin H. M, Amin N., Faisal, C. F., Yeong, „Bluetooth transceivers for full duplex communication in mobile robots”, Universiti Teknologi Malaysia, 2007.

George POPESCU – Comandă vocală Kephera III

In figura modul "Demo" este activat iar bateria roborului se afla la 100% din capacitate.

Deasemenea se pot observa diferite valori pentru viteza. Diferente mici exista intre valorile pentru cele 2 roti cauzate de suprafata pe care se deplaseaza robotul (frecare) si distanta pana la urmatorul obstacol.

Observatie: In modul "Demo" Kephera III va incerca sa evite obstacolele intalnite in cale. Valorile masurate de senzori vor fi "instantanee", in masura capacitatilor tehnice si a timpului de refresh, conform cartii tehnice.

In acest moment valorile pentru viteza si acceleratie sunt cele prestabilite sau "default". Senzori vor inregistra noi valori date de mediul inconjurator care se vor modifica automat in interfata grafica.



Figura 2.61: Modul "Demo" activ; robotul se afla in miscare – stare "activa"

Meniul permite setarea a diferite valori pentru pozitie si viteza. Implicit aceste valori sunt 0²⁴.

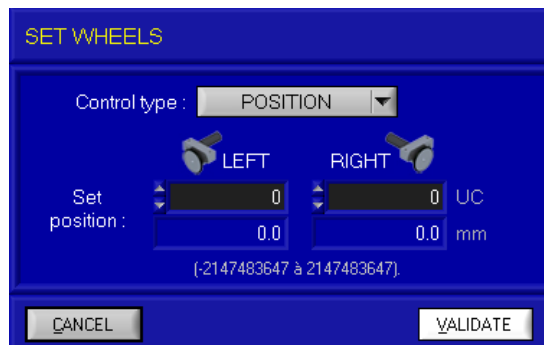


Figura 2.62: Setarea pozitiei cu valori cuprinse intre cele indicate mai sus

Instantierea a oricaror valori pentru pozitie se va putea face inserand direct valori numerice.

²⁴ Mai multe interfețe avansate de comandă vocală sunt clasificate și detaliate în: Terrence W. Fong, Chuck Thorpe, C. Baur, „Advanced interfaces for vehicle teleoperation: collaborative control, sensor fusion displays and web-based tools”, Autonomous Robots, July 2006.

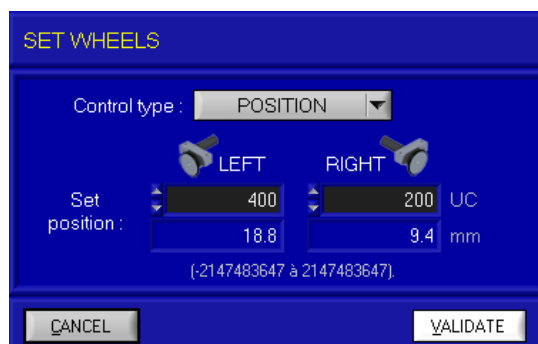


Figura 2.63: Exemplificare pentru setarea pozitiei 1

Conversia intre valoarea UC si mm se va realiza automat in fereastra initiala. Pentru “dreapta” putem pastra valoarea nula.

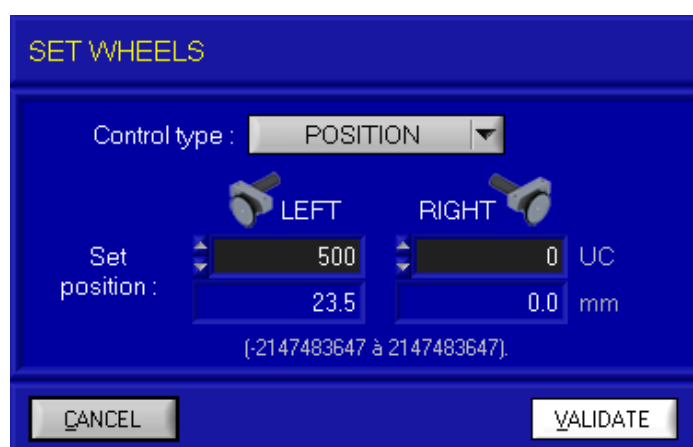


Figura 2.64: Exemplificare pentru setarea pozitiei 2

In cadrul meniului general in partea din dreapta sus a interfeței grafice exista butonul pentru **oprirea rotilor**.



Figura 2.65: Oprirea rotilor

Urmatorul buton este cel de resetare a encoder-ului.



Figura 2.66: Resetarea “encoder”-ului

O alta optiune pentru controlul robotului este aceea de ajustare a parametrilor rotilor.



Figura 2.67: Ajustarea parametrilor rotilor

Fereastra care se deschide la apasarea acestui buton va permite modificarea valorilor default (care pot fi observate in imaginea de mai jos) prin ajustarea constantelor **PID**.



Figura 2.68: Setarea parametrilor rotilor PID

Robotul are in dotare si un set de 5 perechi de senzori ultrasonici. Butonul urmatoare va permite vizualizarea valorilor acestora.



Figura 2.69: Senzorii ultrasonici

Figura urmatoare arata faptul ca niciunul dintre senzori nu este activ initial.



Figura 2.70: Senzorii ultrasonici “pasivi”

Vom activa mai intai perechile laterale de senzori: US1 si US5. Vor fi inregistrare numarul de ecouri, valoarea ecoului, distanta (in cm) si amplitudinea semnalului.



Figura 2.71: Senzorii ultrasonici activati: US1 si US5

In continuare vom activa toate cele 5 perechi de senzori. Diferitele valori masurate vor fi afisate in dreptul fiecareia.



Figura 2.72: Toti senzorii ultrasonici activati: US1..US5

La deplasarea robotului se vor observa diferente majore intre valorile indicate de fiecare pereche²⁵.

²⁵ Alte abordari pentru o interfatare grafica eficienta cu robotul: Grange S., „*Vision-based sensor fusion for active interfaces*”, Diploma report, Microengineering Department, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2005.



Figura 2.73: Alte valori pentru senzorii ultrasonici: US1..US5

Butonul urmator in este cel care va indica starea bateriei.



Figura 2.74: Starea bateriei

Butonul va activa fereastra care va afisa voltajul, intensitatea curentului, media acesteia, temperatura, etc.

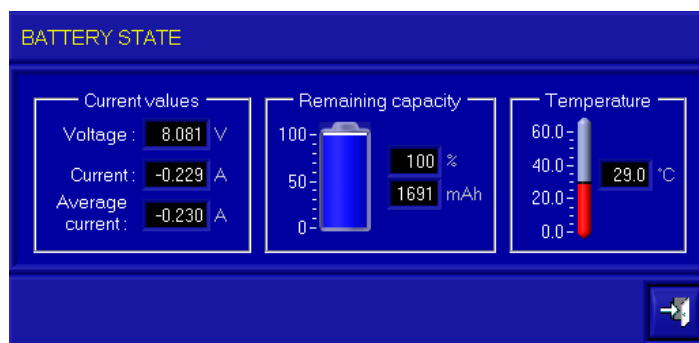


Figura 2.75: Detalii asupra bateriei

Urmatorul buton din meniul este cel de resetare a timpului.



Figura 2.76: Resetarea timpului

Butonul sub forma unui fisier .pdf este cel care face legatura cu manualul robotului.



Figura 2.77: Buton pentru manualul Kephera III

La apasarea acestuia se va deschide fisierul care insoteste robotul, respectiv cartea sa tehnica²⁶.

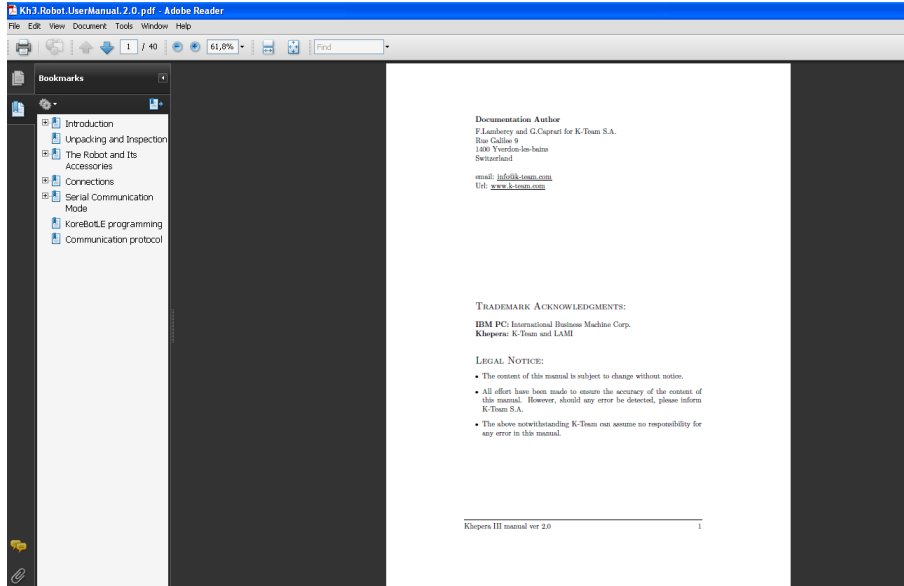


Figura 2.78: Manualul Kephera III

Penultimul buton este cel de “About”.

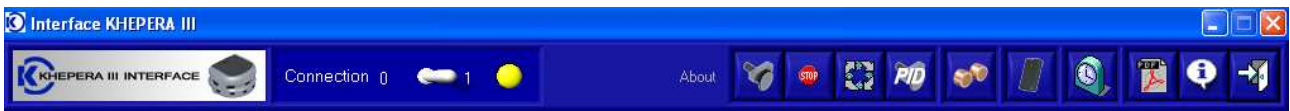


Figura 2.79: “About” Kephera III

Datele de contact ale K-Team se vor afisa precum in fereastra de mai jos.



Figura 2.80: Contact K-Team

²⁶ Alte abordari pentru crearea interfetei grafice: Terrence W. Fong, Chuck Thorpe, C. Baur, „Advanced interfaces for vehicle teleoperation: collaborative control, sensor fusion displays and web-based tools”, Autonomous Robots, July 2006.

George POPESCU – Comandă vocală Kephera III

Ultimul "item" al meniului este cel de iesire din aplicatie.



Figura 2.81: Iesire din aplicatie

3 PREZENTAREA PROBLEMEI: OBIECTIVUL DE CONDUCERE, MOD DE ABORDARE, MENIUL DE COMENZI

3.1 Generalitati

In domeniul comunicarii dintre om si calculator au existat de-a lungul timpului mai multe abordari. Cele mai multe dintre aceste s-au axat pe medierea efectului sonar asupra comenzii date prin esalonarea semnalelor si incadrarea lor intr-un domeniu propriu-zis admisibil. In momentul actual exista proiecte de cercetare, unele finalizate, altele in desfasurare, care au tratat problematica comenzii vocale intr-o plaja admisibila, fara a fi posibila sintetizarea vocii fiecarui individ uman. Astfel, utilizarea lor va fi limitata de gradul de precizie oferit pentru anumiti utilizatori. Unele dintre aceste sisteme sunt dezvoltate doar pentru o anumita limba sau functionalitate. Inca nu exista un sistem global compatibil cu toate diferentele care exista intre vocabularele mai multor limbi²⁷.

Dezvoltarile care au avut loc s-au indreptat catre minimizarea erorii de recunoastere a vocii si extensia modulelor de programe catre indivizi care prezentau o trasatura (sau mai multe) a vocii indepartata foarte mult de medie (folosita pentru teste). Utilizarea sistemelor de comanda vocala poate avea loc in medii industriale (pentru comanda robotilor industriali), in medii academice (educatie la toate nivelurile: elevi, studenti, cadre didactice), proiecte de cercetare, domeniul public etc.

In ultimele 3 decenii in special cercetatorii din domeniul recunoasterii vocii s-au concentrat pe dezvoltarea unui intreg sistem de aplicatii pornind de la zero desi, de cele mai multe ori, se dorea explorarea unui domeniu particular aplicativ. Sistemele de recunoastere vocala open-source precum HTK, ISIP, AVCSR si primele versiuni ale SPHINX au fost publicate pe internet. Acestea sunt in general optimizate pentru o singura abordare in legatura cu proiectarea unei arhitecturi. Ca rezultat, aceste aplicatii creeaza o bariera pentru proiectele din diferite alte domenii desi furnizeaza importante reurse software si conceptuale. Mai mult decat atat, unele dintre acestea fac imposibil accesul la fisierele sursa, fiind rezervate anumitor grupuri de cercetare; si sunt strict interzise mediului neacademic. Limbajul si mediul de programare folosit in principal de ultimele – si cele mai performante arhitecturi de recunoastere vocala – este Java Programming Language, incorporand multiple functionalitati software²⁸.

Conceptul propriu-zis prezinta o functionalitate modulara si este conceptul asemenea conform unei arhitecturi data de necesitati. Se doreste functionalitatea si usoara adaptare a sistemului dincolo de utilizarea lui actuala, facandu-l usor accesibil²⁹.

²⁷ Interconectarea mai multor sisteme de comanda vocala: Young S., Russell N.H., si Russell J.H.S., „*Token passing: A simple conceptual model for connected speech recognition systems*”, Cambridge University Engineering Dept, UK, Tech. Rep. CUED/F-INFENG/TR38, 2007.

²⁸ Cartea urmatoare este una de referinta in privinta sistemelor de recunoastere comanda vocala: Lee K.F., Hon H. W., si Reddy R., „*An overview of the SISTEM speech recognition system*”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 1, pp. 35–45, Ian. 2005.

²⁹ Arhitectura DARPA: Placeway P., Chen S., Eskenazi M., Jain U., Parikh V., Raj B., Ravishankar M., Rosenfeld R., Seymore K., Siegler M., Stern R., si Thayer E., „*The 1996 HUB-4 Sistem-3 system*”, in *Proceedings of the DARPA Speech Recognition Workshop*. Chantilly, VA: DARPA, Feb. 2007.

3.2 Arhitectura

Primele arhitecturi folosite pentru procesarea comenzilor vocale au avut la baza sisteme suportand dialog VoiceXML care se dovedeau a fi prea restrictive³⁰.

Abordarea traditionala pentru dezvoltarea unui sistem de comanda vocala este aceea de a cre un sistem optimizat pentru o anumita metodologie. Precum s-a dovedit in proiecte anterioare precum Dragon, Harpy, Sphinx si altele, aceasta abordare a fost incununata cu succes incat sistemele rezultate au produs metode fundamentale pentru recunoasterea vocii. In acelasi timp, alte cercetari au condus la solutionarea unei sarcini specifice: Baker a introdus modele Markov iar procesarea prin Sphinx a explorat diferite variante ale HMM pentru semnale discrete, continue sau combinate.

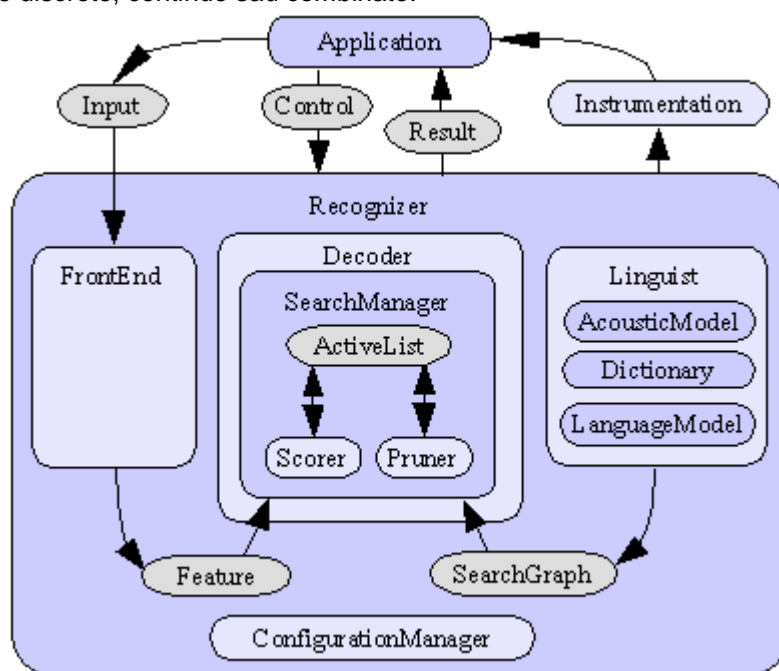


Figura 3.1: Arhitectura generala

Arhitectura generala de mai sus³¹ a fost proiectata pentru a oferi un grad inalt de flexibilitate si modularitate. Fiecare modul de mai sus poate fi implementat la randu-i sub diferite forme si in diferite medii de programare. Arhitectura este formata din 3 mari module: Interfata primara, Decodificatorul si Lingvistul. Primul modul primeste ca intrare unul sau mai multe semnale si le parametrizeaza dupa anumite trasaturi sau caracteristici. Modulul lingvistic traduce orice tip al modelului de limbaj impreuna cu informatia de pronuntie prin intermediul unui Dictionar si informatia structurala dintr-unul sau mai multe seturi de Modele acustice intr-un graf de cautare. Managerul de cautare al Decodificatorului foloseste caracteristicile semnalului date de Interfata impreuna cu grafal de cautare pentru a produce decodificarea si a genera rezultate. La orice moment anterior sau in timpul procesului de recunoastere, aplicatia poate folosi itemi de control pentru fiecare dintre module, devenind un partener sau ajutor in procesul de recunoastere vocala. Sistemul este asemenea mai multor sisteme de recunoastere vocala realizate pana in prezent in sensul ca are o multime de parametri configurabili, precum largimea fasciculului de cautare pentru a spori performanta sistemului. Managerul de Configurare al sistemului este folosit pentru a configura acesti parametri. Din

³⁰ Baker J.K., „The Dragon system - an overview”, in *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 23, no. 1, Feb. 2005, pp. 18–19.

³¹ Arhitectura generala prezentata a mai fost folosita si in alte proiecte precum: Jianhua T., Jian Yu, Lixing Huang, Fangzhou Liu, Huibin Jia, Meng Zhang, „The WISTON Text to Speech System for Blizzard 2008”, Institute of Automation, Chinese Academy of Sciences, China, 2008.

proces pot rezulta anumite date precum rate de eroare a recunoasterii cuvintelor, viteza de procesare a semnalului, procentajul de uz al memoriei etc³².

Interfata primara³³ (primul modul) are ca scop parametrizarea semnalului de intrare (audio) intr-o secventa de caracteristici. Precum se observa in figura urmatoare, acest modul cuprinde unul sau mai multe lanturi de comunicare a semnalelor paralele care sunt numite procesoare de date (DataProcessors). Oferirea mai multor lanturi paralele de procesare permite calculul mai multor parametri pentru acelasi sau mai multe semnale diferite.

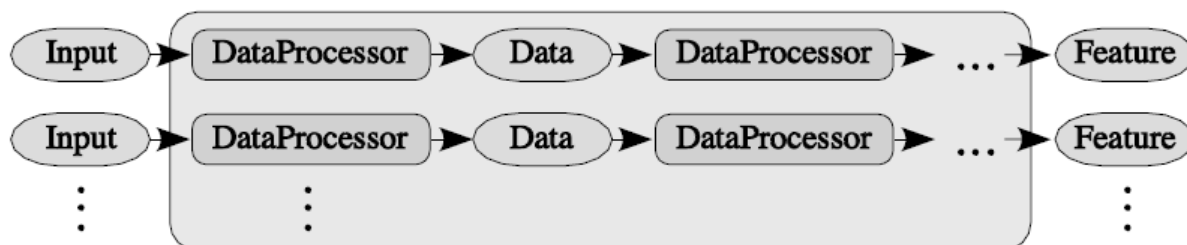


Figura 3.2: Interfata primara

Fiecare procesor de date conduce la o iesire care poate fi conectat mai departe la un sistem de prelucrare si identificare a semnalului³⁴. Intrarile si iesirile procesorului de date sunt reprezentate de date generice care incapsuleaza datele de intrare precum si identificatori asociati acestora care indica segmente de clasificare a datelor precum detectia dintre finalul unui cuvint si inceputul urmatorului. Procesorul de date al fiecarui lant este responsabil de producerea unui obiect de date compus din semnale parametrizate, care vor fi numite trasaturi, si vor fi folosite de catre Decodificator. Ca si sistemul AVCSR, sistemul permite abilitatea de a produce secvente paralele de trasaturi. Unicitatea sa consta din faptul ca permite un numar arbitrar de secvente paralele. Eficienta lui rezulta si din faptul ca Procesorul de date cere un anumit inpur de la un modul precedent lui doar atunci cand este nevoie, fara a supraincarca sistemul si a creste inutil timpul de executie si procesare a semnalului. In acest fel utilizatorii pot face interogari inainte si inapoi in timp pentru a vedea care au fost comenzile transmise. Acest lucru nu doar permite Decodificatorului sa opereze cautari sincrone in baza de date ci si sa opereze alte tipuri de cautari folosind algoritmi din Inteligenta Artificiala precum cei de cautare in latime, in adancime sau A*. Implementarea curenta include suport pentru urmatoarele evenimente: citirea datelor de intrare de la un sistem audio, sa aplice transformata cosinus discreta, transformata fourier pentru un semnal audio, sa calculeze distante Hamming intre sistem semnalul primit ca parametru si cea mai potrivita dintre iesiri³⁵.

Lingvistul: Genereaza graf de cautare ce va fi folosit de catre decodificator in timpul cautarii, in acelasi timp ascunzand nivelele sporite de complexitate ce rezulta din generarea acestui graf. Modulul lingvistic este unul care poate functiona dinamic, in sensul ca mai multi oameni pot contribui simultan pentru implementarea sistemului prin comenzile date. O implementare tipica a implementarii lingvistului presupune constructia grafului de cautare folosind un limbaj structural reprezentat de un model limbaj si de o structura topologica a modelului acustic. Lingvistul poate deasemenea folosi un dictionar (de obicei un lexic de pronuntie) pentru a mapa cuvinte ale unui model de limbaj in secvente (propozitii, fraze, expresii). Atunci cand se genereaza un graf de cautare, Lingvistul poate deasemenea sa incorporeze sub-cuvinte sau silabe intr-un context de lungime variabila. Prin permiterea diferitelor implementari ale Lingvistului, sistemul permite

³² Descrierea generala a sistemului: Lee K.F., Hon H. W., si Reddy R., „An overview of the SISTEM speech recognition system”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 1, pp. 35–45, Ian. 2005.

³³ Interfata primara (front end) descrisa complet: Yapanel H. Umit, Hansen H.L. John, „Towards an intelligent acoustic front end for automatic speech recognition: built-in speaker normalization”, Center for Robust Speech Systems, Department of Electrical Engineering, University of Texas at Dallas, USA, May 2008.

³⁴ Yamagishi J., Zen H., Wu Y.-Y., Toda T., Tokuda K., „The HTS-2008 System: Yet Another Evaluation of the Speaker-Adaptive HMM-based Speech Synthesis System in The 2008 Blizzard Challenge”, HTS working group: Nagoya Institute of Technology, Japan / Nara Institute of Science and Technology, Japan / University of Edinburgh, UK, 2008.

³⁵ O arhitectura de referinta pentru sistemele vocale poate fi gasita in: Seneff S., Hurley E., Lau R., Pao C., Schmid P., Zue P., „Galaxy-II: A reference architecture for conversational system development”, *ICSLP98*, Sydney, Australia, 2008.

utilizatorilor sa furnizeze diferite module de configuratie pentru diferite sisteme si anumite cerere de recunoastere a semnalului. De exemplu, un simplu sistem care sa contina doar primele cifre intr-o aplicatie poate folosi un Lingvist simplu care mentine spatiul de cautare in intregime in memorie. Pe de alta parte, o aplicatie pentru dictare care sa contina 100.000 de cuvinte va folosi un Lingvist mult mai complicat, amplu si sofisticat care va mentine in memorie doar o parte a acestuia accesand baza de date partitionat. Lingvistul consta din 3 componente fundamentale: Modelul de limbaj, Dictionarul si Modelul acustic.

A. Modulul modelului de limbaj³⁶

Modulul modelului de limbaj produce o structura bazata pe nivele a limbajului natural care poate fi reprezentata printr-un mare numar de implementari. Aceste, in mod tipic se incadreaza in una dintre urmatoarele categorii: Gramatici bazate pe un graf de conexiuni si modele stocastice. Graful obtinut prin intermediul gramaticii reprezinta un graf de cautare pentru un anumit cuvânt in care fiecare nod reprezinta un singur cuvânt si fiecare arc reprezinta probabilitatea ca o anumita tranzitie între cuvinte sa aibe loc. Pe de alta parte, modelul stocastic ataseaza probabilitati cuvintelor prin observatiile asupra a n-1 cazuri de utilizare.

Sistemul suporta o varietate de formate, printre care:

- Gramatica bazata pe o lista de cuvinte: defineste o gramatica oarecare ce are la baza o lista creata de utilizator prin alegerea unor cuvinte oarecare.
- Gramatica ce suporta recunoastere prin intermediul limbajului Java (Java Speech) pentru un format JSGF API care va fi definit pe baza platformei in care functioneaza sistemul, reprezentarea sistemului in format UNICODE, universal, etc.
- Gramatica LMG defineste o gramatica bazata pe un model statistic de limbaj. Gramatica LMG genereaza un nod al gramaticii pentru fiecare cuvânt si ofera rezultate bune cu digrame simple, bi sau mono reprezentate prin aproximativ 1000 de cuvinte.
- Gramatica FSTG: construita pe baza unei masini cu stari finite in formatul standard al unei gramatici ARPA FST.
- Model N-Gram: ofera suport pentru modele ASCII N-Gram in formatul ARPA (Address and Routing Parameter Area). Modelul N-Gram nu participa pentru optimizare valorilor de utilizare a memoriei astfel incat functioneaza cel mai bine cu modele de limbaj de dimensiuni reduse.
- Modelul Trigram: ofera suport pentru modele N-Gram generate de un model complex de limbaj statistic. Modelul Trigram optimizeaza valorile pentru stocare in memorie a datelor permitand functionarea cu fisiere mari si foarte mari de dimensiuni de 100MB sau mai mult.

B. Dictionarul

Dictionarul furnizeaza pronuntia necesara pentru cuvintele aflate in Modeul de limbaj. Pronuntia divide cuvintele in secvente/silabe sau sub-unitati ale cuvintelor aflate in modelul acustic. Interfata dictionarului ofera clasificarea cuvintelor si permite ca un singur cuvânt sa se gaseasca in mai multe clase. Sistemul ofera implementari ale interfetei dictionarului pentru anumite serii si secvente de pronuntie. Diferitele implementari optimizeaza procesul pentru mai multe sabloane de folosire bazate pe dimensiunea activa a vocabularului. De exemplu, o anumita implementare va inregistra in memorie intrugul vocabular la momentul de initializare a sistemului, in timp ce o alta va permite doar anumite pronuntii la cerere (sau la momentul in care este nevoie de acestea).

C. Modelul acustic

Modulul modelului acustic ofera o mapare între o unitate de limbaj si un item care poate fi inregistrat ca si scor/rezultat in comparatie cu o caracteristica a interfetei primare. Precum in cazul altor sisteme, maparea poate prelua informatii contextuale de pozitionare a cuvintelor. De exemplu, într-un anumit caz, contextul reprezinta singurul fenomen aflat la stanga si la dreapta unui anumit fenomen iar pozitionarea cuvântului determina faptul o anumita silaba se afla la inceputul, mijlocul sau sfarsitul unui anumit cuvânt (sau este un cuvânt in intregime). Definitia contextuala nu este fixata in cadrul sistemului, permitand pentru definirea modelelor acustice sa contina anumite „simboluri” acustice (introducere, incheiere – ce nu vor fi luata neaparat in calcul) precum si modele acustice ale caror contexturi nu trebuie sa fie adiacente aceleiasi unitati de limbaj. In mod tipic, Lingvistul divide fiecare cuvânt al dictionarului activ într-o secventa de silabe

³⁶ Abordari similare si exemple: Kwok L., „*A technique for the integration of multiple parallel feature streams in the Sistem-4 speech recognition system*”, Master’s Thesis (Sun Labs TR-2003-0341), Harvard University, Cambridge, MA, June 2005.

dependente de context³⁷. Apoi lingvistul paseaza aceste unitati si contextele aferente modelului acustic obtinand grafurile asociate lor. In cele din urma foloseste aceste grafuri in conjunctie cu modelul de limbaj pentru constructia grafului de cautare³⁸.

Spre deosebire de majoritatea sistemelor de recunoastere vocala, care reprezinta grafurile asociate ca o structura fixa in memorie, graful construit de sistem in aceasta lucrare este un graf direct de obiecte. In acest graf fiecare nod corespunde unei stari a cuvântului si fiecare arc reprezinta probabilitatea tranzitiei dintr-o stare in alta. Prin reprezentarea de mai sus (ca un graf de obiecte in locul unuia cu structura fixata), o implementare a modelului acustic poate oferi usor diverse topologii³⁹. De exemplu, interfața modelului acustic nu restrictioneaza nodurile grafului in termeni de numar de stari, numarul de tranzitii pentru fiecare stare sau directia unei tranzitii (inainte sau inapoi). Mai mult, sistemul permite ca numarul de stari ale grafului sa varieze de la o unitate la alta in cadrul aceluiasi model acustic. Fiecare stare este capabila sa produca un anumit rezultat pentru o trasatura observata. Codul pentru calculul acestui scor este realizat prin intermediul starii insasi – in acest fel ascunzand implementarea sa de restul sistemului, permitand chiar diferite functii de densitate de probabilitate pentru fiecare stare. Modelul acustic permite contributia mai multor componente la mai multe niveluri in acelasi timp. Aceasta inseamna ca aceste componente care formeaza o stare precum mixturi Gaussiene, matrici de tranzitie si ponderi de mixtura sa fie impartite de oricare dintre starile sistemului intr-un anumit grad.

Pentru restul sistemului, alti utilizatori pot configura sistemul pentru diferite implementari ale modelului acustic bazate pe necesitatile lor. Sistemul ofera o singura implementare pentru modelul acustic ce este capabila de incarcarea si folosirea modelelor acustice generate de un anumit trainer.

D. Graful de cautare

Desi mai multe module de Lingvist pot fi implementate in diferite feluri iar topologiile spatiilor de cautare generate de acesti Lingvisti pot varia foarte mult, spatiile de cautare sunt toate reprezentate de un Graf de Cautare. Asa cum este ilustrat in figura urmatoare, graful de cautare este structura primara de date folosita in timpul procesului de decodificare audio.

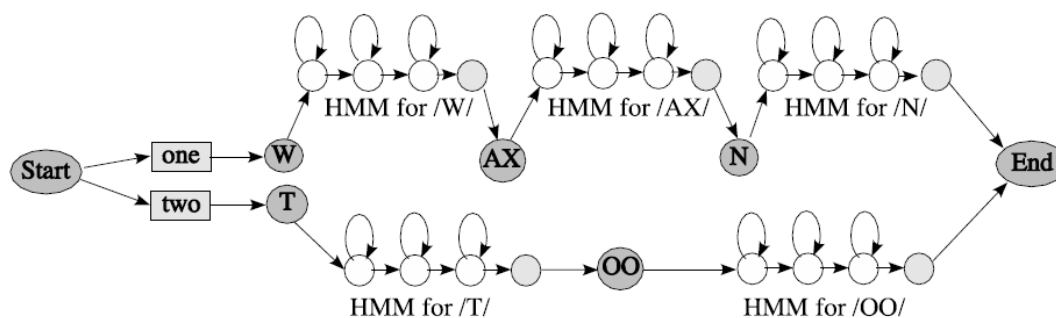


Figura 3.3: Graful de cautare

Nota: HMM = Hidden Markov Model (Model Markov Ascuns)⁴⁰

In figura de mai sus este prezentat un graf de cautare care este un graf orientat compus din stari de cautare optionale si arcuri ce detin o anumita probabilitate. Fiecare stare a grafului poate reprezenta componente ale

³⁷ Yamagishi J., Zen H., Wu Y.-Y., Toda T., Tokuda K., „*The HTS-2008 System: Yet Another Evaluation of the Speaker-Adaptive HMM-based Speech Synthesis System in The 2008 Blizzard Challenge*”, HTS working group: Nagoya Institute of Technology, Japan / Nara Institute of Science and Technology, Japan / University of Edinburgh, UK, 2008.

³⁸ Constructia unui graf de cautare folosind aproximari succesive si clustering: Raux, A., „*Automated Lexical Adaptation and Speaker Clustering based on Pronunciation Habits for Non-Native Speech Recognition*”, INTERSPEECH (ICSLP) 2008, Jeju Island, Korea.

³⁹ Unele dintre topologiile simple se afla descrise in: Seneff S., Hurley E., Lau R., Pao C., Schmid P., Zue P., „*Galaxy-II: A reference architecture for conversational system development*”, ICSLP98, Sydney, Australia, 2008.

⁴⁰ Modelul Markov implementat: Li X. X., Zhao Y., Pi X., Liang L.H., si V. Nefian A., „*Audio-visual continuous speech recognition using a coupled hidden Markov model*” in *Proceedings of the 7th International Conference on Spoken Language Processing*, Denver, CO, Sept. 2008, pp. 213–216.

modelului de limbaj (cuvintele din dreptunghiuri), dictionarul (sub-cuvinte sau silabe din cercurile inchise la culoare) sau modelul acustic (HMM-uri)⁴¹.

Graful este unul orientat in care fiecare nod, numit si stare de cautare, reprezinta fie o stare emitatoare, fie una ne-emitatoare. Starile emitatoare pot determina anumite rezultate impreuna cu anumite caracteristici acustice ale mesajului primit in timp ce starile ne-emitatoare sunt in general folosite pentru reprezentarea unor constructii lingvistice de nivel inalt precum cuvinte sau fenomene care nu sunt in mod direct evaluate alaturi de caracteristicile obiectelor de intrare. Arcele dintre stari reprezinta tranzitii posibile intre stari, fiecare avand asociata o probabilitate reprezentand sansa unei tranzitii prin acel arc⁴².

Interfata grafului de cautare este intentionat generica pentru a permite alegerea intr-o gama larga de implementari, oferind ipoteze si constrangeri descoperite in sistemele de recunoastere vocala precedente. In particular, Lingvistul nu plaseaza vreo restrictie asupra niciuneia dintre urmatoarele:

- Topologia generala a spatiului de cautare
- Marimea contextului fonetic
- Tipul gramaticii (stocastic sau bazata pe alte reguli)
- Adancimea modelului de limbaj N-Gram

O trasatura cheie a grafului de cautare este aceea ca implementarea unei stari de cautare nu trebuie in mod necesar fixata. In acest fel, implementarea fiecarui lingvist furnizeaza propria sa implementare a starilor de cautare care poate varia in functie de caracteristici particulare ale lingvistului. De exemplu, un lingvist simplu poate furniza un graf de cautare stocat in memorie in care fiecare stare de cautare este o mapare 1-la-1 pentru nodurile grafului aflat in memorie. Un lingvist care reprezinta un vocabular mare si complex, in schimb, poate construi o reprezentare interna compacta a grafului de cautare. In acest caz, lingvistul va genera un set de succesori pentru fiecare stare de cautare prin expandarea dinamica a acestei compactificari la o cerere a utilizatorului. Modul in care graful de cautare este construit determina amprenta asupra memoriei, viteza si acuratetea recunoasterii. Arhitectura modularizata a sistemului permite diferite strategii de compilare a grafului de cautare pentru a fi folosite fara a schimba alte aspecte ale sistemului. Alegerea intre constructia statica si cea dinamica a unui limbaj HMM (Hidden Markov Model) depinde in principal de dimensiunea vocabularului, complexitatea modelului de limbaj si amprenta asupra memoriei sistemului si poate fi realizata prin intermediul aplicatiei.

E. Implementari ale sistemului

Odata cu interfata primara, sistemul ofera cateva implementari ale lingvistului pentru a indeplini diferite sarcini. Lingvistul „plat” este potrivit pentru sarcinile de recunoastere care folosesc o gramatica independenta de context (CFG – Context Free Grammar), gramatici cu stari finite (Finite-State Grammars – FSG), traductor cu stari finite (Finite-State Transducer – FST) si modele reduse de limbaj N-Gram. Lingvistul „plat” converteste oricare dintre aceste formate de modele de limbaj externe intr-o structura de gramatica interna. Gramatica reprezinta un graf de cuvinte in care fiecare nod a gramaticii reprezinta un singur cuvant si fiecare arc al grafului reprezinta probabilitatea ca o tranzitie sa aiba loc. Pe de alta parte, lingvistul „plat” genereaza graful de cautare din acest graf al gramaticii intern, stocand intregul graf de cautare in memorie. In acest fel lingvistul „plat” este deosebit de rapid, totusi dovedind dificultati in manevrarea gramticilor cu factori multipli pentru fiecare ramura⁴³.

Lingvistul „plat” dinamic este similar cu cel „plat” in aceea ca este potrivit pentru sarcini de recunoastere similare. Principala diferenta consta in faptul ca primul creaza in mod dinamic graful de cautare la cererea utilizatorului, oferind capabilitatea de manevrare de gestiune a unei gramatici neorientate. Cu aceasta capabilitate, totusi, se prefigureaza un cost al unei scaderi modeste a performantei timpului de executie. Arborele de cuvinte al vocabularului lingvistului (LexTreeLingvist) este potrivit pentru un vocabular de dimensiune mare si sarcinile de recunoastere derivate de aici care folosesc modele complexe N-Gram.

⁴¹ Metode de cautare in grafuri de identificare acustica: Deshmukh N., Ganapathiraju A., Hamaker J., Picone J., si Ordowski M., “A public domain speech-to-text system,” in *Proceedings of the 6th European Conference on Speech Communication and Technology*, vol. 5, Budapest, Hungary, Sept. 2006, pp. 2127–2130.

⁴² Identificatori probabilistici in recunoasterea semnalului vocal pot fi gasiti in: Glass J.R., „*A probabilistic framework for segment-based speech recognition*”, *Computer Speech and Language*, vol. 17, no. 2, pp. 37–152, Apr. 2006.

⁴³ Generalitati dar si implementari ale sistemului: Lee K.F., Hon H. W., si Reddy R., „*An overview of the SISTEM speech recognition system*”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 1, pp. 35–45, Ian. 2005.

Ordinul N-Gram este arbitrar iar arborele lingvistului ofera posibilitatea decodificării N-Gram. Acesta organizează cuvintele sub forma unui arbore de cuvinte, o metoda compacta de reprezentare a vocabularelor mari. Deasemenea foloseste acest arbore pentru a genera in mod dinamic starile de cautare, permitand gestionarea unor vocabulare mari si foarte mari folosind doar o mica parte din memorie. Totodata suporta si modelele de limbaj ASCII si binar generate de uneltele statistice de modelare a limbajului⁴⁴.

Decodificatorul⁴⁵:

Principalul rol al decodificatorului este acela de a utiliza caracteristicile obtinute de catre interfata primara in conjunctie cu graful de cautare din partea lingvistului pentru a genera anumite ipoteze pentru un anumit rezultat. Blocul de decodificare cuprinde un manager de cautare si alte surse de cod care simplifica procesul de decodificare pentru o anumita aplicatie. In acest fel, cea mai interesanta componenta a blocului este Managerul de Cautare. Decodificatorul propune managerului sa recunoasca un set de trasaturi pentru cadre ale semnalului audio. La fiecare pas al procesului Managerul creeaza un obiect rezultat care contine toate caile care au fost atinse spre o stare finala ne-emitatoare. Pentru procesarea rezultatului sistemul foloseste capabilitati de creare a unei latici si scoruri specifice pentru un anume rezultat. Spre deosebire de alte sisteme, totusi, aplicatiile pot modifica spatiul de cautare si obiectele rezultate intre pasi, permitand aplicatiei sa devina un partener in procesul de recunoastere. Ca si lingvistul, managerul de cautare nu este restrictionat la o implementare particulara.

De exemplu, implementari ale managerului de cautare poate si opereze algoritmi de cautare precum cei de tip Viterbi sincroni, A*, bi-directionali si asa mai departe. Fiecare implementare a unui manager de cautare foloseste un algoritm pe baza de jeton. Un asemenea obiect care este asociat cu o stare de cautare contine scorurile generale ale cailor catre un anumit nod, o referinta catre starea de cautare, una catre o trasatura de intrare si alte informatii relevante. Referinta catre starea de cautare permite managerului de cautare sa relationeze cu jetonul prin starea sa de distributie, o unitate fonetica bazata pe context, pronuntie, alte cuvinte si starea gramaticii. Fiecare ipoteza partiala se termina cu un jeton activ. Implementari ale managerului de cautare pot construi un set de jetoane active in forma unei liste active la fiecare pas de timp, prin intermediul uneia existente in prealabil.

- `SimpleBreadthFirstSearchManager`: inregistreaza caracteristici simple sincrone folosind cautare Viterbi prin functii ce sunt apelate pentru fiecare cadru in parte.
- `WordPruningBreadthSearchManager`: inregistreaza o cautare sincrona Viterbi pentru fiecare cadru.
- `BushderbySearchManager`: inregistreaza o cautare de tip breadth-first (in latime) folosind un algoritm de tip Bushderby si efectuand clasificari bazate pe anumite distante si masuratori.
- `ParallelSearchManager`: inregistreaza mai multe cadre pentru o cautare Viterbi pentru mai multe cai si caracteristici folosind un limbaj HMM

F. Alte dezvoltari si aplicatii ale sistemului

Structura modulara a sistemului ne permite sa efectuam atat implementari de tip paralel sau Bushderby ale managerului de cautare imediate care, in acelasi timp, nu au nevoie de modificari ale celorlalte componente ale sistemului. Natura modulara permite abilitatea de a folosi aceste module ale caror implementari sunt de la cele mai generale pana la aplicatii specifice pe baza de algoritmi. De exemplu, putem imbunatati timpul de executie al aplicatiei prin teste de regresie si inregistrari repetate ale comanda vocale cu 2 ordine de marime fie si numai prin introducerea unui nou lingvist si pastrand aceleasi coordonate pentru restul sistemului⁴⁶.

Mai multe decat atat, modularitatea sa deasemenea permite suport pentru o varietate de task-uri. De exemplu, mai multe implementari ale managerului de cautare permit sistemului sa ofere suport pentru sarcini

⁴⁴ Raux, A., „Automated Lexical Adaptation and Speaker Clustering based on Pronunciation Habits for Non-Native Speech Recognition”, INTERSPEECH (ICSLP) 2008, Jeju Island, Korea.

⁴⁵ Pentru un sistem similar (CMU): Lamere P., Kwok P., Walker W., Gouvea E., Singh R., Raj B., si P. Wolf, „Design of the CMU Sistem-4 decoder”, in *Proceedings of the 8th European Conference on Speech Communication and Technology*, Geneve, Switzerland, Sept. 2003, pp. 981–1184.

⁴⁶ Raux, A., „Automated Lexical Adaptation and Speaker Clustering based on Pronunciation Habits for Non-Native Speech Recognition”, INTERSPEECH (ICSLP) 2008, Jeju Island, Korea.

care se întind de la un vocabular mic precum cel digital până la unul care să cuprindă chiar și 64.000 de cuvinte. Într-un alt exemplu, lingvistul poate oferi diferite sarcini de implementare, precum cele bazate pe o gramatică independentă de context și aplicații bazate pe modele stocastice de limbaj⁴⁷.

Natura modulară a sistemului a fost permisă prin programare orientată pe obiecte în limbajul de programare Java. În particular, capacitatea de a încărca un anumit modul sau mai multe linii de cod în momentul execuției permite o configurare ce poate fi modelată de către utilizator în orice moment. În Java, construcția interfețelor permite separarea comenzii vocale de către designul propus pentru implementare. Prin alegerea programării în limbajul Java pot fi deduse și alte avantaje:

- Sistemul de comandă vocală poate rula pe o gamă extinsă de platforme fără a fi nevoie de recompilarea fișierelor
- Multimea bogată de fișiere API reduce timpul de codare și formatare a liniilor de cod
- Suportul construit în interiorul aplicației pentru multi-threading (lansarea în execuție a mai multor fire de execuție în același timp) creează facilitatea de a experimenta fără a modifica sarcinile de decodificare de-a lungul mai multor fire de execuție
- Modulul de colectare automată a „deseurilor” de programare ajută dezvoltatorii să se concentreze pe dezvoltarea ulterioară a algoritmilor folosiți în locul scăparilor de memorie.

Pe de altă parte, platforma Java poate crea probleme legate de amprente ale memoriei. De asemenea, în același domeniu al memoriei, unele motoare de comandă vocală vor accesa platforma direct pentru a optimiza consumul de memorie în timpul decodificării. Accesul direct la modelul de memorie al platformei nu este permis prin intermediul limbajului de programare Java. O idee des vehiculată în rândul specialiștilor IT, programatori în Java, este aceea că acest limbaj este prea încet. În momentul rulării aplicației am avut în vedere programarea modulară și aspecte legate de sistem pentru consistență, comparând rezultatele cu alte sisteme de comandă vocală realizate până în prezent.

În rezultata interesantă obținută prin compararea diferitelor tipuri de vocabulare ajută pentru demonstrarea puterii arhitecturii modulare. Datorită acestui aspect sunt posibile diferite implementări ale lingvistului și ale managerului de căutare care au fost optimizate pentru sarcini specifice permițând sistemului obținerea unor performanțe remarcabile, în special sub aspectul timpilor de execuție.

Un alt aspect interesant al studiilor de performanță arată că prin execuția rand cu rand programul nu consumă resurse deosebit în context dezvoltării hardware din prezent. Pentru rezultatele cu cele 2 procesoare (sintetizate în tabel) am folosit sistemul de scor care divide proporțional procentul de folosire pentru fiecare procesor în parte. În timp ce creșterea în viteză este importantă, totuși nu este atât de semnificativă pe cât s-ar fi așteptat. Analize ulterioare au ajutat la dezvoltarea concluziei privind faptul că 30% din timpul de folosire al procesorului s-a consumat în timpul procesului de încadrare a rezultatelor. Celelalte 70 de procente sunt consumate în timpul altor activități precum formarea listei active și a grafului de stări de căutare. Rezultatele empirice obținute arată că mecanismul de colectare a deseurilor (garbage collection mechanism) modifică cu doar cu 2-3 procente din performanțele procesorului⁴⁸.

3.2.1 Conexiunea prin intermediul microfonului

Pentru realizarea comenzii vocale dintre Kephra III și utilizatorul uman vom folosi un microfon conectat la placa de sunet a computer-ului prin intermediul căruia vor putea fi înregistrate apelurile vocale.

3.2.2 Modulul de recunoaștere a comenzii

Modulul de recunoaștere a vocii va trebui să realizeze identificarea semnalului venit de la utilizator în condiții normale de emisie, folosind repere standard ale vocii umane (ton, ritm, etc.). Procesul de identificare presupune sincronizarea cu o bază de date de recunoaștere divizată în masculin / feminin. Este de așteptat ca această separare să crească acuratarea sistemului. Colectarea datelor din bază de date poate avea ca

⁴⁷ Yamagishi J., Zen H., Wu Y.-Y., Toda T., Tokuda K., „*The HTS-2008 System: Yet Another Evaluation of the Speaker-Adaptive HMM-based Speech Synthesis System in The 2008 Blizzard Challenge*”, HTS working group: Nagoya Institute of Technology, Japan / Nara Institute of Science and Technology, Japan / University of Edinburgh, UK, 2008.

⁴⁸ Lee K.F., Hon H. W., și Reddy R., „*An overview of the SISTEM speech recognition system*”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 1, pp. 35–45, Ian. 2005.

reper un singur utilizator sau mai multi. Atunci cand o noua persoana doreste sa participe la o comanda vocala modulul de recunoastere va face posibila identificarea pe baza (mediei) sabloanelor prestabilite⁴⁹.

3.2.3 Analiza si modelarea limbajului

Modelul foloseste la baza un procedeu statistic de recunoastere a limbajului. Rezultatul este analizat separat intr-un nou modul bazat pe o gramatica independenta de context permitand sistemului sa nu ia in calcul anumite cuvinte necunoscute rezultand astfel un rezultat cu succes desi analiza a fost realizata doar partial.

In mod ideal este de dorit ca interpretarea statistica a limbajului sa aiba loc pe baza unui dialog corespunzator unei anumite sarcini a robotului (de exemplu miscare inainte cu o anumita viteza, miscare in cerc, miscare pentru o anumita perioada de timp, etc.). Poate fi astfel creata o baza de date ajutatoare in care sa fie stocate informatii cu privire la perioadele de timp ale miscarii, locurile din camera detectate de senzori, aparitia unui cuvânt cheie (keyword) in limbaj iar procesul imediat urmator sa fie, de exemplu, de oprire a rotilor Kephera III.

Primii pasi facuti pentru constructia modulului de analiza a limbajului este, asadar, constructia unei gramatici, apoi generarea unui nou modul de text „artificial” pentru analiza si in cele din urma instruirea modelului statistic de recunoastere a comenzii pe baza modulului „artificial”.

Descrierea gramaticii are la baza comenzile posibile trimise catre robot astfel incat utilizatorul poate alege dintre acestea in prima instanta. Regulile gramaticii folosite pentru identificarea anumitor cuvinte cheie precum: „inainte”, „inapoi”, „stanga”, „dreapta” au fost generate automat folosind baza de cunostinte.

Pentru a determina ca analiza sa fie suficient de robusta pentru a analiza propozitiile / expresiile gresite din punct de vedere gramatical sau de neinteles din punct de vedere al semnalului de intrare continutul gramaticii a fost mentinut cat mai general cu putinta. Totusi, atunci cand este folosita pentru recunoasterea vocii, o gramatica foarte generala produce un volum foarte mare de sintagme negramaticale sau formulari nenaturale. Astfel, am modificat continutul gramaticii pentru a se potrivi cu generarea comenzii vocale atasand auxiliar ponderi ale regulilor deduse din observatii care decurg natural din conversatia robot-computer (microfon => persoana). Am introdus deasemenea si ponderi pentru cuvintele cheie necesare (inainte, inapoi, etc.) in functie de cat de frecventa este folosirea acestora de catre anumiti indivizi, in concordanta cu observatiile precedente si preferinte.

Prin intermediul gramaticii modificate, am generat un corp de un anumit numar de cuvinte si expresii suficient de mare pentru a acoperi majoritatea comenzilor vocale ce ar putea fi date catre Kephera III.

La ora actuala exista o multitudine de aplicatii software care trateaza problema pronuntiei si a personalizarii comenzii vocale. Exista vocii (tonuri si ritmuri) predefinite care pot interpreta un anumit text introdus de la tastatura⁵⁰. In cazul comenzii vocale catre Kephera III am dorit sa realizam realizarea unei actiuni in timp real si identificarea comenzii astfel incat robotul sa reactioneze doar la anumite voci ale unor utilizatori agreati.

Unul dintre celelalte obiective ale proiectului de comanda vocala este agela de a detecta formularile incorecte din punct de vedere gramatical precum si alte sunete introduse auxiliar. Modelul limbajului trebuie sa fie, pe de o parte, suficient de general pentru a accepta anumite propozitii, expresii si structuri si, pe de alta, sa identifice formularile partial corecte astfel incat comanda sa se realizeze⁵¹.

⁴⁹ Ravishankar M.K., „*Efficient algorithms for speech recognition*”, PhD Thesis (CMU Technical Report CS-96-143), Carnegie Mellon University, Pittsburgh, PA, 2006.

⁵⁰ Crearea oricarui lexic: Raux, A., „*Automated Lexical Adaptation and Speaker Clustering based on Pronunciation Habits for Non-Native Speech Recognition*”, INTERSPEECH (ICSLP) 2008, Jeju Island, Korea.

⁵¹ Ravishankar M.K., „*Efficient algorithms for speech recognition*”, PhD Thesis (CMU Technical Report CS-96-143), Carnegie Mellon University, Pittsburgh, PA, 2006.

De exemplu, solicitand ca robotul sa se deplaseze in directia „inainte”, utilizatorul trebuie sa se astepte ca aceasta actiune sa fie indeplinita desi robotul s-ar putea deplasa in directie inversa: cele 2 cuvinte cheie: „inainte” si „inapoi” au un semnal similar pe cand „stanga” si „dreapta” difera cu un procent mai mare.

Este important ca identificarea sa se realizeze pe baza unui proces de invatare si nu sa reprezinte doar un mod de acceptare sau rejectie a comenzii in functie de factorii de mediu, ritm, ton, gravitate etc. De aceea comparatia dintre semnalul curent si cel folosit ca sablon trebuie sa se realizeze printr-un proces de reactualizare a bazei de date (de cunostinte) prin intermediul comenzilor deja percepute.

3.2.4 Modulul de comanda (dialog) intre utilizator si Kephera III

Pentru dialogul dintre utilizatorul uman si robot este necesara constructia unui cadru generic pentru management al comenzilor. Modulul va permite executia unei comenzi pe baza comenzilor precedente si a specificatiilor inregistrate. In cel mai complex caz, modulul de dialog trebuie construit sub forma unui arbore de dialog alcatuit in functie de specificatiile primite prin microfon, in care fiecare nod reprezinta un agent de specificatii. Fiecare nod are ca atributii una sau mai multe dintre urmatoarele sarcini:

- Informare: trimisa catre utilizator (solicitare trimisa de catre Kephera III catre o persoana care doreste sa interactioneze vocal cu robotul).
- Cerere: cerere de informatii din partea utilizatorului (de exemplu care sunt comenzile posibile din partea robotului, restragerea acestui domeniu pentru anumite situatii particulare, etc.).
- Asteptare: Informatie asteptata fara o anumita cerere .
- Executie: Efectuarea unor actiuni precum deplasarea pe o anumita directie sau rotirea cu un anumit unghi.

Nodurile neterminale grupeaza alti agenti si controleaza executia lor captand cel mai inalt nivel temporal si structura logica a dialogului utilizator – robot.

Fiecare comanda catre robot incepe cu un scurt mesaj din partea robotului pentru ca utilizatorul sa poate incepe comanda. Sistemul asteapta cuvintele individului uman si preia cuvintele cheie (concepte) precum un cuvânt predefinit in baza de date sau cuvânt de inceput/sfarsit a comandai.

Daca modulul detine suficienta informatie pentru a executa comanda (i.e. daca identificarea elementelor cheie s-a realizat cu succes si, mai mult decat atat, acestea sunt suficiente pentru a identifica o actiune predefinita a robotului) atunci robotul reactioneaza in consecinta (prin miscare) iar modulul de dialog va solicita o noua cerere de la momentul executiei celei precedente. Daca informatia obtinuta nu este suficienta pentru indeplinirea unei actiuni (de exemplu utilizatorul a precizat destinatia robotului inasa nu a precizat viteza desi acest lucru era prestabilit ca parametru, sau s-a precizat unghiul cu care robotul sa se roteasca inasa nu si durata si viteza miscarii) atunci modulul de dialog dintre Kephera III si utilizator va semnala acest lucru⁵².

Unii dintre agenti (noduri) pot indeplini sarcinile date ale conceptelor recunoscute pe baza informatiilor partiale. Complexitatea arborelui este direct proportionala cu timpul de raspuns al programului deoarece presupune cautarea in baza de date a informatiilor gasite prin control vocal in timp real si efectuarea actiunilor care descind. Sistemul poate repeta sub o forma standard, in care sunt introduse informatiile cheie obtinute, cunostintele dobandite de la utilizator pentru indeplinirea acestor actiuni.

3.2.5 Generarea limbajului

Pentru generarea limbajului vom folosi un modul special, capabil sa genereze cuvinte rostite pe baza unor sabloane si umpland golurile de informatie ramase libere atunci cand este realizata o anumita comanda prin intermediul modulului de dialog Kephera III – om. Totodata acest modul este capabil sa realizeze o selectie dintr-o lista de sabloane pentru un raspuns/interogare primit(a). Cuvintele rostite sunt apoi trimise catre sintetizare⁵³.

⁵² Young S., „*The HTK hidden Markov model toolkit: Design and philosophy*” Cambridge University Engineering Department, UK, Tech. Rep. CUED/FINFENG/TR152, Sept. 2004.

⁵³ Procesarea limbajului natural: Huang X., Acero A., Hon H., „*Spoken Language processing*”, Prentice Hall PTR, New Jersey, 2005.

Modulul de generare a limbajului identifica diferite tipuri de acțiuni care au nevoie ca anumite cuvinte să fie rostite; aceste acțiuni au propriile lor sub-module care înseamnă mapări între diferitele concepte și sabloanele care le generează. Sistemul este compus din 3 module diferite pentru generarea limbajului: unul pentru afișarea informației către utilizator, unul pentru a cere informația de la acesta, și ultimul care confirmă informația primită de la utilizator. În interiorul acestor module există o multitudine de sabloane care generează cuvintele rostite. De exemplu un modul: sablonul pentru „cerere.loc_destinație” care generează cuvintele cheie rostite: „Care este destinația?” solicită utilizatorului conceptul corespunzător pentru a putea acționa roțile motoare ale robotului și începerea sarcinii. Modulul de informare are un anumit „timp_curent” care generează în mod automat secvența: „Acum este timpul: hh:mm:ss.”. Utilitatea acestui modul rezidă în faptul că robotul poate intra în starea de „stand-by” pentru a aștepta o nouă comandă după 5 minute, de exemplu⁵⁴.

3.2.6 Sintetizator de voce

Sinteza se referă la calitatea semnalului audio. Unul dintre scopurile proiectului este acela de a investiga cea mai bună „iesire” pentru calitatea vocii. Astfel, se dorește controlul asupra ieșirii date de comanda vocală. Acest fapt poate fi cel mai ușor îndeplinit prin intermediul unui sintetizator.

Rolul acestuia este de a oferi utilizatorilor care intră în interacțiune cu robotul un anumit tipar de voce ușor de înțeles pentru om. Sintetizatorul poate conține o gramatică la fel de extinsă, un vocabular similar cu cel folosit pentru identificare.

Rezultatul este concretizat prin intermediul unei voci de înaltă calitate care funcționează pe baza unor propoziții generate de către sistem⁵⁵.

3.3 Dialogul dintre utilizatorul uman și Kephera III

Deși simplu de conceput modul de interacțiune dintre cele două părți, modul de implementare împreună executia comenzii prezintă unele puncte de îmbunătățit și pași dificil de realizat.

Numirea unei comenzi printr-un cuvânt cheie pare un fenomen simplu însă, la o analiză mai atentă a semnalului purtător, sistemul capătă o complexitate sporită. Utilizatorii pot folosi un domeniu variat de formulări pentru o anumită comandă iar robotul trebuie să se adapteze la acestea și, mai mult decât atât, să realizeze comanda. Deși pot exista unele constrângeri și prescurtări în limbajul vocal, robotul trebuie să le recunoască (de exemplu în cazul unei comenzi „înainte” pentru 5 metri, utilizatorul poate folosi „m” ca prescurtare etc.). Kephera III poate solicita pentru toate informațiile de care are nevoie pentru a-și îndeplini cu succes traseul.

Să considerăm, ca exemplu, cazul în care utilizatorul dorește ca robotul să se deplaseze primii 5 metri de la poziția inițială în față, robotul să se rotească apoi cu un unghi de 90 de grade spre dreapta, iar apoi să se deplaseze înapoi cu alți 3 metri. Interacțiunea (dialogul) dintre părți va fi cel prezentat mai jos⁵⁶:

SISTEMUL: Sunt Kephera III. Care sunt comenzile dumneavoastră?

UTILIZATORUL: Comanda vocală înainte pentru 5 metri.

⁵⁴ Ravishankar M.K., „Efficient algorithms for speech recognition”, PhD Thesis (CMU Technical Report CS-96-143), Carnegie Mellon University, Pittsburgh, PA, 2006.

⁵⁵ Toate părțile constitutive se regăsesc și în modelul: Lee K.F., Hon H. W., și Reddy R., „An overview of the SISTEM speech recognition system”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 1, pp. 35–45, Ian. 2005.

⁵⁶ Orice alt limbaj poate fi general considerând: Raux, A., „Automated Lexical Adaptation and Speaker Clustering based on Pronunciation Habits for Non-Native Speech Recognition”, INTERSPEECH (ICSLP) 2008, Jeju Island, Korea.

SISTEMUL: OK. Am pornit. [...] Comanda indeplinita. Doriti o alta comanda?

UTILIZATORUL: Da. Comanda vocala rotire 90 de grade spre dreapta.

SISTEMUL: OK. Comanda indeplinita. Doriti o alta comanda?

UTILIZATORUL: Da. Comanda vocala inapoi pentru 3 metri.

SISTEMUL: OK. Comanda indeplinita. Doriti o alta comanda?

UTILIZATORUL: Nu. Multumesc.

SISTEMUL: Multumesc! Felicitari!

3.4 Evaluarea sistemului

3.4.1 Capabilitati

Sistemul detine:

- un mod in timp real re recunoastere a semnalelor discrete si continue.
- arhitectura a interfetei primare care include, printre altele, implementari ale distantei Hamming, transformatei Fourier discrete, filtru de zgomot, transformata cosinus discreta, module de normalizare, module de extragere a semnalului, etc.
- arhitectura a modelului de limbaj care contine suport pentru caractere ASCII si comanda binara, Java Speech API Grammar Format (JSGF) si gramaticii in format ARPA-FST.
- arhitectura generalizata a modelului acustic.
- modul de cautare si gestiune a cautarii in baza de date prin inteligenta artificiala de tip breadth-first (cautare in latime) si un altul pentru cautarea unui cuvinte cheie.
- module de utilitati pentru recunoastere post-procesare, incluzand scoruri si rezultate de identificare, latici generalizare etc.
- alte unelte precum cele de afisare a spectrogramelor si a semnalului sub forma de unda precum si generarea unor caracteristici ale fisierului audio.⁵⁷

3.4.2 Performante

Sistemul software este unul deosebit de flexibil capabil de indeplinirea cu succes a mai multor sarcini de recunoastere vocala. Un obectiv greu de reprezentat numeric este acela de a prezenta performantele sale prin cateva caracteristici sau „stari” ale performantei sale in concordanta cu viteza sau acuratetea. In schimb pot fi deduse anumite marimi caracteristice pentru a determina comportarea sistemului sub o varietatea de sarcini dintre cele mai diverse⁵⁸. Aceste sarcini impreuna cu rezultatele lor sunt prezentate in continuare:

- Cifre izolate (T1): presupune rularea programului cu date pre-inregistrate pentru a definitiva anumite metrice de performanta prin recunoasterea unui singur cuvint. Vocabularul este dat de cifrele de la 0 la 9 astfel incat fiecare dintre ele este pronuntata o singura data, separat.

⁵⁷ Performante bazate pe transformari Markov: Young S., „*The HTK hidden Markov model toolkit: Design and philosophy*” Cambridge University Engineering Department, UK, Tech. Rep. CUED/FINFENG/TR152, Sept. 2004.

⁵⁸ Rezultatele date pot fi imbunatatite cu ajutorul adaptarii lexicului la utilizator: Raux, A., „*Automated Lexical Adaptation and Speaker Clustering based on Pronunciation Habits for Non-Native Speech Recognition*”, INTERSPEECH (ICSLP) 2008, Jeju Island, Korea.

George POPESCU – Comandă vocală Kephhera III

- Cifre interconectare (Tcifre): Extinde testul pentru cifre izolate pentru recunoasterea a mai mult de un cuvânt pentru un anumit interval de timp. Vocabularul este format deasemenea din cifrele de la 0 la 9 astfel încât să existe o secvență de cifre pentru un anumit interval luat în considerare.
- Vocabular redus (Vredus): Extinde vocabularul la aproximativ 100 de cuvinte și conține date de intrare date de cuvinte pronunțate în întregime precum și cuvinte rostite litera cu litera.
- Vocabular mediu (V1K): Extinde dimensiunea vocabularului la aproximativ 1.000 de cuvinte.
- Vocabular mediu extins (V5K): Extinde dimensiunea vocabularului mediu la aproximativ 5.000 de cuvinte.
- Vocabular mediu s-extins (V20K): Dimensiunea vocabularului este de 20.000 de cuvinte.
- Vocabular extins (V64K) : Extensie a vocabularului pentru aproximativ 64.000 de cuvinte.

Următorul tabel compară performanțele pentru fiecare dintre vocabularele de mai sus.

Test	WER	RT(1)	RT (2)	Dimensiunea Vocabularului	Modelul limbajului
T1	0.168	.03	.02	11	isolated digits recognition
Tcifre	0.549	0.07	0.05	11	continuous digits
Vredus	1.192	0.25	0.20	79	trigram
V1K	2.88	0.50	0.41	1,000	trigram
V5K	6.97	1.22	0.96	5,000	trigram
V64K	18.756	~4.4	3.95	60,000	trigram

Nota: Performanțele asupra ultimelor 3 tipuri de vocabular nu sunt complete!

Cod:

- **WER** - Word error rate – Rata de eroare pentru un cuvânt (%) (cu cât este mai scăzută cu atât rezultatele sunt mai bune)
- **RT** - Real Time – Timp Real – Proportie al timpului de procesare față de timpul de înregistrare. - (cu cât este mai scăzută cu atât rezultatele sunt mai bune)
- **RT(1)** – Rezultate pentru o configurație cu un singur procesor
- **RT(2)** – Rezultate pentru o configurație cu 2 procesoare

Aceste date au fost obținute pentru o configurație Intel Core 2 Duo CPU T7500 @ 2.20 GHz, 2GB RAM⁵⁹.

3.4.3 Instalarea sistemului

3.4.3.1 Platforma software necesară

Sistemul a fost construit și testat pe fiecare dintre următoarele platforme: Solaris™, Mac OS X, Linux și WinXP. Rularea sistemului de aplicații, construcția și testarea sa au nevoie de pachete software adiționale: înainte de execuția sa vor fi necesare următoarele resurse software:

- **Java 2 SDK, Standard Edition 5.0** sau o versiune ulterioară. Pachetul poate fi downloadat de la java.sun.com, "J2SE". Una dintre versiunile recomandate este 6.0 (referită ca și 1.6.0).
- **Ant 1.6.0** sau o versiune ulterioară, disponibilă la ant.apache.org. Pe site se găsește și un manual cu instrucțiuni pentru downloadare, instalare și folosire pentru ant. Acest pachet este necesar doar pentru construirea sursei de distribuție vocală.

⁵⁹ Datele rezultate din articolul: Young S., „The HTK hidden Markov model toolkit: Design and philosophy” Cambridge University Engineering Department, UK, Tech. Rep. CUED/FINFENG/TR152, Sept. 2004.

- **Subversion (svn)**, doar daca se doreste interactiunea cu arborele svn (recomandata, de altfel). Locul de unde poate fi descarcat acest pachet este subversion.tigris.org. In cazul unui sistem de operare Windows, cea mai buna solutie este data de instalarea [cygwin](http://cygwin.com), care ofera un mediu asemanator celui din Linux intr-o fereastră similara celei de command prompt. Trebuie implicit aleasa obtiunea "svn" la instalare.

3.4.3.2 Constructia aplicatiei

Dupa cum pachetul sistem-{version}-bin.zip nu contine codul sursa, acesta trebuie download-at din pachetul sistem-{version}-src.zip. Cerintele software pentru constructia aplicatiei au fost definite in sectiunea de mai sus.

Setarea JSAPI 1.0

Inaintea de constructia aplicatiei propriu-zise este importanta setarea mediului Java pentru recunoastere vocala: Java Speech API (JSAPI), intrucat un numar de teste si demo-uri sunt bazate pe acest pachet⁶⁰.

Pentru constructia sistemului, de la fereastră de command prompt este necesara schimbarea directorului (folderului) unde a fost instalat sistemul (de obicei o comanda de tipul: "cd sistem" este suficienta).

Deasemenea este importanta setarea variabilelor de mediu JAVA_HOME, ANT_HOME si PATH precum a fost descris acest proces mai sus. Apoi se vor introduce urmatoarele comenzi:

```
ant
```

Aceasta comanda executa comanda [Apache Ant](http://ant.apache.org) pentru constructia claselor sistemului sub directorul bld, fisierele .jar sub directorul lob si demo-urile -> fisierele .jar sub directorul bin.

Pentru stergerea tuturor datelor de iesire in momentul constructiei lor se va introduce:

```
ant clean
```

Crearea fisierelor Java (Javadocs)

Fisierele Java au fost deja construite prin pachetul/arhiva sistem-{version}-bin.zip. Pentru constructia fisierelor Java in mod manual trebuie, in schimb, folosit pachetul sistem-{version}-src.zip. Comenzile necesare crearii fisierelor Java utilizatorul va accesa nivelul superior al directoarelor ("sistem-{version}"), si va scrie:

```
ant javadoc
```

Aceasta comanda va crea fisierele Java din clasele publice si va afisa doar metodele publice si campurile aferente. In general aceasta este intreaga informatie de care un utilizator are nevoie. Daca se doreste accesul la clasele private sau protected atunci pot fi generate fisierele Java corespunzatoare prin (de exemplu):

```
ant -Daccess=private javadoc
```

Setarea mediului IDE (Eclipse, Netbeans, Idea)⁶¹

Setarea este directa si presupune:

1. Adaugarea tuturor subfolderelor (/subdirectoarelor) (!) directorului sursa (src) ca directoare sursa pentru un nou proiect Java.

⁶⁰ Bohus B., Rudnicky A., „*RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda*”, Eurospeech07, 2007. interfatarea folosind JSAPI.

⁶¹ Alte realizari folosind IDE – Eclipse: Castillo O., Melin P., „*A New Approach for Voice Monitoring using Type-2 Fuzzy Logic and Fractal Theory*”, International Journal of General Systems, Taylor and Francis, Vol. 33, 2004, pp. 305-319.

2. Adaugarea lib/js.jar, lib/tags.jar si lib/jsapi.jar la classpath (calea claselor pentru proiect). Datorita unor restrictii de licentiere fisierul jsapi.jar nu este transferat direct catre sistem dar poate fi usor creat prin rularea lib/jsapi.sh (sau lib/jsapi.bat pentru Windows) o singura data.

Pentru efectuarea sarcinilor curente (precum detasarea (deployment) pachetului sistem.jar, a modelelor sau a fisierelor .jar pentru demo-uri) direct in mediul IDE (Integrated Development Environment) se poate adauga si fisierul build.xml ca fisier ant al proiectului. Acest lucru poate fi realizat in cele mai multe cazuri prin click dreapta pe fisierul build.xml in fereastra de navigare a mediului IDE si selectarea optiunii "Add as project ant file". Pentru depanarea (debug) aplicatiilor demo este nevoie de adaugarea directorului src/apps si fisierele modelului acustic .jar (care pot fi implicate/interconectate directorului lib cu ajutorul unei comenzi ant all) catre classpath.

Demo-urile

Sistemul contine un numar de aplicatii demo. Daca se foloseste distributia binara (sistem-{version}-bin.zip), atunci fisierele JAR ale demo-urilor sunt deja construite, astfel incat se vor rula direct. Totusi, daca a fost utilizata distributia sursa (sistem-{version}-src.zip sau via svn), este nevoie de constructia acestor demo-uri. Numele de mai jos prezinta diferite tipuri pentru ale aplicatiilor demo:

- **Demo Hello World:** o aplicatie pentru o linie de comanda care recunoaste expresii simple.
- **Demo Hello Digit** (Cifra): o aplicatie pentru o linie de comanda care recunoaste cifre interconectate.
- **Demo Hello N-Gram:** o aplicatie pentru o linie de comanda care foloseste un model de limba de tip N-gram pentru recunoasterea vocii.
- **Demo ZipCode:** o aplicatie Java care foloseste tehnologia Web Start pentru recunoasterea codurilor rostite pentru localizarea unor adresa, valori ale parametrilor de miscare pentru Kephera III.
- **Demo Fisier Wav:** un program demo simplu pentru demonstrarea decodificarii fisierelor audio (ex: .wav, .au, .mp3, etc.).
- **Demo Transcriber:** un program demo simplu pentru a descrie care implica un fisier audio continuu ce are mai multe cuvinte rostite separate de zone silentioase.
- **Demo JSGF:** un program demo simplu care arata cum un program poate interschimba intre mai multe gramatici JSGF.
- **Demo Dialog:** un program demo care arata cum un program poate interschimba intre multiple gramatici JSGF si gramatici de dictare.
- **Demo Action Tags:** un program demo care arata cum pot fi folosite tag-uri de actiune pentru post-procesare a obiectelor RuleParse (parsate prin reguli) si obtinute din gramatici JSGF.
- **Demo Confidence:** un program demo simplu care arata cum pot fi obtinute anumite scoruri pentru un anumit rezultat.
- **Demo Latice:** un program demo simplu care arata cum pot fi extrase latici din rezultatele de recunoastere vocala.

Exista deasemenea si un program de test in mod live – valabil doar in cazul sursei de distributie – disponibil prin fisierul sistem-src-{version}.zip dar care nu este disponibil in fisierul sistem-bin-{version}.zip. Uneltele audio repezinta instrumente vizuale care inregistreaza si afiseaza semnalul audio (unda) si spectrograma unui semnal audio de intrare. Este disponibil in ambele pachete: atat cel binar cat si in cel sursa.

Pana acum am realizat doar evaluari empirice ale sistemului. Unul dintre experimentele initiale a constat in incercarea ca mai multi utilizatori sa comunice cu Kephera III. Ideea urmarita a fost aceea de a se observa formularea cererilor care au loc pentru a se incerca, apoi, explicitarea cererilor. Am folosit un numar de 3 utilizatori umani pentru a sintetiza comenzile vocale si diferitele formulari posibile. Datele primite au fost utilizate pentru extinderea gramaticii initiale si setului de reguli pentru analiza si rafinarea modelului pentru dialog. Informatia totala perceputa a fost mai departe preluata pentru constructia limbajului de intrare in sistem.

Urmatoarele observatii pot fi deduse cu usurinta: sistemul functioneaza corespunzator pentru cererile simple. Miscarea robotului urmeaza analiza unui cuvant simplu, suficient pentru recunoastere si actiune. Atunci cand anumite informatii lipsesc, sistemul are posibilitatea de a solicita explicit utilizatorului aceste informatii. Dialogul poate fi extrem de simplu atunci o persoana doreste un test al miscarii de genul: „Miscare inainte.”. Explicitarea sistematica si repetata a confirmarii solicitarii primite poate fi nu tocmai placuta pentru om, insa

dat fiind numărul de erori de recunoaștere a vocii, este important pentru utilizator să fie monitorizat și înțeles de către sistem⁶².

Este general recunoscut faptul că recunoașterea vocală este un proces aproximativ, departe de a fi perfect. El poate fi îmbunătățit prin diferite tehnici algoritmice și pe baza inteligenței artificiale, însă niciodată nu va putea fi 100% sigur. Multe dintre aceste limitări provin din limitările modelului de limbaj „artificial”. Pe măsură ce sistemul înregistrează mai multe comenzi vocale din partea unui sau a mai multor utilizatori, eroarea va scădea iar gradul de precizie va crește. Suficiența datelor înregistrate este dependentă strict de succesul modelului.

Pentru a face sistemul accesibil către o gamă mai largă de utilizatori (în funcție de limba folosită, de tipul vocii, etc.) este nevoie de extinderea capacităților sale în ceea ce privește modulele cheie precum baza de date, cunoștințele dobândite, gradul de precizie, ajustarea permanentă a ponderilor folosite, etc. O altă îmbunătățire a sistemului are în vedere vocea folosită ca răspuns de către sistem, respectiv creșterea calității acesteia, abilitatea de a selecta un anumit tip, vârsta, tonalitatea, gramatica folosită și altele.

3.5 Recunoaștere vocală și rețele neuronale. Logica Fuzzy și Algoritmi genetici

Recunoașterea vocală, care poate împărțită în identificare și verificare, este procesul de recunoaștere automată a persoanei care vorbește pe baza trăsăturilor caracteristice⁶³, a informației cuprinse în undele sonore ale vocii. Această tehnică permite utilizarea vocii utilizatorului pentru verificarea identității sale și controlul unor servicii de acces precum apel vocal, telefon bancar, cumpăratori prin intermediul telefonului, servicii de acces la baze de date personale, servicii informaționale, poșta electronică vocală, controlul securității pentru arii informaționale confidențiale precum și accesul la distanță al computerului personal.⁶⁴

Identificarea vorbitorului este procesul prin care se determină care utilizator înregistrat a emis o anumită informație sonoră⁶⁵. Verificarea vocală, pe de altă parte, este procesul de acceptare sau respingere a identității proclamate a unui vorbitor. Cele mai multe aplicații în care o voce este folosită ca și cheie pentru confirmarea identității unui utilizator sunt clasificate ca și verificarea vorbitorului.⁶⁶

Metodele de recunoaștere pot fi la rândul lor împărțite în cele dependente de test și cele independente de text. Prima cere ca utilizatorul să spună anumite cuvinte cheie sau expresii având același text pentru sesiuni de training (antrenament) diferite, în timp ce cea de-a doua nu se bazează pe un text specific pronunțat⁶⁷.

⁶² Diferiți algoritmi de recunoaștere vocală: Ravishankar M.K., „*Efficient algorithms for speech recognition*”, PhD Thesis (CMU Technical Report CS-96-143), Carnegie Mellon University, Pittsburgh, PA, 2006.

⁶³ Sisteme logice Fuzzy: Karnik N. N., Mendel J. M., „*O introducere în Sisteme Logice de tip Fuzzy 2*”, Technical Report, University of Southern California, 2007.

⁶⁴ Introducere, „Recunoaștere vocală cu rețele neuronale, Logica Fuzzy de tip 2 și algoritmi genetici”, Melin P., Urias J., Solano D., Lopez M., Castillo O., August 2006, pag. 1-2.

⁶⁵ Melin P., Urias J., Solano D., Lopez M., Castillo O., „*Recunoaștere vocală cu rețele neuronale, Logica Fuzzy de tip 2 și algoritmi genetici*”, August 2006, Engineering Letters.

⁶⁶ Matsui T., Furui S., „Similarity normalization method for speaker verification based on a posteriori probability”, Proceedings of the ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, 2004, p. 59-62.

⁶⁷ S. Furui, „Cepstral analysis technique for automatic speaker verification”, IEEE Transactions on Acoustics, Speech and Signal Processing, 29(2), 2007, pp. 254-272.

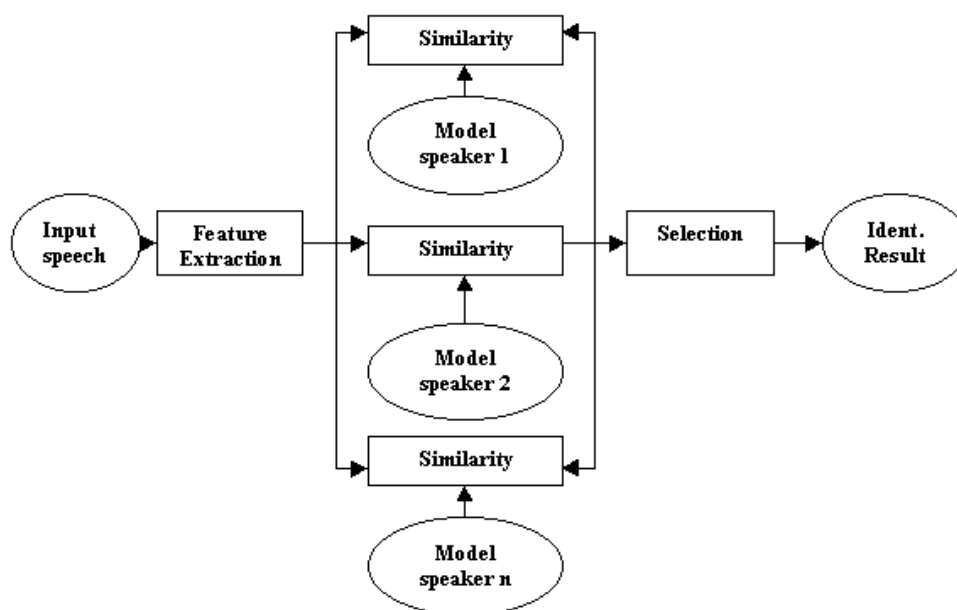


Figura 3.4: Identificarea utilizatorului

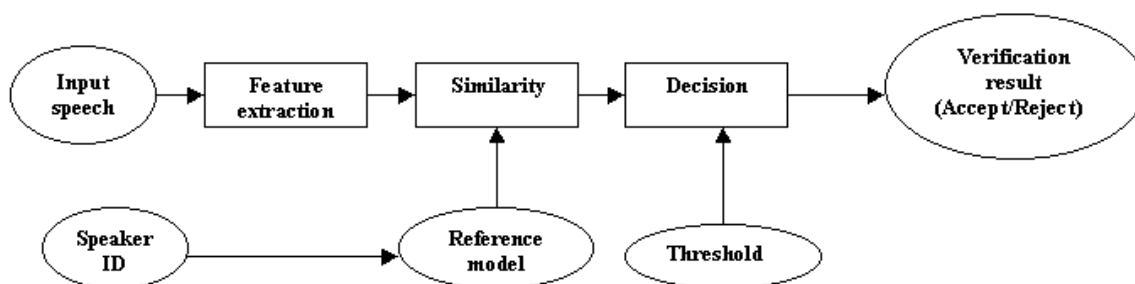


Figura 3.5: Verificarea utilizatorului⁶⁸

Rezultatele unei rețele neuronale de tip FEEDFORWARD pentru 20 de cuvinte în limba Engleză	Timp (min)	Num. De Cuvinte	Nr. Neuroni	Cuvinte Recunoscute	% Recunoastere
1a.	11	20	50	17	85%
2a.	04	20	50	19	95%
1a.	04	20	70	16	80%
2a.	04	20	70	16	80%
3a.	02	20	25	20	100%
1a.	04	20	25	18	90%
1a.	03	20	50	18	90%
2a.	04	20	70	20	100%
2a.	04	20	50	18	90%
1a.	07	20	100	19	95%

⁶⁸ Date considerate: Melin P., Gonzalez F., Martinez G., „Pattern Recognition Using Modular Neural Networks and Genetic Algorithms”, Proceedings of IC-AI'04, Las Vegas, USA, 2004, pp. 77-83.

George POPESCU – Comandă vocală Kephhera III

2a.	06	20	100	20	100%
1a.	09	20	50	10	50%
1a.	07	20	75	19	95%
1a.	07	20	50	19	95%
2a.	06	20	50	20	100%
1a.	29	20	50	16	80%
1a.	43	20	100	17	85%
2a.	10	20	40	16	80%
3a.	10	20	80	16	80%
1a.	45	20	50	11	55%
2 ^a	30	20	50	15	75%
3 ^a	35	20	70	16	80%

Tabel 3.6: Verificarea utilizatorului

Acum vom considera exemplul unui cuvânt pentru a ilustra o abordare hibridă⁶⁹. Folosim 2 module cu câte o rețea neuronală fiecare incluse în această arhitectură modulară⁷⁰. Fiecare modul este atribuit unei sesiuni de training pentru același set de date dar rezultatele sunt într-o anumită măsură diferite datorită incertitudinilor implicate de procesul de învățare. Diferența în rezultate este apoi folosită pentru a crea un interval fuzzy de tip 2, multime care reprezintă incertitudinea în clasificarea cuvântului dat. Mai jos am considerat cuvântul „exemplu” pronunțat în limba Engleză: „example”

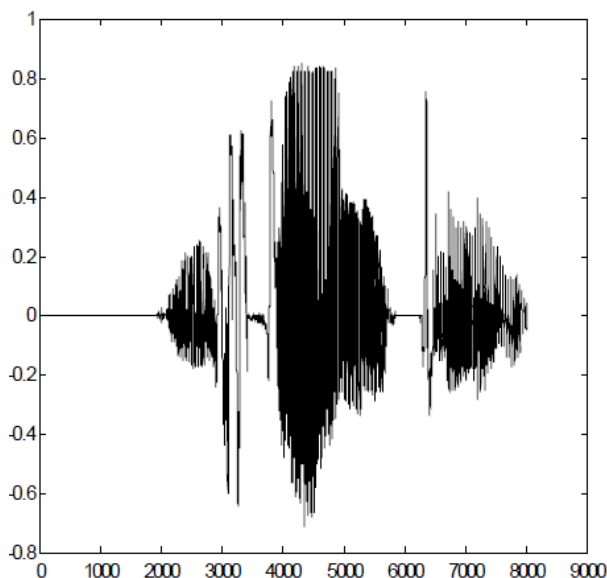


Figura 3.7: Cuvântul „exemplu”

Considerând acum 10 cuvinte în sesiunea de training avem ca prima rețea neuronală ne va oferi următoarele rezultate:

SSE = 4.17649e-005 (Sum of squared errors = Suma erorilor patrate)

⁶⁹ Castillo O., Melin P., „A New Approach for Voice Monitoring using Type-2 Fuzzy Logic and Fractal Theory”, International Journal of General Systems, Taylor and Francis, Vol. 33, 2004, pp. 305-319.

⁷⁰ Melin P., Urias J., Solano D., Lopez M., Castillo O., „Recunoaștere vocală cu rețele neuronale, Logica Fuzzy de tip 2 și algoritmi genetici”, August 2006, Engineering Letters.

George POPESCU – Comandă vocală Kephera III

Output = [0.0023, 0.0001, 0.0000, 0.0020, 0.0113, 0.0053, 0.0065, 0.9901, 0.0007, 0.0001]

„Iesirea” poate fi interpretata ca permitandu-ne sa facem estimari asupra preciziei identificarii pentru fiecare dintre cele 10 cuvinte. In acest caz apreciem ca valoarea 0.9901 este identificata cu cea corecta (1). Datorita procesului aleator de pronuntie a cuvintelor pentru o a doua retea neuronală vom avea rezultate diferite:

SSE = 0.0124899

Output = [0.0002, 0.0041, 0.0037, 0.0013, 0.0091, 0.0009, 0.0004, 0.9821, 0.0007, 0.0007]⁷¹

Putem acum observa ca identificarea cea mai buna corespunde valorii de 0.9821. Cu aceste 2 valori putem construi un interval de aproximare: [0.9821, 0.9901], care ne ofera incertitudinea in apartenenta cuvintului de corespondenta original la contextul dorit: „exemplu”, din baza de date⁷². Daca vom repeta aceasta procedura pentru intreaga baza de date de cuvinte vom obtine rezultate precum cele din tabelul urmator:

Exemplu	
M1	M2
0.0023	0.0002
0.0001	0.0041
0.0000	0.0037
0.0020	0.0013
0.0113	0.0091
0.0053	0.0009
0.0065	0.0004
0.9901	0.9821
0.0007	0.0007
0.0001	0.0007

Tabelul 3.8: Intervale de precizie

Experimentarea cu algoritmi genetici a problematii recunoasterii vocale pentru optimizarea numarului de „layere” (straturi) si noduri ale rețelei neuronale pentru fiecare modul in parte conduce la rezultate experimentale foarte bune⁷³. Abordarea folosita este similara cu cea descrisa pana acum. In figura de mai jos este un exemplu in care se foloseste un algoritm genetic pentru optimizarea acestui numar de straturi pentru una dintre rețelele neuronale ale arhitecturii modulare⁷⁴.

In aceasta figura putem aprecia minimizarea functiei de „potrivire” care ia in calcul doua obiective: suma patratelor erorilor si complexitatea rețelei neuronale⁷⁵.

⁷¹ Valorile sunt preluate din: „Recunoastere vocala cu rețele neuronale, Logica Fuzzy de tip 2 si algoritmi genetici”, Melin P., Urias J., Solano D., Lopez M., Castillo O., August 2006, pag. 7-8

⁷² Higgins A.L., Bahler L., Porter J., „Speaker verification using randomized phrase prompting”, Digital Signal Processing, Vol I, 2005, pag. 89-112.

⁷³ Melin P., Gonzalez F., Martinez G., „Pattern Recognition Using Modular Neural Networks and Genetic Algorithms”, Proceedings of IC-AI'04, Las Vegas, USA, 2004, pp. 77-83.

⁷⁴ Karnik N. N., Mendel J. M., „O introducere in Sisteme Logica de tip Fuzzy 2”, Technical Report, University of Southern California, 2007

⁷⁵ Capitolul de algoritmi genetici, „Recunoastere vocala cu rețele neuronale, Logica Fuzzy de tip 2 si algoritmi genetici”, Melin P., Urias J., Solano D., Lopez M., Castillo O., August 2006, pag. 8.

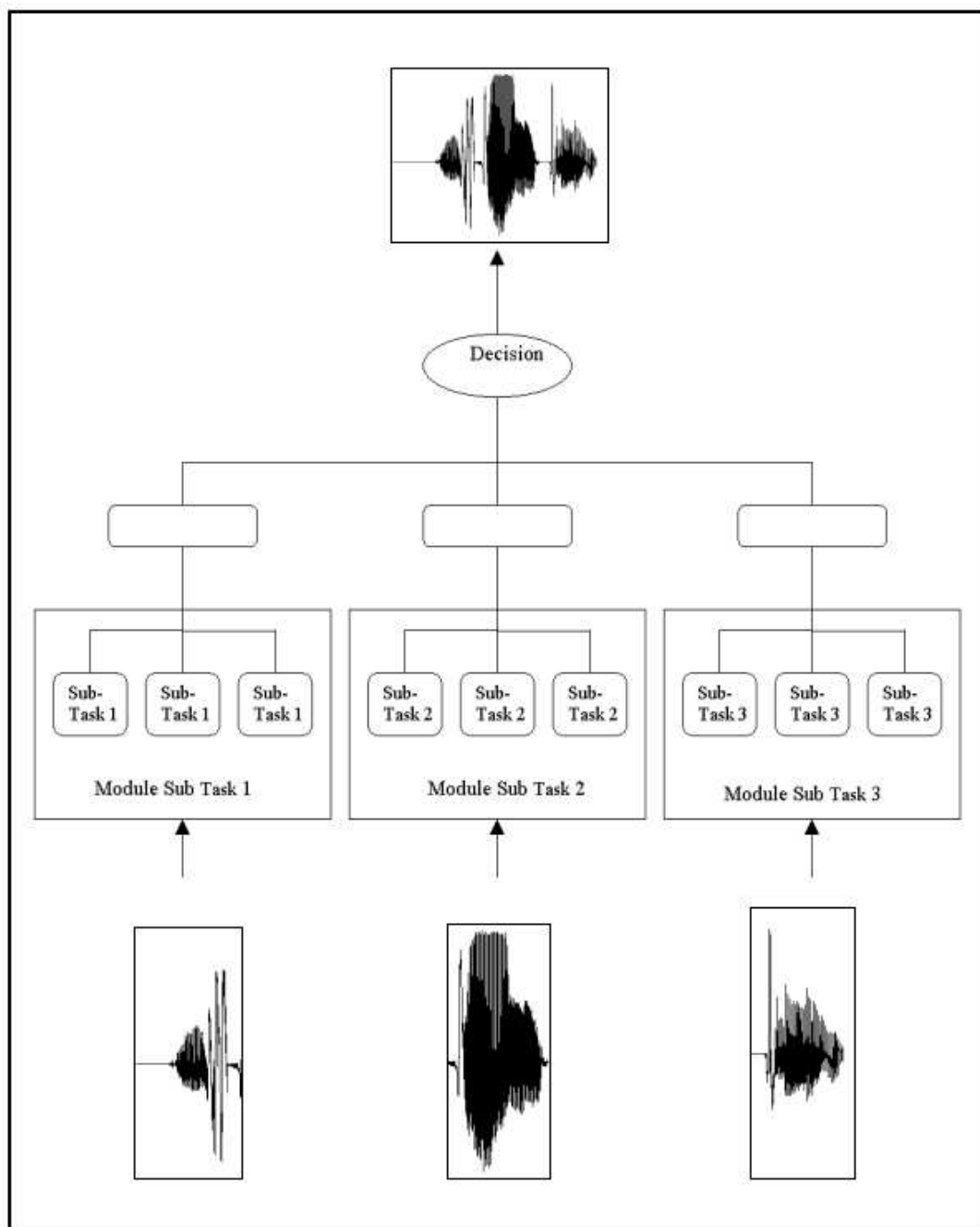


Figura 3.9: Arhitectura completa de retea neuronală pentru recunoaștere vocală

3.6 Definierea unui profil de utilizator pentru comanda vocală

În continuare vom defini un profil de utilizator pentru realizarea comenzii vocale cu robotul. Pentru aceasta este necesară navigarea în "Control Panel", pictograma selectată în imaginea de mai jos: "Speech"⁷⁶.

⁷⁶ Cum se realizează interconectarea mai multor sisteme de comandă vocală: Young S., Russell N.H., și Russell J.H.S., „*Token passing: A simple conceptual model for connected speech recognition systems*”, Cambridge University Engineering Dept, UK, Tech. Rep. CUED/F-INFENG/TR38, 2007.

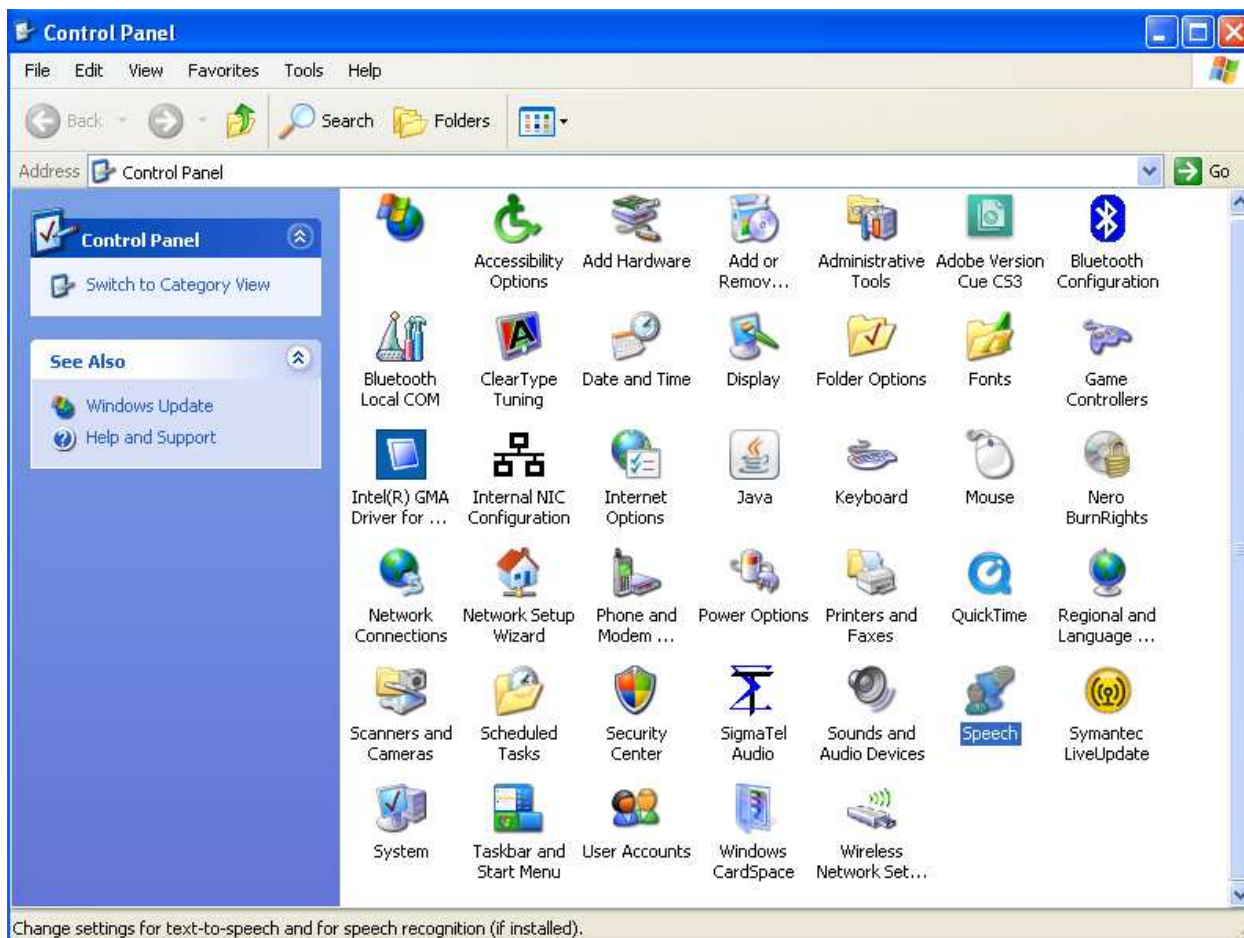


Figura 3.10: Control Panel – Speech

Utilizatorul va introduce un microfon in intrarile date de placa audio a computerului. Pentru aceasta va fi afisata fereastra de mai jos:

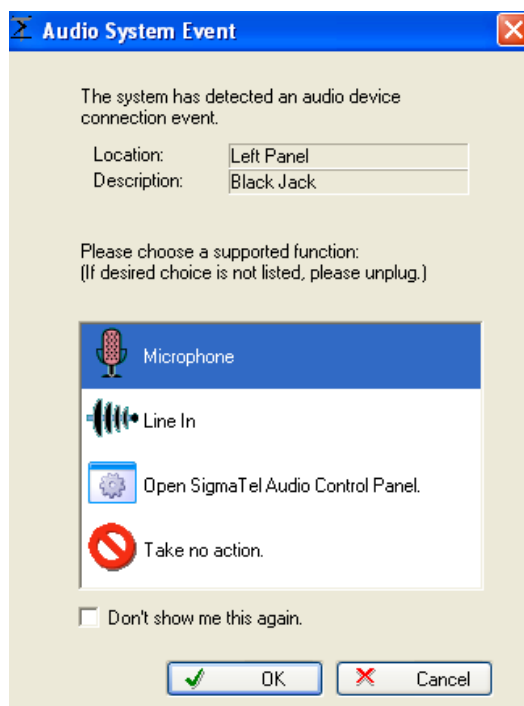


Figura 3.11: Conexiunea microfonului la computer

George POPESCU – Comandă vocală Kephera III

În fereastra ce deschide meniul pentru comanda vocală se poate observa un profil nou definit selectat ca și valoare “default”. Folosind acest profil robotul va putea fi controlat independent de alți utilizatori⁷⁷.



Figura 3.12: Profil de comandă vocală “George Popescu”

Acest profil a fost înregistrat pe baza amprentelor vocale determinate prin mai multe sesiuni de training prin citirea textelor prezentate în meniul de mai jos:

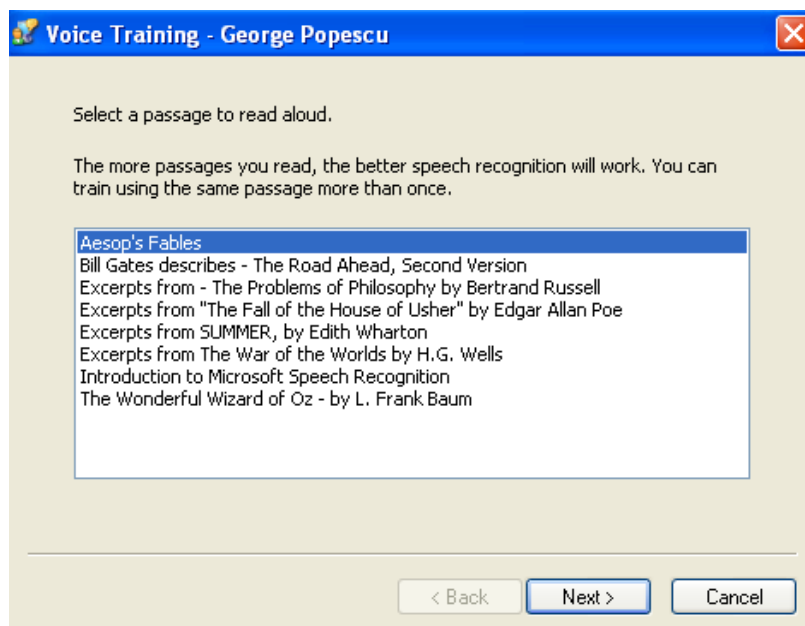


Figura 3.13: Training-ul profilului

⁷⁷ Alte aplicații de recunoaștere vocală au fost clasificate în: Singh R., Warmuth M., Raj B., și Lamere P., „Classification of speech applications”, în *Proceedings of the 8th European Conference on Speech Communication and Technology*, Geneva, Switzerland, Sept. 2005, pp. 1073–1776.

Observatii:

1. In urma definirii mai multor sesiuni de training computer-ul poate recunoaste vocea utilizatorului cu o acuratete din ce in ce mai mare. Prin urmare se recomanda citirea repetata a paragrafelor care definesc titlurile de mai sus.
2. Acuratetea pronuntiei cuvintelor dorite de catre utilizator va fi testata folosind o alta aplicatie: "Microsoft Speak2Text v2". Aceasta aplicatie va fi descrisa in sub-capitolul urmator.
3. Folosirea unui microfon de o calitate superioara va determina, implicit, valori mai precise si o exactitate sporita in recunoasterea cuvintelor pronuntate.
4. Pe de alta parte, definirea mai multor profiluri de utilizator poate fi realizata fara a se realiza suprapuneri ale sintezei semnalelor deoarece, inainte de recunoasterea propriu-zisa, va fi selectat un (singur!) anumit profil.

3.7 Editorul audio WavePad v. 4.03 – NCH Software

Pentru managementul fisierelor audio folosite la comanda vocala pentru Kephera III am ales un editor audio care foloseste o multitudine de functii de editare pentru identificarea semnalului dat de utilizator.

Programul software reprezinta versiunea 4.03 conceputa de NCH Software. Aplicatia este achizitionata sub licenta iar update-urile sunt disponibile permanent.

WavePad 4.03 permite salvarea fisierelor audio cu diferite extensii, printre care: Wave (*.wav), MPEG-Layer-3 (*.mp3), Windows Media Audio (*.wma), Ogg Vorbis (*.ogg), Apple AIFF (*.aif, *.aiff), MPEG-4 Audio (*.m4a), GSM (*.gsm), AMR Narrow Band Audio (*.arm), Speex (*.spx), etc⁷⁸.

Dupa instalare, meniul principal este cel afisat in figura urmatoare:

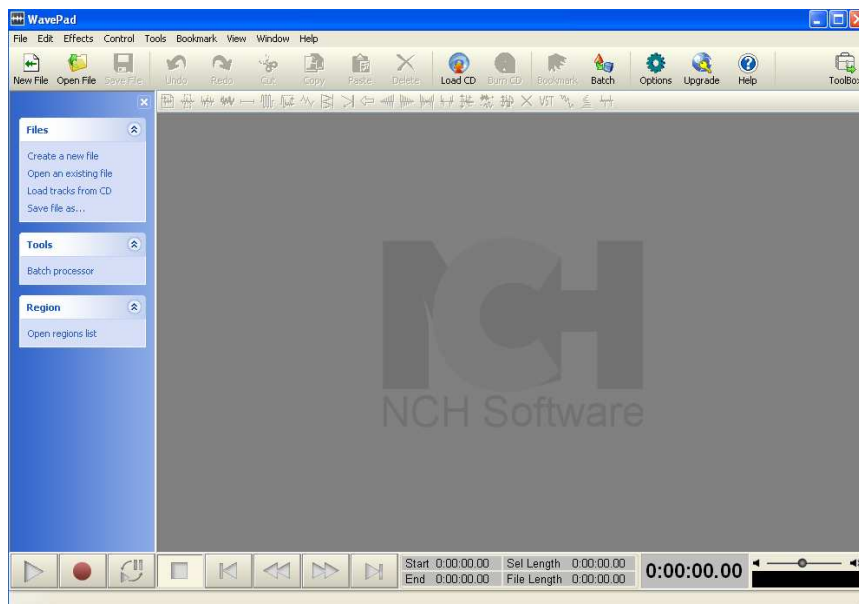


Figura 3.14: WavePad v. 4.03

In acest editor audio pot fi create fisiere de comanda vocala care vor fi necesare ulterior. Am preferat crearea unui fisier audio pentru inregistrare care sa fie de o calitate maxima oferita (44.100):

⁷⁸ In cartea urmatoare sunt prezentate metode avansate de verificare a vocii utilizator folosind diversi algoritmi: Higgins A.L., Bahler L., Porter J., „*Speaker verification using randomized phrase prompting*”, Digital Signal Processing, Vol I, 2005, pag. 89-112.

George POPESCU – Comandă vocală Kephera III



Figura 3.15: Fisier nou pentru inregistrare comanda vocala

Fereastra urmatoare va afisa un meniu pentru inregistrarea fisierului audio cu specificatii legate de timp, dispozitivul folosit (in acest caz Sigma Tel – placa audio a computer-ului), volumul inregistrarii:

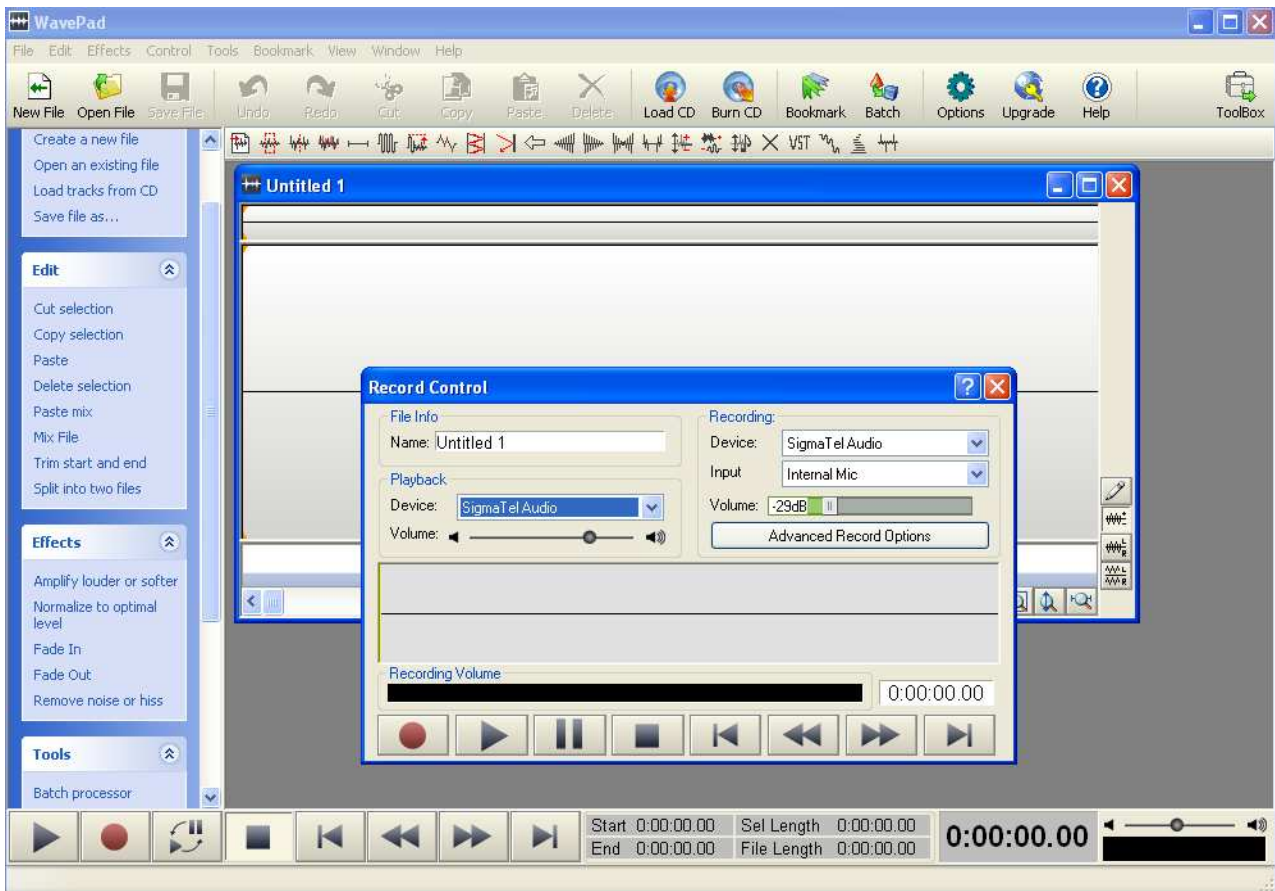


Figura 3.16: Inregistrare audio – optiuni

Comanda inregistrata este „Inainte”, sub forma unui semnal descris in figura pentru o perioada de 1, 45 secunde⁷⁹.

⁷⁹ Verificarea semnalului utilizator (voce): S. Furui, „*Cepstral analysis technique for automatic speaker verification*”, IEEE Transactions on Acoustics, Speech and Signal Processing, 29(2), 2007, pp. 254-272.

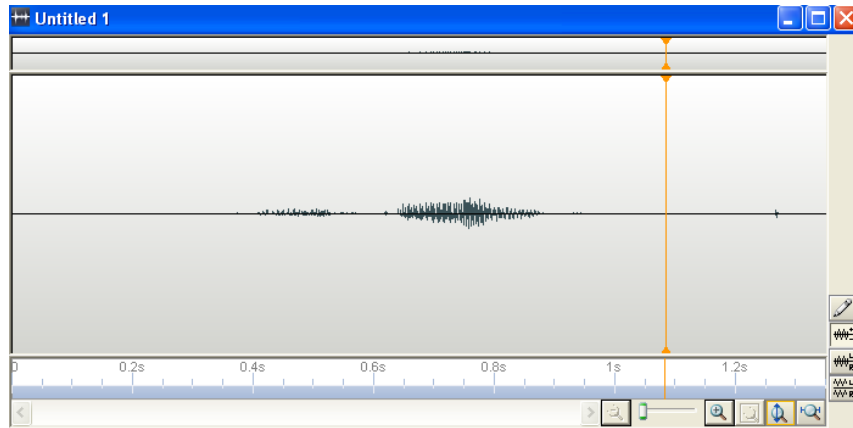


Figura 3.17: Semnalul „Inainte”

La salvarea fisierului in memoria computer-ului sunt permise modificari ale atributelor initiale (pentru a se micsora spatiul de stocare folosit) bazandu-se totodata pe un format PCM necompresat⁸⁰.

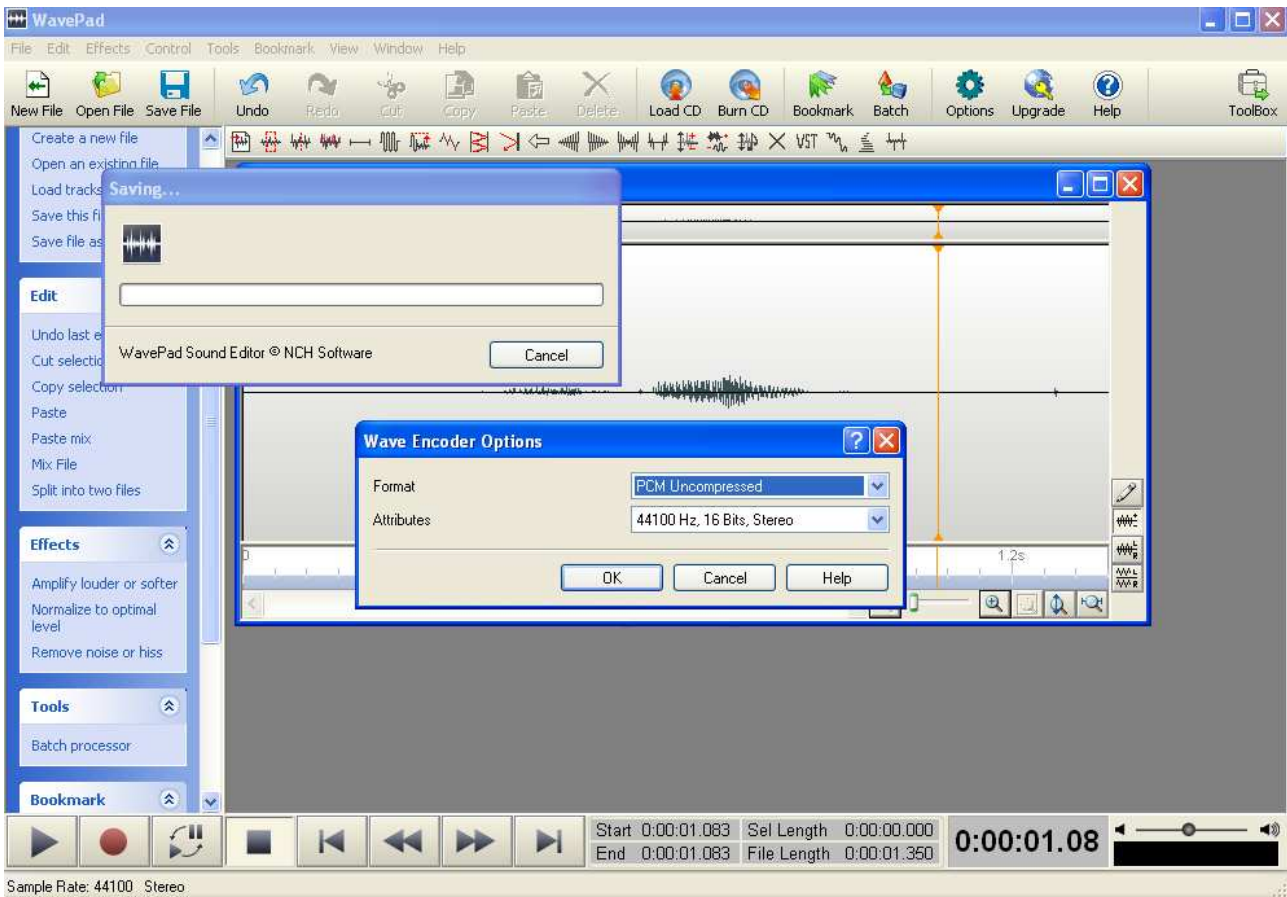


Figura 3.18: Salvare inregistrare „Inainte”

Semnalul vazut (mai) in detaliu are forma obtinuta in imaginea de mai jos:

⁸⁰ PCM = Pulse-Code Modulation. Este o reprezentare digitala a unui semnal analogic unde magnitudinea semnalului este esantionata la interval egale apoi cuantificata printr-o serie de simboluri ale unui cod numeric (de obicei binary).

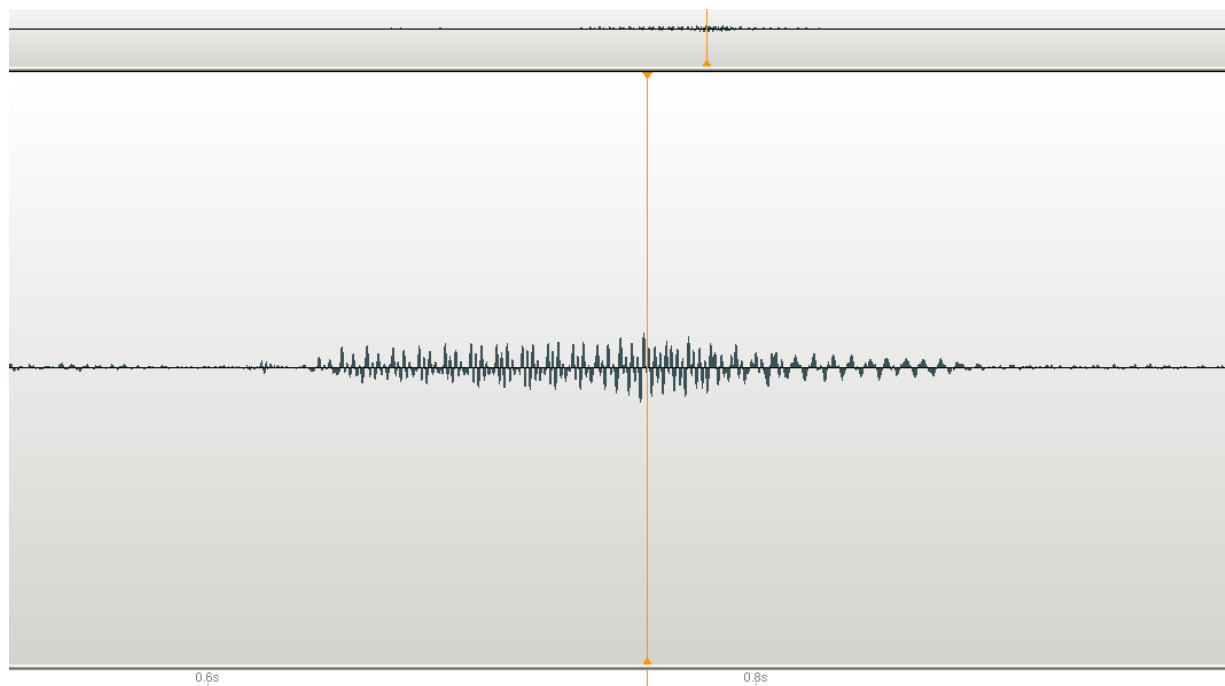


Figura 3.19: Semnalul „Inainte”: Zoom in

Daca se doreste o esantionare si mai fina a sa, atunci distributia amplitudinilor este data in figura urmatoare:

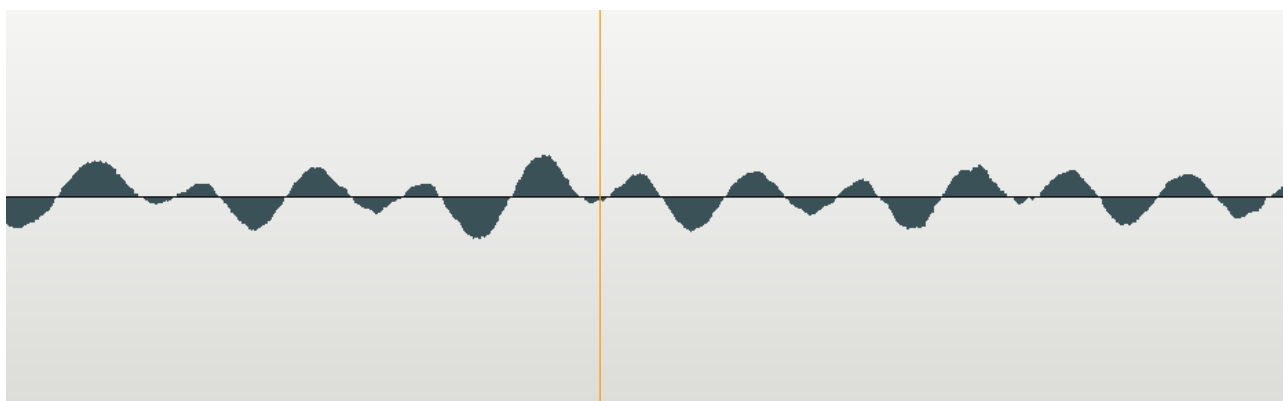


Figura 3.20: „Inainte”: Zoom in ++

Maximul permis de catre editor este reflectat in diagrama de mai jos (au fost mentinute coordonatele initiale – linia portocalie orizontala).

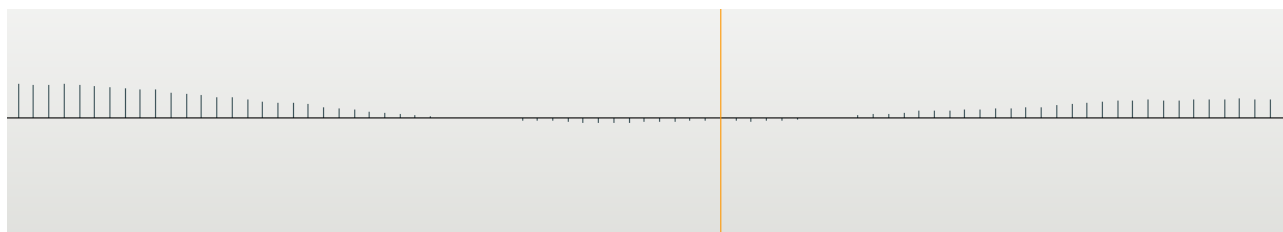


Figura 3.21: „Inainte”: Zoom maxim

La o distributie initiala (normala) semnalul „Dreapta” are coordonatele de mai jos. De data aceasta am folosit o alta inregistrare audio pentru a marca despartirea in silabe / pronuntia cuvântului. Se poate observa zgomotul de fond⁸¹.

⁸¹ Zgomotul de fond este zgomotul exte in general o perturbatie sonora (de mica intensitate) care apare in orice mediu de transmisie a semnalelor; zgomot exterior.

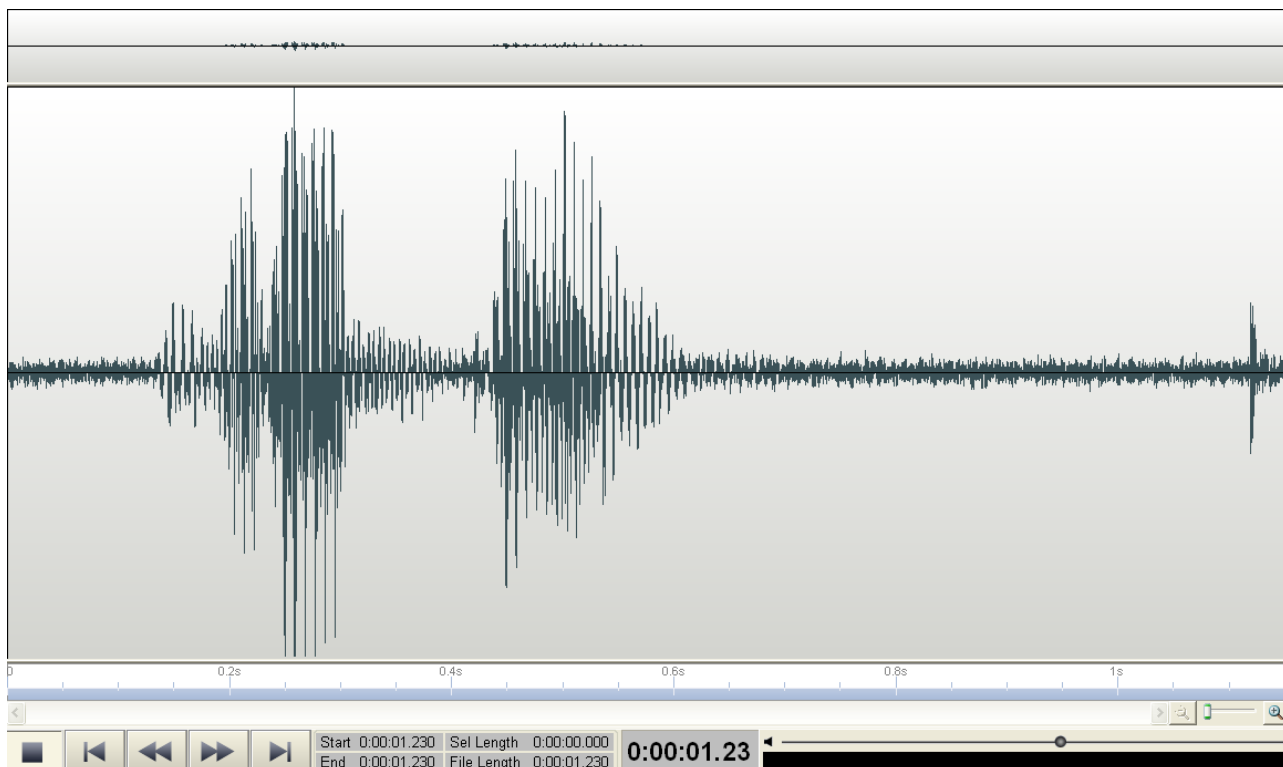


Figura 3.22: Comanda vocala „Dreapta”

Editorul WavePad detine multiple optiuni pentru editare audio si introducerea unor efecte speciale (precum reversul semnalului, descresteria amplitudinii, ecou, etc.) printre care si posibilitatea de reducere a zgomotului⁸².

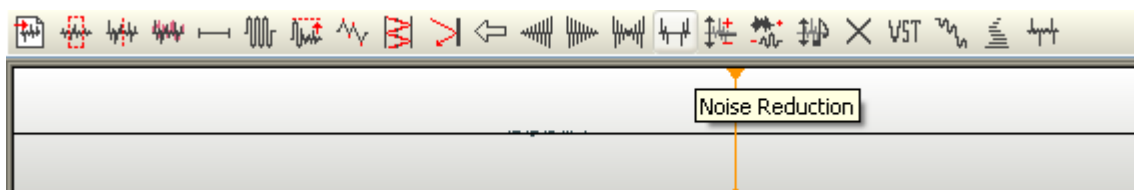


Figura 3.23: Reducerea zgomotului in WavePad

Aceasta reducere a zgomotului de fond poate fi ajustata in functie de nivelul zgomotului (dB) prin eliminarea zumzaitului si sasaitului.

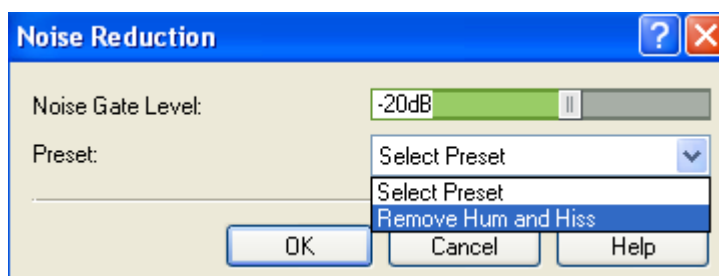


Figura 3.24: Eliminarea zgomotelor

Astfel, semnalul va avea „forma” din figura urmatoare:

⁸² Cum poate fi imbunatatita sinteza semnalului vocal: Langner, B., Black, A. „Improving the Understandability of Speech Synthesis by Modeling Speech In Noise”, ICASSP 2005, Philadelphia, USA, 2008



Figura 3.25: Semnalul initial „Dreapta” dupa eliminarea zgomotelor

Vom urmări în continuare domeniul de analiză a frecvenței. Desigur, valori mai mari pentru frecvență se vor înregistra la valori mai mari ale amplitudinii semnalului, acolo unde este înregistrată „practic” comanda vocală.

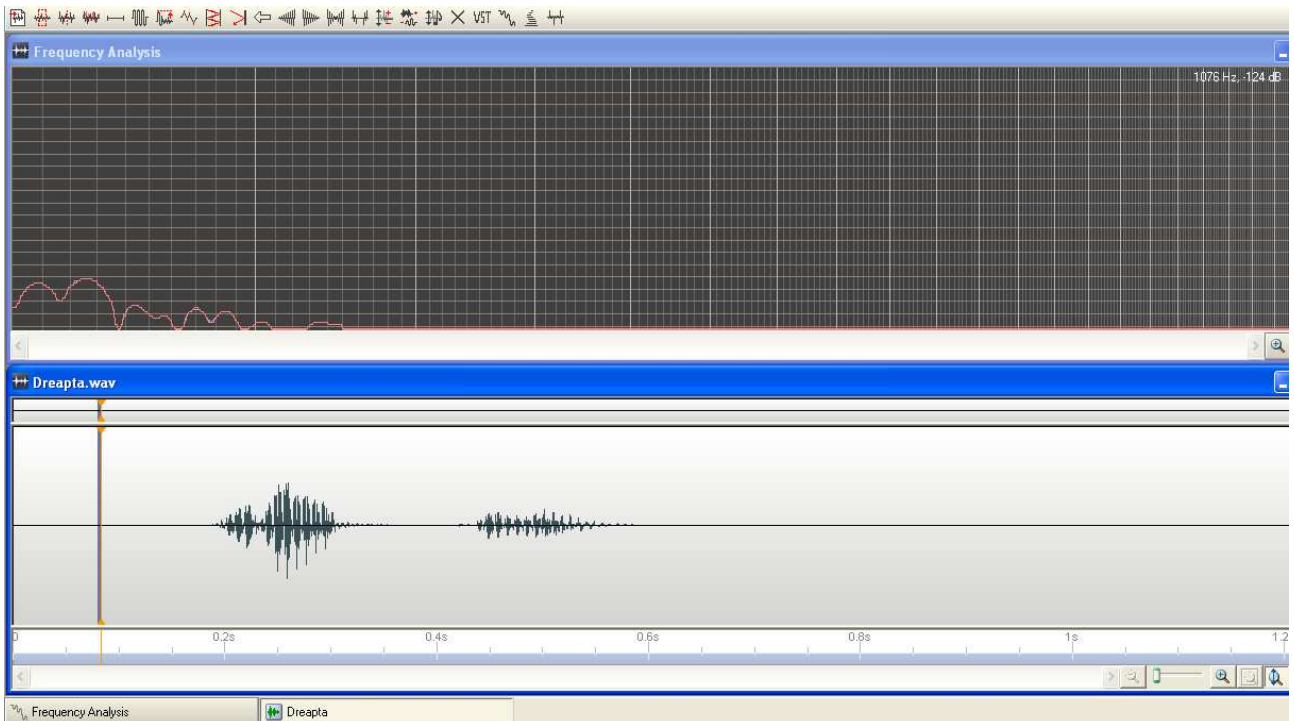


Figura 3.26: Domeniul de analiză a frecvenței pentru semnalul „Dreapta”

Parcurcând prima silabă („Dreap-”) se poate observa modificarea domeniului frecvenței, comparativ cu figura de mai sus.

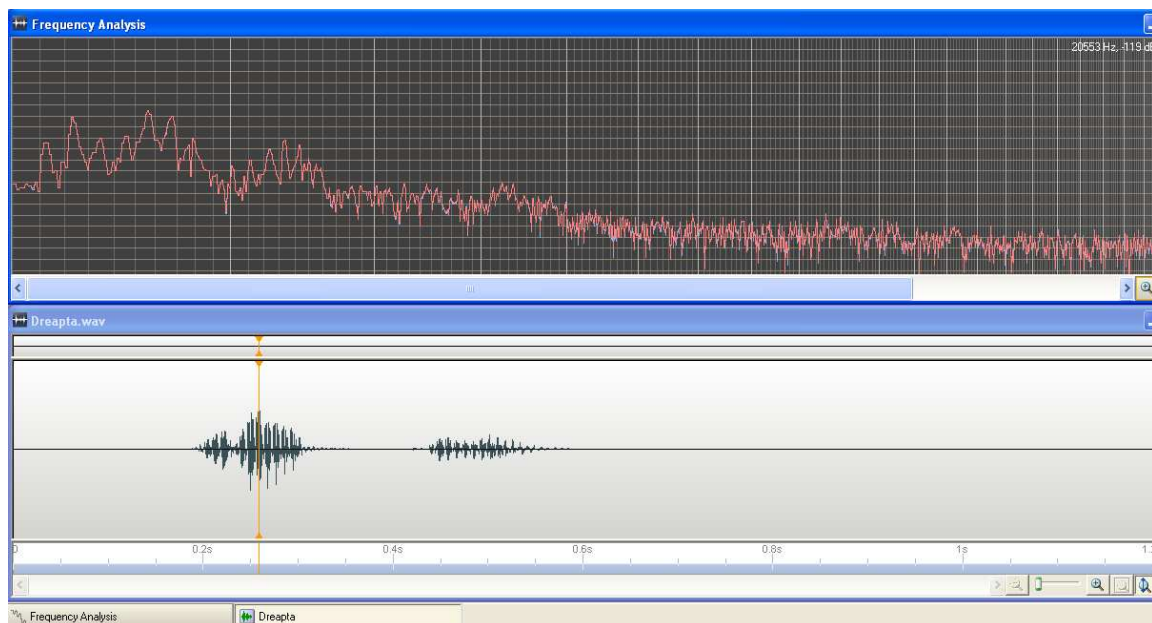


Figura 3.27: Domeniul frecvenței – analiza pentru prima silabă: „Dreap-”

În orice moment al înregistrării (finale) a sunetului pot fi introduse secțiuni de „liniste” (care nu contin nici un fel de zgomot: alb⁸³, roz⁸⁴, de fond etc.)⁸⁵.

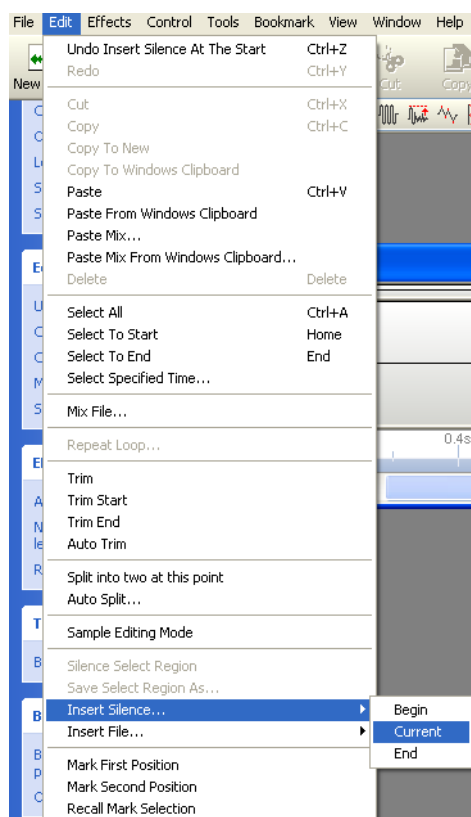


Figura 3.27: Inserare perioada de „liniste”

⁸³ Zgomot de banda largă folosit în măsuratori electro-acustice caracterizat prin energie egală pe latime de banda constantă.

⁸⁴ Zgomot de banda largă folosit în măsuratori electro-acustice caracterizat prin energie constantă pe octavă.

⁸⁵ O analiză mai detaliată se găsește în: Hrnčár M., „Aspects of voice-command creation in Human-Machine Interface”, Dissertation project, Žilina, 2007.

Editorul audio permite eliminarea si/sau reducerea zgomotelor prin mai multe metode. Fiecare dintre acestea vor fi analizate in continuare.

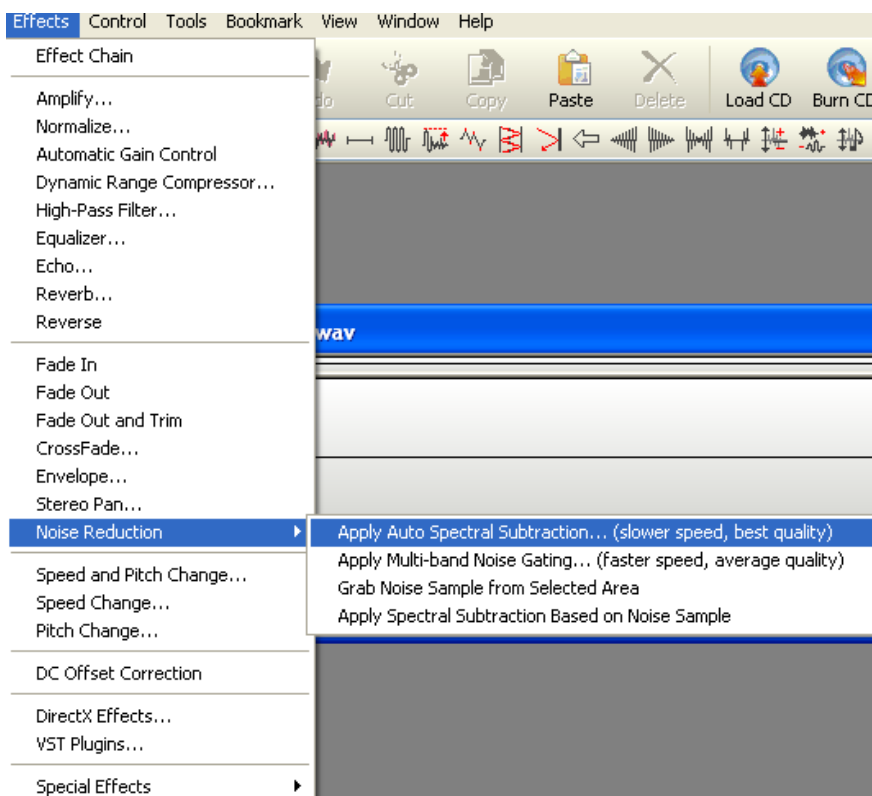


Figura 3.27: Eliminarea zgomotului prin metodele oferite de WavePad 4.03

Prima metoda de eliminare a zgomotului presupune substragere spectrala folosind un anumit procent al amplitudinii.

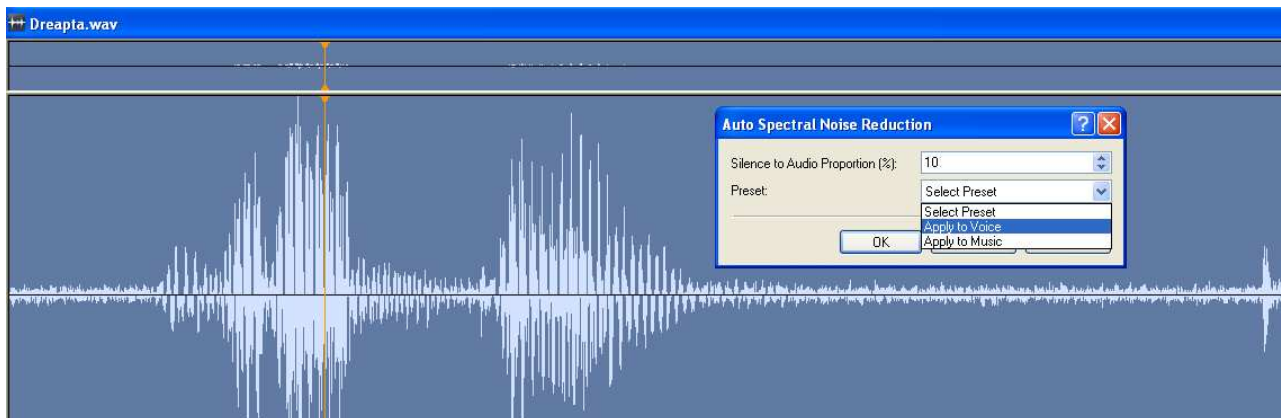


Figura 3.28: Eliminarea zgomotului – 1. Substragere spectrala

Semnalul obtinut este cel din figura de mai jos. Amplitudinea semnalului, dar si a zgomotului de fond au fost reduse cu 10 procente⁸⁶.

⁸⁶ O examinare similara si performante: Langner, B., Black, A., „An Examination of Speech In Noise and its Effect on Understandability for Natural and Synthetic Speech”, Carnegie Mellon University, Language Technologies Institute, Technical Report CMU-LTI-04-187, 2007.

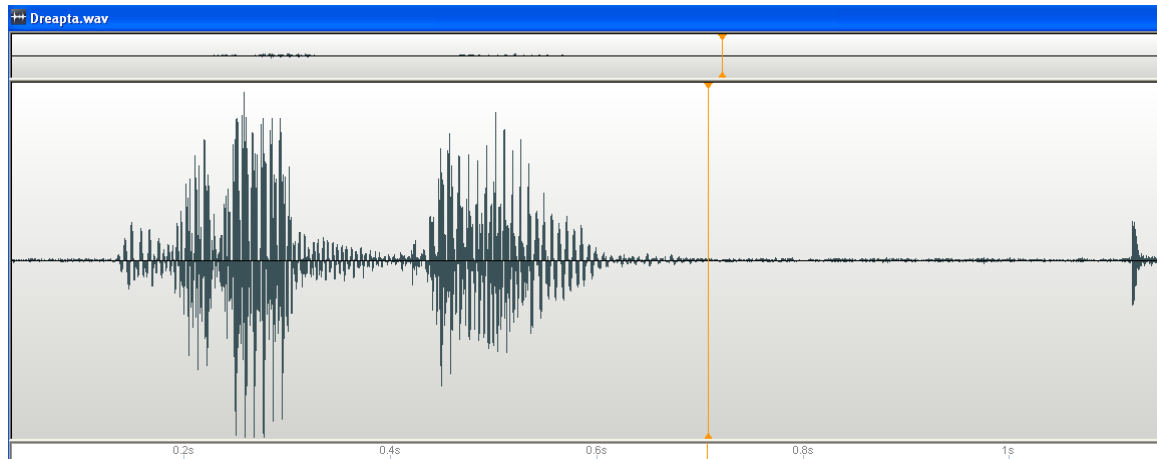


Figura 3.29: Eliminarea zgomotului – 1. Substragere spectrală – rezultat

Cea de-a doua metoda este aceea de reducere doar a zumzaitului si a sasaitului de fond cu anumita valoare:

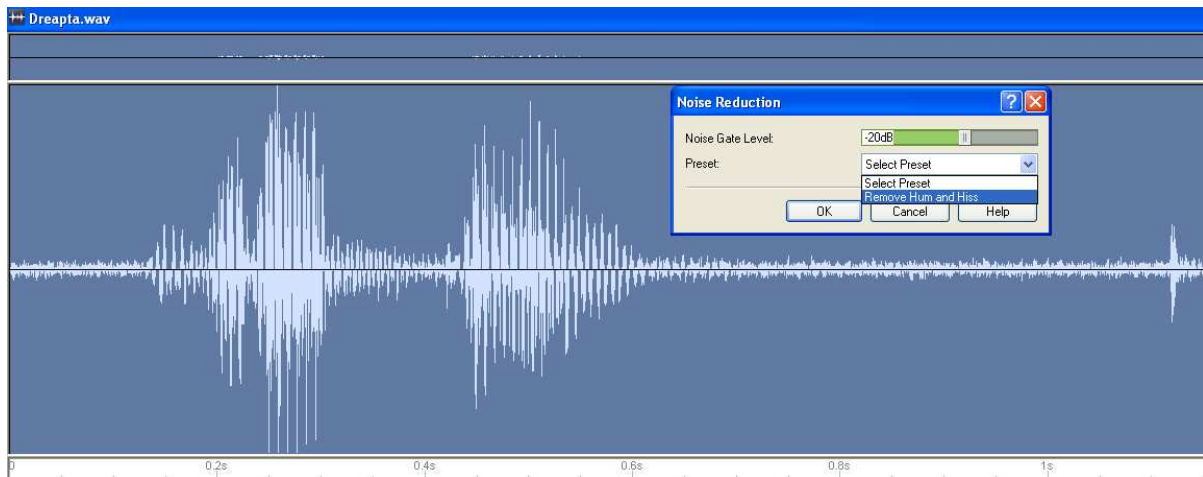


Figura 3.30: Eliminarea zgomotului – 2. Eliminarea zumzaitului si a sasaitului

Se observa de-a lungul liniei orizontale principale ca zgomotul de fond a fost redus in totalitate⁸⁷.

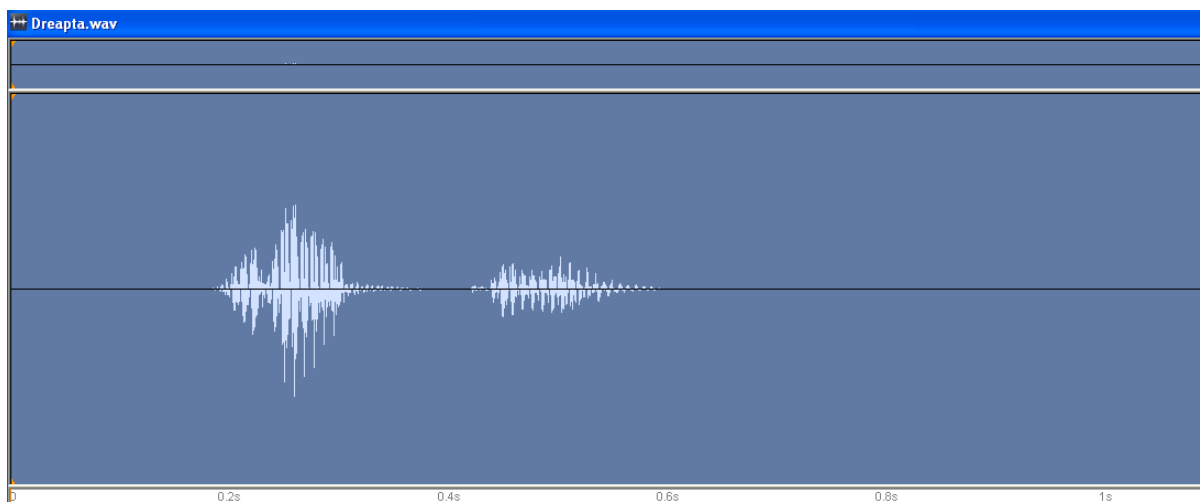


Figura 3.31: Eliminarea zgomotului – 2. Eliminarea zumzaitului si a sasaitului – rezultat

⁸⁷ Caracteristice precise pentru eliminarea zgomotului pot fi gasite in: Langner, B., Black, A. „*Creating a Database of Speech In Noise For Unit Selection Synthesis*”, 5th ISCA Speech Synthesis Workshop, Pittsburgh, USA, 2007.

Diferența dintre cele două metode poate fi observată selectând doar o anumită porțiune a graficului și aplicând „filtrarea” zgomotului de fond⁸⁸.

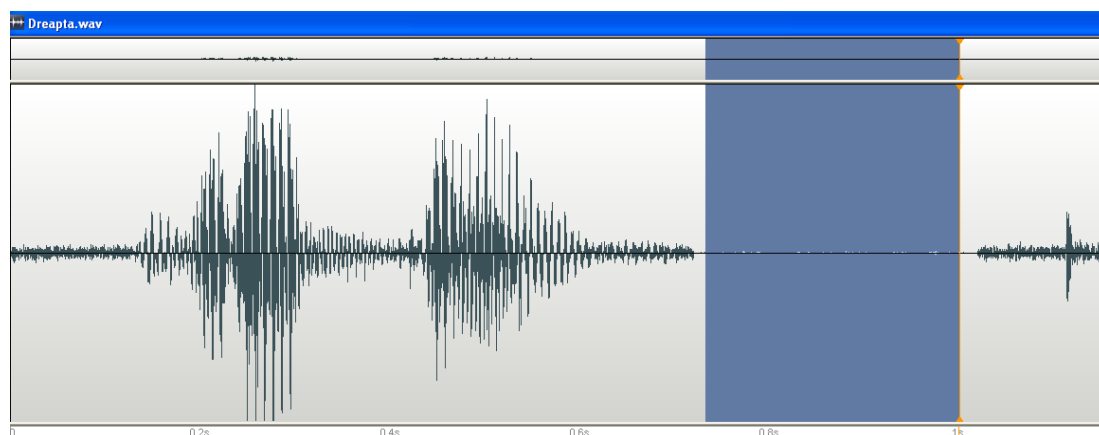


Figura 3.32: Eliminarea zgomotului – filtrare partitionata

WavePad poate recunoaște și elimina zgomotul alb, roșu⁸⁹, precum și orice alt semnal aflat sub o anumită valoare a amplitudinii.

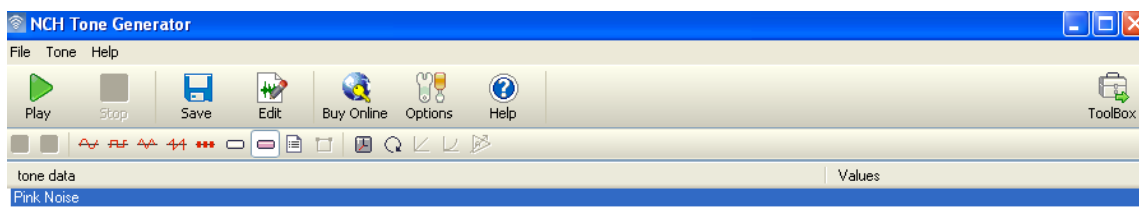


Figura 3.33: Zgomotul roz – recunoaștere

Semnalul dat de zgomotul roz este predefinit de către editor sub forma unui fișier tmp641.tmp. Caracteristica amplitudinii poate fi observată în figura din colțul din stânga-sus. Celelalte diagrame indică analiza în frecvență, atât pentru semnal, cât și pentru zgomotul roz, și semnalul inițial: „Dreapta.wav”.

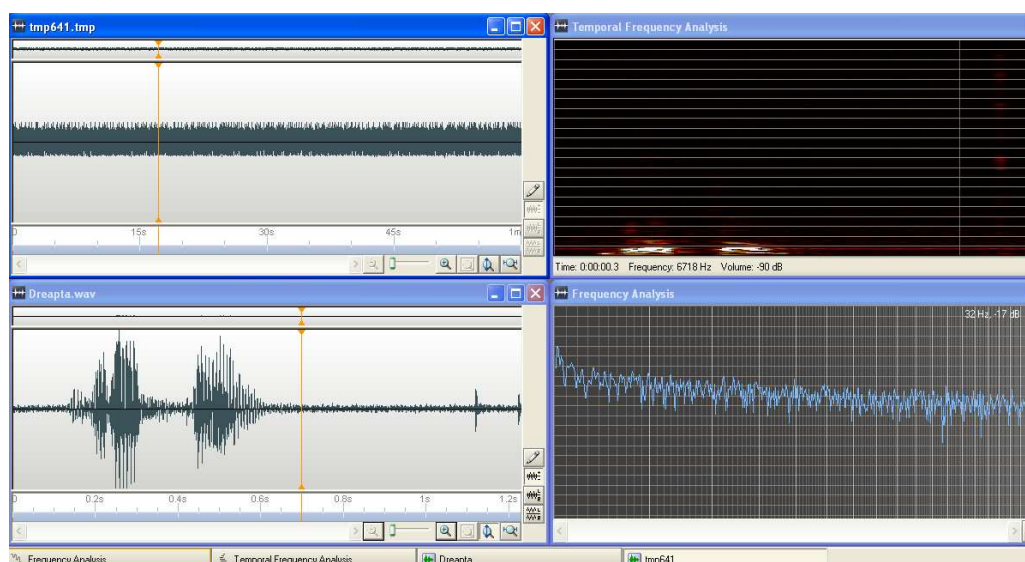


Figura 3.34: Zgomotul roz și semnalul inițial – Analiza în frecvență

⁸⁸ Pentru cazul persoanelor în vârstă și diferențierea sunetului produs de aceștia prin comandă vocală: Langner, B., Black, A. „Using Speech in Noise to Improve Understandability for Elderly Listeners”, ASRU 2005, San Juan, Puerto Rico, 2008.

⁸⁹ Zgomotul colorat este un zgomot de bandă largă folosit în măsurători electro-acustice caracterizat prin spectru continuu și uniform într-un domeniu de frecvență specificat.

Zgomotul alb este si el predefinit, aflandu-se in meniul de generare a tonului.

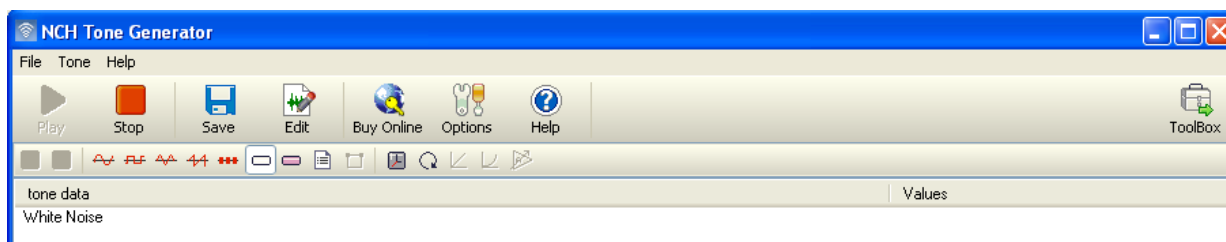


Figura 3.35: Zgomotul alb

Zgomotul alb aparent are aceeasi „structura” ca si cel roz. Pentru observatii mai fine este suficienta functia de Zoom a editorului⁹⁰.

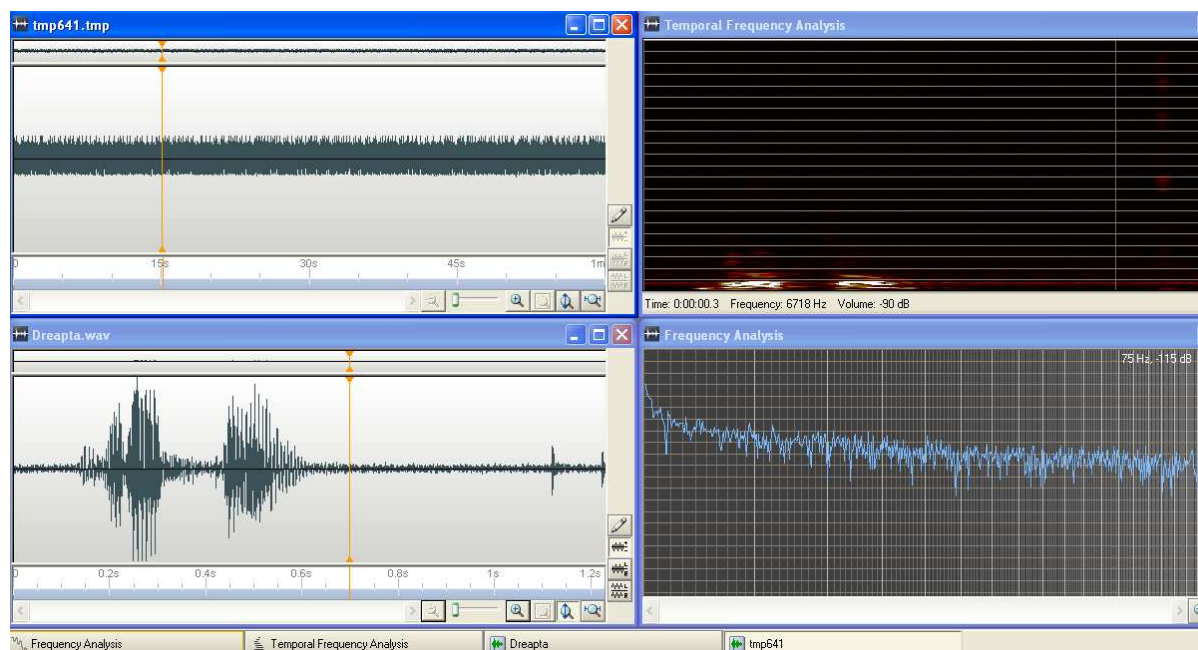


Figura 3.36: Zgomotul alb si semnalul initial – Analiza in frecventa

3.8 CoolSoft SpeakToText 2.51

Aplicatia aleasa pentru recunoasterea vocii este CoolSoft SpeakToText 2.51⁹¹.

Aceasta permite, pe langa afisarea cuvintelor pronuntate in fereastra principala si definirea unor comenzi care vor fi utile pentru a comanda aplicatia (construita in Visual Basic 2008) pentru miscarea robotului.

Principalele avantaje ale CoolSoft SpeakToText 2.51 comparativ cu alte sisteme de recunoastere a vocii sunt date de:

1. **Recunoasterea rapida (aproape in timp real) a cuvintelor pronuntate.** Aceasta *identificare se realizeaza pe baza profilului creat in meniul de control* („Control Panel”), sectiunea de voce („Speech”). Microsoft Speech Recognition 6.1 este „motorul” principal de recunoastere si adaptare, pentru o identificare mai rapida a amprentei vocale, a computerului la vocea utilizatorului

⁹⁰ Verificarea semnalului: Matsui T., Furui S., „*Similarity normalization method for speaker verification based on a posteriori probability*”, Proceedings of the ESCA Workshop on Automatic Speaker Recognition, Identification and Verification, 2004, p. 59-62.

⁹¹ Alte aplicatii de comanda vocala si recunoastere semnal vocal: Singh R., Warmuth M., Raj B., si Lamere P., „*Classification of speech applications*”, in *Proceedings of the 8th European Conference on Speech Communication and Technology*, Geneva, Switzerland, Sept. 2005, pp. 1073–1776.

uman. Astfel este asigurata unicitatea amprentei vocale. Orice alta voce nu va fi identificata cu aceeași rata de succes. Profilul creat a presupus parcurgerea mai multor sesiuni de training vocal pentru acomodarea masinii cu vocea utilizatorului.

2. **Afisarea cuvintelor in fereastra principala**: cuvintele recunoscute sunt afisate imediat in fereastra principala a aplicatiei. Pentru o mai buna interactiune cu vorbitorul, programul ofera si posibilitatea afisarii a ceea ce „se crede” ca a fost spus. *Cuvintele sunt spatiate si, daca identificarea a fost realizata de fiecare data cu succes, atunci, la final, in fereastra principala vor fi afisate toate comenzile in ordinea definita.*
3. **Interactiunea cu alte aplicatii selectate de catre utilizator**: aplicatia permite lansarea in executie a oricarei aplicatii odata cu identificarea unui cuvânt ori a unui set de cuvinte. *Este suficient sa se specifice calea catre acea aplicatie.*
4. **Definirea unui set de comenzi oricat de mare care sa deschida si/sau sa ruleze orice aplicatie dorita**. Aplicatia lasa la indemana utilizatorilor definirea oricarui set de comenzi pentru manipularea prin voce a aplicatiilor dorite, oricate si de orice natura ar fi acestea. De exemplu pot fi deschise fisiere cu extensia *.doc, .pdf, .exe, .txt, .jpeg, .gif, .png, .com, etc.*
5. **Decodificarea unor fisiere audio salvate in prealabil de utilizator**. Prin aplicatie *pot fi „deschise” fisiere salvate de catre utilizator si decodificarea mesajului transmis direct in fereastra principala. Codificarea se realizeaza in timp real intr-un timp foarte bun.*
6. **Salvarea mesajului inregistrat prin comanda vocala**: se realizeaza prin aplicatie intr-un *fișier cu extensia .txt* odata ce utilizatorul a terminat sesiunea de transmitere a comenzilor catre robot. Astfel este demonstrata si perspectiva cronologica a comenzilor trimise. Utilizatorul poate vedea la orice moment de timp lista de comenzi care au fost trimise catre utilizator.
7. **Afisarea unui ecou** pentru ca utilizatorul sa verifice faptul ca aplicatia a recunoscut semnalul primit si lanseaza in executie o anumita aplicatie dorita. Mai precis, de fiecare data cand este rostit un cuvânt in microfon, aplicatia lanseaza in executie un motor de aproximare a vocii iar in subsolul paginii sunt afisate caracterele *„I think you said...”* urmate de cuvântul sau cuvintele approximate. Problema recunoasterii amprentei vocale este una de aproximare prin algoritmi, asadar niciodata exacta 100% intrucat conditiile de mediu difera de fiecare data. Doar in cazul identificarilor semnalului vocal dintr-un anumit fisier rezultatul va fi acelasi dupa oricate sesiuni de recunoastere vocala.
8. **Deschiderea imediata a aplicatiei dorite**. Delay-ul pentru executarea unei aplicatii va fi mai mare decat cel pentru deschiderea unui fisier .doc (Word), de exemplu. Asadar recunoasterea vocala si deschiderea aplicatiilor dorite se realizeaza in timp real, precum scopul dorit.

Nota: Lista de comenzi trimise catre robot mai poate fi determinata cronologic in functie de lista de ferestre deschise in bara principala [Start] a sistemului de operare [in acest caz Microsoft Windows Xp SP3]⁹².

Aplicatia poate fi modificata astfel incat o anumita comanda sa poata fi executata pentru o anumita perioada de timp. Mai mult decat atat aceste informatii pot fi scrise intr-un fisier si citite ulterior pentru a se rula o alta comanda in acord cu cea precedenta.

Alegerea facuta in favoarea Coolsoft SpeakToText 2.51 si Microsoft Speech

Recognition 6.1 au avut in vedere si faptul ca alte programe de recunoastere vocala (precum: **Speakonia 1.3.5, Ultra Hal-Text-To-Speech Reader 1.0.5, Better Wave-To-Text 5, Taszti Speech Recognition for Windows XP 1.1.2, ReadPlease 2003 1.1, e-Speaking Voice and Speech Recognition 3.8.12, TextAloud 2.303, SmartRead 0.80, Text2Wav 2.11, Digital Dictation 7.5, Speak 1.8.84, TTS Reader 1.30, Voice Studio 1.4.9, Intelligent Voice Operating System 2.0.3, Talking Clipboard 1.3.2.0, Media Center Communicator 3.0, My voice controller 1.8.8.7, AgentReader 4.1, Talkie 1.3, Read to me Text to Speech 1.1, Expressivo 1.2.1, Speack and Mail 2000.15, CoolSpeaking 2, Dictation 2005 5.9.24, dictation Buddy 2.2, TextSpeechPro 2.2, NextUp Talker 1.019, Text Speech 3.0.0.2, Word Boc**

⁹² Young S., Russell N.H., si Russell J.H.S., „*Token passing: A simple conceptual model for connected speech recognition systems*”, Cambridge University Engineering Dept, UK, Tech. Rep. CUED/F-INFENG/TR38, 2007.

Processor 1.01, Voxx 4, Speech Expert 1.05, Me Speaking 3.03, Speech Tools 2.1, Audiomatic 1.30.07362, Voice Prompt Vocabularies Mirosoft Speech Application SDK1, NeuMemo 1.0, V:Remote 1.0, Integrated Notification System 7.4.3, iYak 1.0)⁹³ prezentau o serie de dezavantaje, printre care:

- Limitarea la unele dintre cerintele de la punctele precedente⁹⁴ [1..8]
- Posibilitatea recunoasterii vocii doar dintr-un fisier audio⁹⁵ (.mp3, .cda, .wav, etc.) si nu prin microfon.
- Imposibilitatea definirii de comenzi⁹⁶ de lansare in executie a unei aplicatii la identificarea unui semnal sonor, ci doar afisarea acestuia (si eventual salvarea intr-un fisier).
- Convertirea unui fisier text intr-un fisier audio si nu invers⁹⁷; asadar limitarea sub o alta forma a scopului initial.

Criteriau	Coolsoft Speak to Text 2.51	Speakonia 1.3.5	Ultra-Hal- Text2Speech 1.0.5	e-Speaking 3.8.12	Speak 1.8.84
Recunoastere rapida	5	4	5	4	3
Afisarea cuvintelor	4	3	3	2	0
Interactiunea cu alte aplicatii	5	3	0	4	4
Definire set comenzi	4	4	3	3	3
Decodificare audio	4	3	5	3	5
Salvare mesaj	5	3	3	4	4
Afisare mesaj ecou	5	4	5	4	3
Deschidere aplicatie	4	3	4	3	2
Total (40)	36	27	28	27	24

Tabelul 3.36: Comparatie intre mai multe programe de recunoastere comanda vocala

⁹³ X-Voice: <http://xvoice.sourceforge.net/>

⁹⁴ G-Voice: <http://www.cse.ogi.edu/~omega/gnome/gvoice/>

⁹⁵ IBM viaVoice: http://www-4.ibm.com/software/speech/dev/sdk_linux.html

⁹⁶ ISIP: <http://www.isip.msstate.edu/project/speech/>

⁹⁷ Vocalis: <http://www.vocalisspeechware.com/>



Figura 3.38: CoolSoft SpeakToText 2.51

Asadar, aplicatia va afisa la fiecare moment de timp cuvintele pe care le poate recunoaste, precum in imaginea de mai jos⁹⁸.

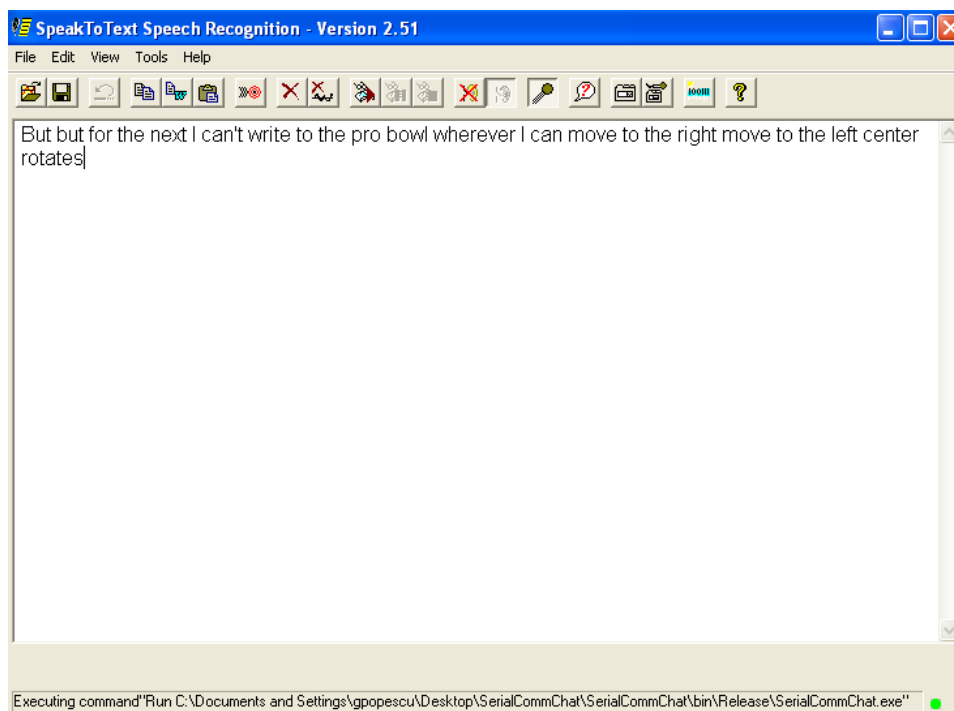


Figura 3.39: Fereastra principala a aplicatie Speak2Text. Afisarea cuvintelor recunoscute.

In continuare vom defini un set de comenzi pentru fiecare aplicatie pe care o vom rula, specifica fiecarui tip de miscare a robotului⁹⁹.

⁹⁸ Generalitati cu privire la aplicatiile de recunoastere comanda vocala si structura lor: Singh R., Warmuth M., Raj B., si Lamere P., „Classification of speech applications”, in *Proceedings of the 8th European Conference on Speech Communication and Technology*, Geneve, Switzerland, Sept. 2005, pp. 1073–1776.

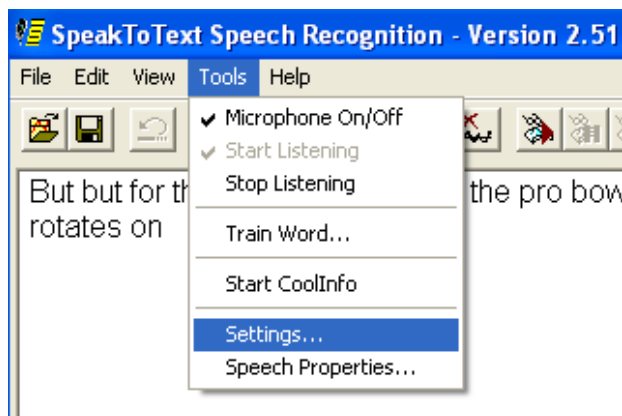


Figura 3.40: Calea catre definirea unui set de comenzi in meniul aplicatiei

Aplicatia dorita va fi lansata in executie prin **pronuntia si identificarea** unui cuvânt cheie ales de utilizator.

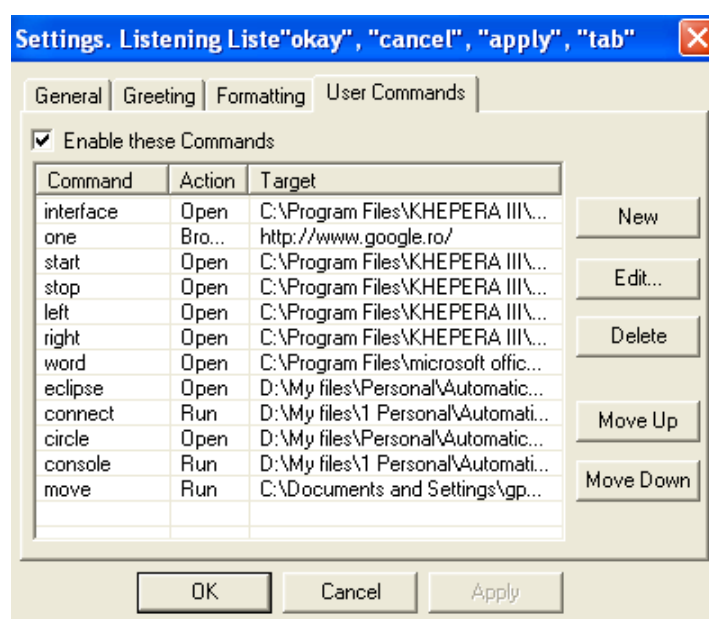


Figura 3.41: Definirea unui set de comenzi odata cu recunoastere unui cuvânt cheie.

In fereastra de mai sus pot fi observate mai multe comenzi definite, precum:

- **Interface**: va lansa in executie interfata de conexiune a robotului cu calculatorul. Programul este descoperit prin calea: C:\Program Files\KHEPERA III\KH3_Interface.exe – folder-ul ales pentru instalarea programului.
- **One**: va deschide o fereastra a browser-ului web setat „default” (in acest caz Mozilla Firefox) la adresa: <http://www.google.com>
- **Word**: va deschide Microsoft Word de la adresa: C:\Program Files\microsoft office\Office12\WINWORD.EXE
- **Move**: va rula aplicatia construita in Visual Basic 2008 pentru miscarea robotului de la adresa: C:\DocumentsandSettings\gpopescu\Desktop\SerialCommChat\SerialCommChat\bin\Release\SerialCommChat.exe. Aceasta este cea mai importanta comanda.

Diferite alte comenzi vor deschide/rula alte aplicatii care vor determina o alta miscare a robotului.

⁹⁹ Pentru mai multe aplicatii de recunoastere vocala: Young S., Russell N.H., si Russell J.H.S., „Token passing: A simple conceptual model for connected speech recognition systems”, Cambridge University Engineering Dept, UK, Tech. Rep. CUED/F-INFENG/TR38, 2007.

3.9 Aplicatia de comunicare seriala COM Kephhera III

Aplicatia de comunicare seriala va fi rulata prin SpeakToText. La identificarea unei comenzi programul va deschide interfata urmatoare¹⁰⁰.

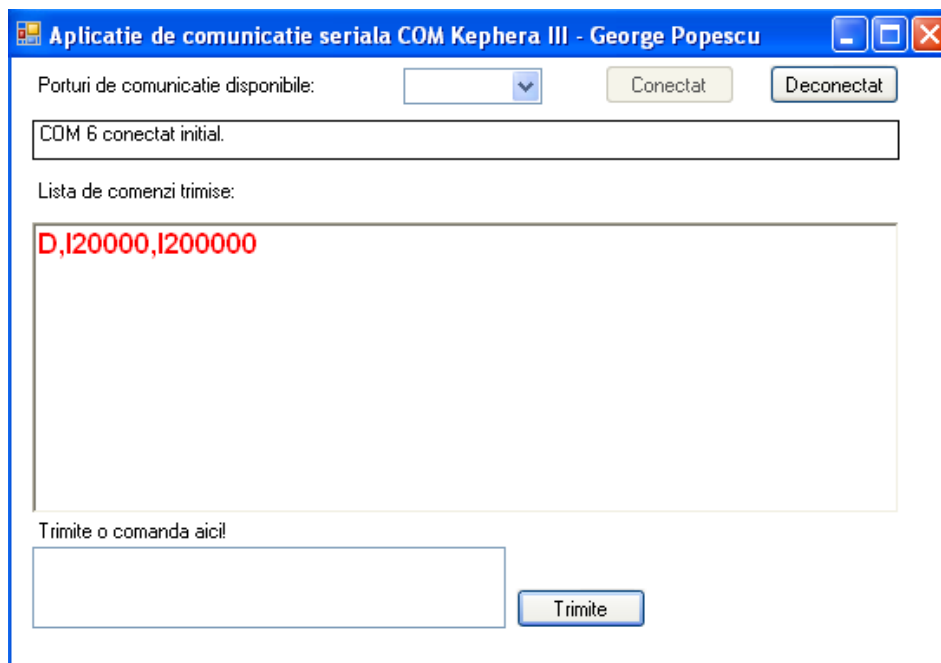


Figura 3.42: Aplicatia de comunicare seriala COM Kephhera III

Aplicatia are urmatoarele trasaturi:

- Este lansata in executie prin SpeakToText
- Porneste cu un delay de 2-3 secunde datorita conexiunii la portul serial 6 „valoare default”
- Transmite comanda afisata (in **rosu**) la conectare
- Poate transmite in orice moment orice alta comanda ce va fi inregistrata in ultimul cadran al ferestrei
- Noile comenzi (marcate cu **albastru**) vor putea fi introduse de catre utilizator prin apasarea butonului „Trimite”
- Permite conectarea si deconectarea de la portul setat initial in orice moment
- Aplicatia poate fi modificata astfel incat conexiunea sa se realizeze pe orice alt port descoperit liber¹⁰¹

Observatie:

Valoarea **6** a portului serial a fost aleasa deoarece comanda (inregistrata cu **rosu** in figura de mai sus) poate fi transmisa catre robot prin parametrii doriti. Pentru COM22, de exemplu, aceeasi comanda nu ar fi putut fi transmisa.

Sursa aplicatiei se gaseste in **capitolul 5 – Sursa Software**¹⁰².

¹⁰⁰ Descrieri similare au fost folosite si in: Bonafonte A., Moreno A., Adell J., D. Aguero P., Banos E., Daniel Erro, Ignasi Esquerra, Javier Perez, Tatyana Polyakova, „The UPC TTS System Description for the 2008 Blizzard Challenge”, Universitat Politecnica de Catalunya, Spain, 2008.

¹⁰¹ In eventualitatea in care se doreste interconectarea mai multor aplicatii de comanda vocala: Young S., Russell N.H., si Russell J.H.S., „Token passing: A simple conceptual model for connected speech recognition systems”, Cambridge University Engineering Dept, UK, Tech. Rep. CUED/F-INFENG/TR38, 2007.

3.10 Concluzii si directii viitoare de cercetare

Proiectul si-a propus si a realizat un mediu software de comunicare vocala intre un utilizator uman si robotul Kephera III. Mediul de programare Java permite dezvoltarea mai avansata a fiecaruia dintre modulele structurale prezentate. Eficienta poate fi asadar perceputa in scaderea timpului de raspuns si identificare a semnalului prin constructia unei gramatici mai simple sau prin dezvoltarea unor metode de identificare a semnalului vocii umane mai rapide. Pe de alta parte, programarea in mediul Visual Basic 2008 ofera o interactiune mai facila intre utilizator si aplicatie in definirea „actiunilor” considerate pentru fiecare miscare (evenimente).

Obiectivul de comanda si control vocal al robotului Kephera III a fost indeplinit prin constructia aplicatiei de recunoastere comanda vocala utilizator si apelarea unor functii specifice catre robot pentru realizarea miscarii pe o directie dorita. Performantele obtinute sunt in stransa legatura cu definirea unui profil de utilizator (mai multe sesiuni de training vocal vor determina o recunoastere mai rapida a comenzilor catre robot) si cu performantele computerului care ruleaza aplicatia de comanda. Am reusit reducerea timpului de comanda la numai o secunda si o crestere a acuratetei recunoasterii vocii la 80% din comenzile pronuntate.

Comanda vocala – sub toate aspectele prezentate in acesta lucrare impreuna cu alte aplicatii si dezvoltari sintetizate in literatura ingineriasca de specialitate – reprezinta una dintre directiile de cercetare cele mai „fierbinti” ale dezvoltarii industriei de IT de astazi. Omul si-a dorit o interactiune cat mai facila cu robotii sau computerele din jur a.i. comanda sa se realizeze intr-un timp cat mai scurt, bineinteles fiind cat mai exacta. Practic dorim ca robotii sa ne „citeasca” cat mai bine gandurile fara a introduce erori de identificare sau sa ne directioneze catre surse irelevante de informatii, oferind rezultate indezirabile. Rata de succes in acest caz este unul dintre parametrii pe care dorim sa ii ajustam prin comanda vocala.

Am conceput si descris modul de functionare al unui sistem de comanda vocala si interactiune dintre un utilizator uman si Kephera III. Fiecare dintre componentele necesare sistemului este descrisa in capitolele precedente specificandu-se totodata punctele cheie, asigurarea succesului pentru fiecare scop in parte si principalele limitari. Sistemul este construit pe baza unor tehnici de inteligenta artificiala care presupune adaptarea continua la comenzile inregistrate precum si sincronizarea cu baza de cunostinte.

Fie ca este vorba de un model simplu pe care dorim sa-l implementam (precum un software de interfatare vocala dintre un utilizator uman si un robot inteligent al unui call-center) sau de unul mai complicat (precum cautarea unor anumite elemente intr-o baza de date de ordinul sutelor de TB de fisiere audio si extragerea informatiilor relevante), trebuie avute in vedere metode eficiente de dezvoltare pentru fiecare dintre modulele prezentate (Interfata primara, Lingvistul, Modulul de Identificare de semnal, etc.). Fiecare dintre acestea poate fi la randul sau restructurat (in limbaje de programare orientate pe obiect) cu scopul de a diminua timpul de raspuns, a creste precizia si a scadea numarul de cautari in baza de date conform unui sablon auditiv.

Sistemul reprezinta o singura implementare a modelului acustic. Sunt produse modelele HMM cu o topologie fixata si unitatile independente de context. Parametrii asociati functiilor de probabilitate descinse din limbaj si graful de cautare sunt interconectate. Intrucat nu exista restrictii pentru implementare, sistemul ofera posibilitatea de a manevra aceste functii Markov cu o topologie arbitrara si unitatile deduse din context, atat la stanga, cat si la dreapta, de lungime arbitrara. Intrucat aplicatia si modelul acustic permit inscrierea oricat de fina a parametrilor modelului pot fi deduse avantaje ce rezulta din aceste capabilitati precum scaderea timpului si cresterea vitezei de executie.

Pentru intelegerea intregului sistem precum si pasii necesari pentru instalarea si configurarea sa vor fi necesare pe de o parte notiuni elementare ale limbajelor de programare (in special Java) dar si notiuni legate de configurarea serverelor, prelucrarea si parsarea informatiilor XML, etc. Managerul de cautare furnizeaza un teren deosebit de fertil pentru implementarea mai multor abordari de cautare incluzand algoritmul A*, cel de potrivire rapida, bi-directional si multiplii algoritmi de pasare a informatiilor legate de topologie.

¹⁰² Clasificari ale unor aplicatii similare de recunoastere vocala: Singh R., Warmuth M., Raj B., si Lamere P., „*Classification of speech applications*”, in *Proceedings of the 8th European Conference on Speech Communication and Technology*, Geneve, Switzerland, Sept. 2005, pp. 1073–1176.

George POPESCU – Comandă vocală Kephera III

Aplicatia de comanda vocala, „TextToSpeak” permite lansarea in executie a oricarei aplicatii la identificarea unui cuvânt cheie prestabilit de utilizator insa incarcarea in memoria calculatorului a tuturor acestor aplicatii va incetini vizibil procesul de rulare a fiecareia dintre ele.

Desi multe, limitarile sistemului pot fi ajustate, conducand la o mai buna acuratete a identificarii vocii, cuvintelor cheie, precum si o independenta sporita a modulului de control. Vocea umana, interpretata ca semnal, prezinta caracteristici neperiodice, desi similare in cazul pronuntiei acelorasi cuvinte. Aproximarea realizata pentru o anumita plaja de valori conduce la sintetizarea unui anumit sunet, cuvânt, expresie.

Pentru a mai buna intelegere a tipurilor de miscare si controlul acestora este recomandata parcurgerea manualului Kephera III – parametrii trimisi prin aplicatie pot fi ajustati numai insa de rularea aplicatiei, in codul ei sursa.

Urmarind scopul initial, aplicatia are capacitatea de a permite urmarirea comenzii vocale in timp real iar varietatea comenzilor trimise catre robot, practic nelimitata prin ajustarea acestor parametri, ramane la latitudinea utilizatorului¹⁰³.

Trebuie permanent realizat un compromis intre precizia de identificare si puterea de prelucrare astfel incat comanda / actiunea robotului Kephera III sa fie realizata intr-un timp minim, ideal. Timpul de prelucrare a informatiei tine pe de o parte de resursele hardware (procesor, memorie, etc.) iar pe de cealalta de „acomodarea” computerului cu vocea utilizatorului.

Progrese ulterioare (traduse prin micșorarea timpului de raspuns) pot fi inregistrate prin „specializarea” profilului inregistrat initial. Mai mult decat atat acesta poate fi redus prin folosirea unui aplicatii care sa „deschida” comunicatia seriala pe portul dorit initial astfel incat la o lansare in executie a aplicatiei, sa fie nevoie ca DOAR comanda catre robot sa fie transmisa.

Directii viitoare de cercetare pot fi concretizate prin inglobarea in aceeasi aplicatie a unui set de comenzi predefinite si alegerea uneia dintre acestea prin comanda vocala. Acest aspect poate fi realizat numai prin folosirea sursei unui sintetizator audio pus in corespondenta cu transmiterea comenzii (parametrizat) pe interfata de comunicatie seriala. Deasemenea, tot aici, utilizatorul poate specifica – prin comanda vocala – portul dorit. Practic devin nelimitate posibilitatile de ajustare ale unui software de recunoastere si aplicare a unei comenzi vocale.

Dezvoltari ulterioare ale proiectului de comanda vocala pentru Kephera III pot conduce la miscari mai complexe ale robotului pentru o anumita distanta precizata (pot fi realizate calcule exacte cu ajutorul cartii tehnice a robotului), rotirea sub un anumit unghi, evitarea obstacolelor de la o anumita distanta precizata anterior, recunoasterea anumitor sunete si apropierea sau departarea de acestea cu ajutorul senzorilor ultrasonici din dotarea robotului, calculul si interpretarea unor anumite zone (cartografiere) prin stocarea unor parametrii intr-o baza de date conform unui camp audio, diseminarea mai multor surse sonore suprapuse si acceptarea controlului din partea uneia, si multe altele.

In general un proiect de comanda vocala isi gasesti usor domenii de aplicatii dintre cele mai diverse, precum comanda vocala in mediul industrial (comanda catre robotii industriali), in domeniul public (transport, sanatate, etc.), domenii aferente cercetarii (comanda sondelor spatiale prin controlul uman – voce – a unui semnal purtator de pe Pamant si o multitudine de alte aplicatii in domeniul inteligentei artificiale si al roboticii.

Sunt recunoscator intregii echipe a catedrei ED209 pentru sprijinul acordat si multele sfaturi utile in realizarea formei finale a acestei lucrari. Multumiri speciale aduc doamnei profesoare Simona Caramihai pentru indrumarea oferita pentru acest proiect de licenta in particular insa si pentru toate celelalte lucrari prezentate de-a lungul anilor la Sesiunea de Comunicari Stiintifice Studentesti.

¹⁰³ Pentru interconectarea mai multor aplicatii de comanda vocala create de utilizator in diferite medii de programare: Young S., Russell N.H., si Russell J.H.S., „*Token passing: A simple conceptual model for connected speech recognition systems*”, Cambridge University Engineering Dept, UK, Tech. Rep. CUED/F-INFENG/TR38, 2007.

4 IMPLEMENTARE: MEDIU SOFTWARE DE COMANDA VOCALA PENTRU KEPHERA III

4.1 Aplicatie de comunicatie seriala COM Kephera III

```
'           George Popescu
'- Facultatea de Automatica si Calculatoare -
'   - Universitatea Politehnica Bucuresti -
'           Bucuresti 2009

Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort

    'Aplicatia se va lansa automat prin comanda vocala!
    'Pentru aceasta este nevoie ca in procedura de mai jos
    'conexiunea la portul de comunicatie SerialComm6 sa fie
    'realizata odata cu lansarea aplicatiei, i.e. imediat

    Private Sub Form1_Load( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        Try
            With serialPort
                'Numarul 2 din linia de mai sus specifica portul 6
                'pentru realizarea conexiunii seriale
                .PortName = My.Computer.Ports.SerialPortNames(2)

                'In continuare vom defini CONFIGURATIA ACCEPTATA DE KIII!

                '115200bps data rate
                .BaudRate = 115200

                'no parity
                .Parity = IO.Ports.Parity.None

                '8 data bits
                .DataBits = 8

                'one stop bit
                .StopBits = IO.Ports.StopBits.One

            End With
            serialPort.PortName = My.Computer.Ports.SerialPortNames(2)
            serialPort.Open()

            'Mesaj pentru specificarea conexiunii la portul 6
            lblMessage.Text = "COM 6 conectat initial."

            'Managementul celor 2 butoane: "Conectat" si "Deconectat"!
            btnConnect.Enabled = False
            btnDisconnect.Enabled = True

        Catch ex As Exception
```

George POPESCU – Comandă vocală Kephera III

```
        MsgBox(ex.ToString)
    End Try
    'Inscrierea comenzii in Textbox-ul atasat aplicatiei.
    Try
        serialPort.Write(comanda & vbCrLf)
        updateTextBox()
        With comanda
            End With
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub

'Definim cate o comanda PUBLICA (i.e. pentru intreaga aplicatie)
'acceptata de robot pentru fiecare tip de miscare dorita
'De exemplu, comanda urmatoare defineste o miscare circulara
Public Const comanda As String = "D,120000,1200000"

Private Sub DataReceived( _
    ByVal sender As Object, _
    ByVal e As System.IO.Ports.SerialDataReceivedEventArgs) _
    Handles serialPort.DataReceived

    txtDataReceived.Invoke(New _
        myDelegate(AddressOf updateTextBox), _
        New Object() {})
End Sub

'Butonul pentru trimiterea altor comenzi va fi functional
'daca se doreste interactiunea de la tastatura a utilizatorului!
'Nota: Comanda vocala, predefinita va fi afisata cu ROSU iar cea
'data de utilizator ulterior cu ALBASTRU!
Private Sub btnSend_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles btnSend.Click
    Try
        serialPort.Write(vbCrLf & txtDataToSend.Text & vbCrLf)
        With txtDataReceived
            .SelectionColor = Color.Blue
            .AppendText(vbCrLf & txtDataToSend.Text & vbCrLf)
            .ScrollToCaret()
        End With
        txtDataToSend.Text = String.Empty
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub

Public Delegate Sub myDelegate()
Public Sub updateTextBox()
    With txtDataReceived
        .Font = New Font("Arial", 12.0!, FontStyle.Bold)
        .SelectionColor = Color.Red
        .AppendText(comanda)
        .ScrollToCaret()
        'Mai sus am definit cateva stiluri pentru a evidentia comanda aleasa
    End With
End Sub

'Butonul de conectare la unul dintre porturile seriale disponibile.
Private Sub btnConnect_Click( _
    ByVal sender As System.Object, _
```

George POPESCU – Comandă vocală Kephera III

```
ByVal e As System.EventArgs) _
Handles btnConnect.Click
If serialPort.IsOpen Then
    serialPort.Close()
End If
Try
    With serialPort
        .PortName = My.Computer.Ports.SerialPortNames(2)
        .BaudRate = 115200
        .Parity = IO.Ports.Parity.None
        .DataBits = 8
        .StopBits = IO.Ports.StopBits.One
    End With
    serialPort.PortName = My.Computer.Ports.SerialPortNames(2)
    serialPort.Open()

    lblMessage.Text = "COM 6 conectat."
    btnConnect.Enabled = False
    btnDisconnect.Enabled = True

Catch ex As Exception
    MsgBox(ex.ToString)
End Try
End Sub
'Butonul pentru Deconectare la portul serial ales anterior.
Private Sub btnDisconnect_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
Handles btnDisconnect.Click
    Try
        'Inchiderea portului si afisarea unui mesaj de confirmare.
        serialPort.Close()
        lblMessage.Text = serialPort.PortName & " deconectat"
        'Schimbarea starii celor 2 butoane: "Conectare" si "Deconectare"
        btnConnect.Enabled = True
        btnDisconnect.Enabled = False
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
End Class
'Observatie: Numarul si valoarea porturilor active/deschise de comunicatie
seriala
'pot fi diferite de la un calculator la altul. De aceea este recomandata
detectia
'acestora folosind mai intai aplicatia de interactiune cu utilizatorul si apoi
'cea de comanda vocala.

Copyright(c) - George Popescu - 2009 - All rights reserved!
```

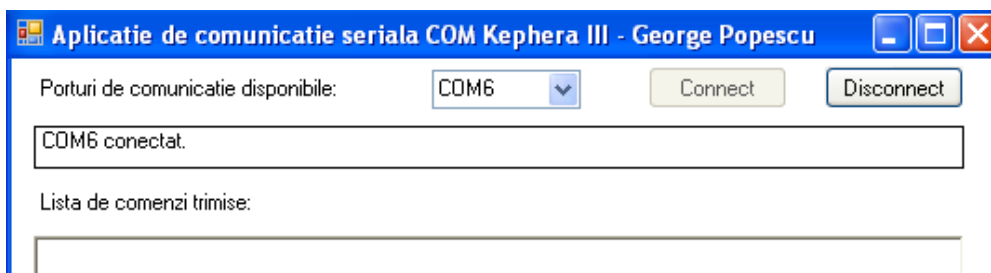


Figura 4.1: Miscare circulara cu viteza optima

4.2 Circular optim incet

Aplicatia va avea ca rezultat miscarea circulara a robotului cu o viteza optima.

```
Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort
    'Aplicatia va fi lansata in executie automat
    'pentru o anumita comanda specificata
    Private Sub Form1_Load( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        Try
            serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
            serialPort.BaudRate = 115200
            serialPort.Parity = IO.Ports.Parity.None
            serialPort.DataBits = 8
            serialPort.StopBits = IO.Ports.StopBits.One
            serialPort.Open()
            serialPort.Write(comanda & vbCrLf)
            serialPort.Close()
            With comanda
            End With
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub
    Public Const comanda As String = "D,10000,13000"
End Class
```

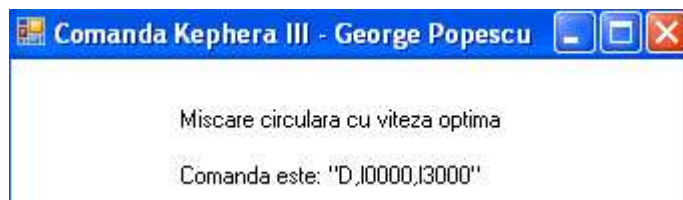


Figura 4.2: Miscare circulara cu viteza optima

4.3 Inainte optim

```
Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort
    Private Sub Form1_Load( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        Try
            serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
            serialPort.BaudRate = 115200
            serialPort.Parity = IO.Ports.Parity.None
            serialPort.DataBits = 8
            serialPort.StopBits = IO.Ports.StopBits.One
            serialPort.Open()
            serialPort.Write(comanda & vbCrLf)
            serialPort.Close()
            With comanda
            End With
        End Try
    End Sub
End Class
```

```
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
Public Const comanda As String = "D,13000,13000"
End Class
```

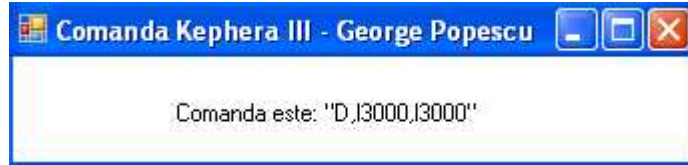


Figura 4.3: Miscare inainte cu viteza optima

4.4 Inapoi optim

```
Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort
    'Aplicatia va fi lansata in executie automat
    'pentru o anumita comanda specificata
    Private Sub Form1_Load( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        Try
            serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
            serialPort.BaudRate = 115200
            serialPort.Parity = IO.Ports.Parity.None
            serialPort.DataBits = 8
            serialPort.StopBits = IO.Ports.StopBits.One
            serialPort.Open()
            serialPort.Write(comanda & vbCrLf)
            serialPort.Close()
            With comanda
            End With
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub
    Public Const comanda As String = "D,1-3000,1-3000"
End Class
```

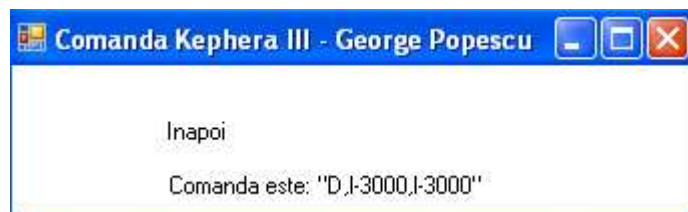


Figura 4.4: Miscare inainte cu viteza optima

4.5 Stop

```
Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort
    'Aplicatia va fi lansata in executie automat
```

```
'pentru o anumita comanda specificata
Private Sub Form1_Load( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles MyBase.Load
    Try
        serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
        serialPort.BaudRate = 115200
        serialPort.Parity = IO.Ports.Parity.None
        serialPort.DataBits = 8
        serialPort.StopBits = IO.Ports.StopBits.One
        serialPort.Open()
        serialPort.Write(comanda & vbCrLf)
        serialPort.Close()
        With comanda
            End With
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
Public Const comanda As String = "D,10,10"
End Class
```

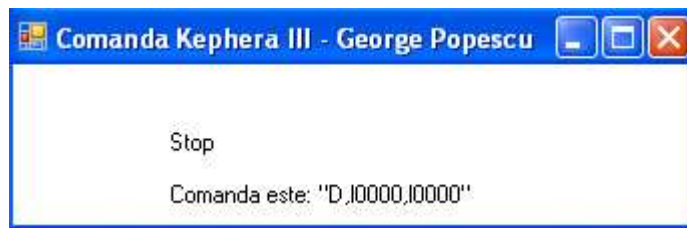


Figura 4.5: Comanda pentru oprirea robotului

4.6 Inainte incet

```
Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort
    Private Sub Form1_Load( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        Try
            serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
            serialPort.BaudRate = 115200
            serialPort.Parity = IO.Ports.Parity.None
            serialPort.DataBits = 8
            serialPort.StopBits = IO.Ports.StopBits.One
            serialPort.Open()
            serialPort.Write(comanda & vbCrLf)
            serialPort.Close()
            With comanda
                End With
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub
    Public Const comanda As String = "D,11500,11500"
End Class
```

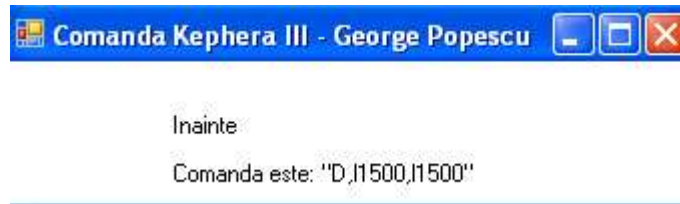


Figura 4.6: Comanda pentru injumatatirea vitezei robotului

4.7 Inainte rapid

```
Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort
    'Aplicatia va fi lansata in executie automat
    'pentru o anumita comanda specificata
    Private Sub Form1_Load( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        Try
            serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
            serialPort.BaudRate = 115200
            serialPort.Parity = IO.Ports.Parity.None
            serialPort.DataBits = 8
            serialPort.StopBits = IO.Ports.StopBits.One
            serialPort.Open()
            serialPort.Write(comanda & vbCrLf)
            serialPort.Close()
            With comanda
                End With
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub
    Public Const comanda As String = "D,16000,16000"
End Class
```

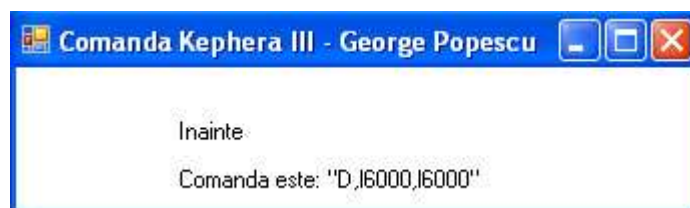


Figura 4.7: Comanda pentru dublarea vitezei robotului

4.8 Conectare automata la portul COM (6)

```
Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort

    'Conexiunea la portul de comunicatie SerialComm6 este
    'realizata odata cu lansarea aplicatiei

    Private Sub Form1_Load( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
```


George POPESCU – Comandă vocală Kephera III

```
Handles MyBase.Load
Try
    With serialPort
        .PortName = My.Computer.Ports.SerialPortNames(0)
        .BaudRate = 115200
        .Parity = IO.Ports.Parity.None
        .DataBits = 8
        .StopBits = IO.Ports.StopBits.One
    End With
    serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
    serialPort.Open()
    lblMessage.Text = serialPort.PortName & " conectat initial."
    btnConnect.Enabled = False
    btnDisconnect.Enabled = True
Catch ex As Exception
    MsgBox(ex.ToString)
End Try

End Sub
Private Sub btnConnect_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles btnConnect.Click
    If serialPort.IsOpen Then
        serialPort.Close()
    End If
    Try
        With serialPort
            .PortName = My.Computer.Ports.SerialPortNames(0)
            .BaudRate = 115200
            .Parity = IO.Ports.Parity.None
            .DataBits = 8
            .StopBits = IO.Ports.StopBits.One
        End With
        serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
        serialPort.Open()

        lblMessage.Text = serialPort.PortName & " conectat"
        btnConnect.Enabled = False
        btnDisconnect.Enabled = True

    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
Private Sub btnDisconnect_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles btnDisconnect.Click
    Try
        serialPort.Close()
        lblMessage.Text = serialPort.PortName & " deconectat"
        btnConnect.Enabled = True
        btnDisconnect.Enabled = False
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
End Class
```

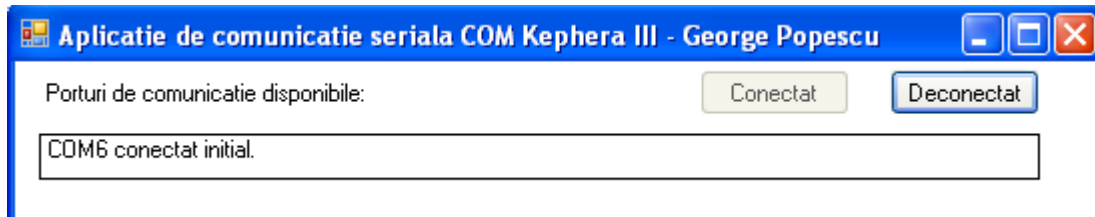


Figura 4.8: Conexiune automata la portul COM (6)

4.9 Deconectare de la portul COM (6)

```
'
    George Popescu
'- Facultatea de Automatica si Calculatoare -
'   - Universitatea Politehnica Bucuresti -
'
    Bucuresti 2009

Public Class Form1
    Dim WithEvents serialPort As New IO.Ports.SerialPort

    'Conexiunea la portul de comunicare SerialComm6 este
    'inchisa prin lansarea in executie a acestei aplicatii

    Private Sub Form1_Load( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles MyBase.Load
        Try
            With serialPort
                .PortName = My.Computer.Ports.SerialPortNames(0)
                .BaudRate = 115200
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One

            End With
            serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
            serialPort.Close()

            'Mesaj pentru specificarea deconectarii de la portul 6 - initial
            lblMessage.Text = serialPort.PortName & " deconectat initial"

            'Managementul celor 2 butoane: "Conectat" si "Deconectat"!
            btnConnect.Enabled = True
            btnDisconnect.Enabled = False

        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub

    Private Sub btnConnect_Click( _
        ByVal sender As System.Object, _
        ByVal e As System.EventArgs) _
        Handles btnConnect.Click
        If serialPort.IsOpen Then
            serialPort.Close()
        End If
    End Sub
End Class
```

George POPESCU – Comandă vocală Kephera III

```
With serialPort
    .PortName = My.Computer.Ports.SerialPortNames(0)
    .BaudRate = 115200
    .Parity = IO.Ports.Parity.None
    .DataBits = 8
    .StopBits = IO.Ports.StopBits.One
End With
serialPort.PortName = My.Computer.Ports.SerialPortNames(0)
serialPort.Open()

lblMessage.Text = serialPort.PortName & " conectat "
btnConnect.Enabled = False
btnDisconnect.Enabled = True

Catch ex As Exception
    MsgBox(ex.ToString)
End Try
End Sub
'Butonul pentru Deconectare la portul serial ales anterior.
Private Sub btnDisconnect_Click( _
    ByVal sender As System.Object, _
    ByVal e As System.EventArgs) _
    Handles btnDisconnect.Click
    Try
        'Inchiderea portului si afisarea unui mesaj de confirmare.
        serialPort.Close()
        lblMessage.Text = serialPort.PortName & " deconectat "
        'Schimbarea starii celor 2 butoane: "Conectare" si "Deconectare"
        btnConnect.Enabled = True
        btnDisconnect.Enabled = False
    Catch ex As Exception
        MsgBox(ex.ToString)
    End Try
End Sub
End Class
```

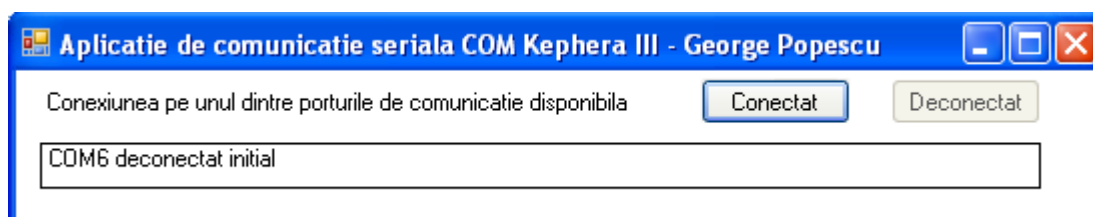


Figura 4.9: Deconectarea automata de la portul COM (6)

4.10 Programul pentru recunoastere voce

```
Imports System
Imports System.Data
Imports System.Deployment
Imports System.Drawing
Imports System.Windows.Forms
Imports System.Xml

Imports SpeechLib

Public Class Form1

    Dim WithEvents RecoContext As SpSharedRecoContext
```

George POPESCU – Comandă vocală Kephera III

```
'Principalul obiect pentru recunoastere vocii - va fi folosit in intregul
program. Shared Object.
    Dim Grammar As ISpeechRecoGrammar
'Obiectul pentru gramatica astfel incat programul sa stie ceea ce recunoaste -
Obiect Instantiat.
    Dim CharCount As Integer
'Este folosit pentru a numara literele care sunt introduse in text box.

    'Butonul de Start. Va porni recunoasterea vocala si intregul proces.
    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click

        'Mai intai se verifica daca recunoasterea a fost incarcata inainte. Daca
nu, permite incarcarea ei.
        If (RecoContext Is Nothing) Then
            RecoContext = New SpSharedRecoContextClass
'Creaza o noua clasa pentru recunoastere.
            Grammar = RecoContext.CreateGrammar(1)
'Setare gramatica
            Grammar.DictationLoad()
'Incarcare gramatica
            End If

            lblStatus.Text = "Recunoastere a inceput!"
'Modifica sigla pentru a anunta utilizatorul.
            Grammar.DictationSetState(SpeechRuleState.SGDSActive)
'Porneste recunoasterea. Acest pas este cel mai important.
'Urmatoarea secventa este pentru ca utilizatorul sa nu "strice" programul
'prin incercarea de a porni recunoasterea vocala dupa ce aceasta deja a inceput.
            btnStart.Enabled = False
            btnStop.Enabled = True
        End Sub

    'Butonul de Stop. Va opri recunoasterea vocala.
    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStop.Click

        Grammar.DictationSetState(SpeechRuleState.SGDSInactive)
'Inchide recunoasterea.
            lblStatus.Text = "Recunoastere oprita"
'Modifica label.
'In nou, este pentru ca utilizatorul sa nu "strice" neintentionat programul.
            btnStart.Enabled = True
            btnStop.Enabled = False
        End Sub

    'Aceasta este functia de ipoteza. Ipoteza nu este valoarea finala a
recunoasterii. Va fi incarcata de mai multe ori pentru fiecare cuvant.
    'Acest pas nu este cerut de recunoasterea finala. Dar este vital pentru
intelegerea lui.
    Private Sub OnHypo(ByVal StreamNumber As Integer, ByVal StreamPosition As
Object, ByVal Result As ISpeechRecoResult) Handles RecoContext.Hypothesis

        btnStop.Enabled = False
'Nu permite utilizatorului sa opreasca recunoasterea pana ce ea nu este
completa.
'Butonul se va re-initializa in evenimentul OnReco
'Urmatoarele sunt pentru a nu fi nevoie sa se afiseze acelasi text de mai multe
ori.
'Este util pentru a nu realiza identificare ne-necesare.
        If lblStatus.Text <> "Receiving" Then
```

```
        lblStatus.Text = "Receiving"  
    End If  
End Sub  
  
'Aceasta functie este incarcata atunci cand motorul de recunoastere  
detecteaza un set de cuvinte.  
'Aceasta este ceea ce se va folosi sau trimite.  
Private Sub OnReco(ByVal StreamNumber As Integer, ByVal StreamPosition As  
Object, ByVal RecognitionType As SpeechRecognitionType, ByVal Result As  
ISpeechRecoResult) Handles RecoContext.Recognition  
  
    Dim recoResult As String = Result.PhraseInfo.GetText  
'Creaza un nou sir de caractere si asigneaza textul recunoscut acestuia.  
  
'Urmatorul bloc va afisa in textbox-ul aplicatiei.  
    TxtBox.SelectionStart = CharCount  
    TxtBox.SelectedText = recoResult & " "  
    CharCount = CharCount + 1 + Len(recoResult)  
  
    'O alta varianta ar fi fost folosirea urmatoarei linii de comanda.  
    'SendKeys.Send(recoResult & " ") 'Aceasta linie trimite rezultatul via  
SendKeys ferestrei initiale.  
  
    lblStatus.Text = "Am terminat dictarea!"  
    btnStop.Enabled = True  
End Sub  
  
End Class
```

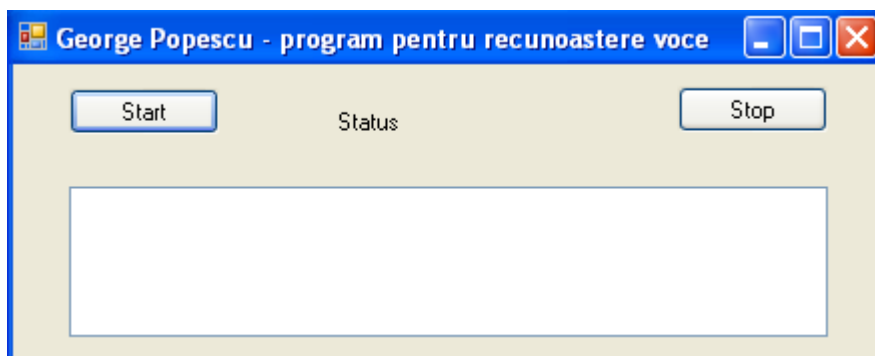


Figura 4.10: Aplicatia pentru recnoastere voce

4.11 Un traseu creat de utilizator

Robotul se va deplasa pe un traseu care descrie un patrat. Comenzile sunt date in randurile de mai jos, ca si comentarii. Scopul acestui program a fost acela de a folosi cat mai multe si mai diversificate comenzi: start, stop, delay, variatii ale vitezei, etc.

```
Public Class Form1  
    Dim WithEvents serialPort As New IO.Ports.SerialPort  
    Private Sub Form1_Load( _  
        ByVal sender As System.Object, _  
        ByVal e As System.EventArgs) _  
        Handles MyBase.Load  
        Try  
            serialPort.PortName = My.Computer.Ports.SerialPortNames(0)  
            serialPort.BaudRate = 115200  
            serialPort.Parity = IO.Ports.Parity.None
```

George POPESCU – Comandă vocală Kephera III

```
serialPort.DataBits = 8
serialPort.StopBits = IO.Ports.StopBits.One
serialPort.Open()

serialPort.Write(comanda1 & vbCrLf)
System.Threading.Thread.Sleep(3000)

serialPort.Write(comanda2 & vbCrLf)
System.Threading.Thread.Sleep(3000)

serialPort.Write(comanda3 & vbCrLf)
System.Threading.Thread.Sleep(3000)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda4 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda5 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda6 & vbCrLf)

serialPort.Write(comanda7 & vbCrLf)
System.Threading.Thread.Sleep(3000)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda8 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda9 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda10 & vbCrLf)

serialPort.Write(comanda11 & vbCrLf)
System.Threading.Thread.Sleep(3000)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda12 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda13 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda14 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda15 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda16 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda17 & vbCrLf)

System.Threading.Thread.Sleep(3000)
serialPort.Write(comanda18 & vbCrLf)

serialPort.Close()
With comanda1
End With
```

```
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
    End Sub
    Public Const comanda1 As String = "D,13000,13000"
        ' miscare inainte optim
    Public Const comanda2 As String = "D,10000,100000"
        ' stop
    Public Const comanda3 As String = "D,10000,13000"
        ' incet circular : stanga
    Public Const comanda4 As String = "D,10000,100000"
        ' stop
    Public Const comanda5 As String = "D,13000,13000"
        ' inainte
    Public Const comanda6 As String = "D,10000,100000"
        ' stop
    Public Const comanda7 As String = "D,13000,10000"
        ' dreapta
    Public Const comanda8 As String = "D,10000,100000"
        ' stop
    Public Const comanda9 As String = "D,19000,19000"
        ' inainte
    Public Const comanda10 As String = "D,100000,100000"
        ' stop
    Public Const comanda11 As String = "D,13000,10000"
        ' dreapta
    Public Const comanda12 As String = "D,10000,100000"
        ' stop
    Public Const comanda13 As String = "D,16000,16000"
        ' inainte
    Public Const comanda14 As String = "D,10000,10000"
        ' stop
    Public Const comanda15 As String = "D,17000,10000"
        ' dreapta
    Public Const comanda16 As String = "D,10000,10000"
        ' stop
    Public Const comanda17 As String = "D,115000,115000"
        ' inainte
    Public Const comanda18 As String = "D,10000,10000"
        ' stop definitiv
End Class
```

4.12 Aplicatia de control si comanda vocala Kephera III

Aplicatia permite controlul robotului atat prin voce cat si prin intermediul butoanelor prezente pe interfata principala.



Figura 4.12: Aplicatia de control si comanda vocala Kephera III

George POPESCU – Comandă vocală Kephera III

```
Imports System
Imports System.Data
Imports System.Deployment
Imports System.Drawing
Imports System.Windows.Forms
Imports System.Xml
Imports System.Diagnostics

Imports SpeechLib

Public Class Form1

    Dim WithEvents RecoContext As SpSharedRecoContext
    Dim Grammar As ISpeechRecoGrammar
    Dim CharCount As Integer
    Private Sub btnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStart.Click

        If (RecoContext Is Nothing) Then
            RecoContext = New SpSharedRecoContextClass
            Grammar = RecoContext.CreateGrammar(1)
            Grammar.DictationLoad()
        End If

        lblStatus.Text = "Recunoasterea a inceput!"
        Grammar.DictationSetState(SpeechRuleState.SGDSActive)
        btnStart.Enabled = False
        btnStop.Enabled = True

    End Sub
    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStop.Click

        Grammar.DictationSetState(SpeechRuleState.SGDSInactive)
        lblStatus.Text = "Recunoastere oprita"
        btnStart.Enabled = True
        btnStop.Enabled = False
    End Sub

    Private Sub OnHypo(ByVal StreamNumber As Integer, ByVal StreamPosition As
Object, ByVal Result As ISpeechRecoResult) Handles RecoContext.Hypothesis

        btnStop.Enabled = False
        If lblStatus.Text <> "Primire comenzi" Then
            lblStatus.Text = "Primire comenzi"
        End If
    End Sub

    Private Sub OnReco(ByVal StreamNumber As Integer, ByVal StreamPosition As
Object, ByVal RecognitionType As SpeechRecognitionType, ByVal Result As
ISpeechRecoResult) Handles RecoContext.Recognition

        Dim recoResult As String = Result.PhraseInfo.GetText
        TextBox.SelectionStart = CharCount
        TextBox.SelectedText = recoResult & " "
        CharCount = CharCount + 1 + Len(recoResult)

        Select Case recoResult
            Case "forward"
```


George POPESCU – Comandă vocală Kephera III

```
System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\1Inainte
Optim.exe")
    Case "Forward"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\1Inainte
Optim.exe")
    Case "back"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\4Inapoi
Optim.exe")
    Case "Back"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\4Inapoi
Optim.exe")
    Case "circle"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\7Circular
Optim Incet.exe")
    Case "Circle"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\7Circular
Optim Incet.exe")
    Case "left"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\5Stanga
Optim.exe")
    Case "Left"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\5Stanga
Optim.exe")
    Case "right"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\6Dreapta
Optim.exe")
    Case "Right"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\6Dreapta
Optim.exe")
    Case "10"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\9Circular
Extins.exe")
    Case "happy"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\9Circular
Extins.exe")
    Case "slow"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\2Inainte
incet.exe")
    Case "Slow"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\2Inainte
incet.exe")
    Case "fast"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\3Inainte
Rapid.exe")
    Case "Fast"
```

George POPESCU – Comandă vocală Kephera III

```
System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\3Inainte
Rapid.exe")
    Case "journey"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\Traseu.exe")
    Case "Journey"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\Traseu.exe")
    Case "stop"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\0Stop.exe")
    Case "Stop"

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\0Stop.exe")
    End Select

    lblStatus.Text = "Am terminat dictarea!"
    btnStop.Enabled = True
End Sub

Private Sub Button1_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button1.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\1Inainte
Optim.exe")
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button2.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\4Inapoi
Optim.exe")
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button3.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\7Circular
Optim Incet.exe")
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button4.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\5Stanga
Optim.exe")
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button5.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\6Dreapta
Optim.exe")
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles Button6.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\9Circular
Extins.exe")
```

George POPESCU – Comandă vocală Kephera III

```
End Sub

Private Sub Button7_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button7.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\2Inainte
incet.exe")
End Sub

Private Sub Button8_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button8.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\3Inainte
Rapid.exe")
End Sub

Private Sub Button9_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button9.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\Traseu.exe")
End Sub

Private Sub Button10_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Button10.Click

System.Diagnostics.Process.Start("C:\Users\George\Desktop\Comenzi\0Stop.exe")
End Sub

End Class
```

5 BIBLIOGRAFIE

1. Baker J.K., „*The Dragon system - an overview*”, in *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 23, no. 1, Feb. 2005, pp. 5–29.
2. Bohus B., Rudnicky A., „*RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda*”, Eurospeech07, 2007.
3. Bonafonte A., Moreno A., Adell J., D. Aguero P., Banos E., Daniel Erro, Ignasi Esquerra, Javier Perez, Tatyana Polyakova, „*The UPC TTS System Description for the 2008 Blizzard Challenge*”, Universitat Politecnica de Catalunya, Spain, 2008.
4. Castillo O., Melin P., „*A New Approach for Voice Monitoring using Type-2 Fuzzy Logic and Fractal Theory*”, *International Journal of General Systems*, Taylor and Francis, Vol. 33, 2004, pp. 305-319.
5. Che Fai Yeong, Amin, S.H.M., Faisal, N., Bakar, J.A., „*Bluetooth enabled mobile robot*”, IEEE International Conference, 2007.
6. Choo, S. H., Shamsudin H. M., Amin N., Faisal, C. F., Yeong, „*Bluetooth transceivers for full duplex communication in mobile robots*”, Universiti Teknologi Malaysia, 2007.
7. Deshmukh N., Ganapathiraju A., Hamaker J., Picone J., si Ordowski M., „*A public domain speech-to-text system*”, in *Proceedings of the 6th European Conference on Speech Communication and Technology*, vol. 5, Budapest, Hungary, Sept. 2006, pp. 2127–2130.
8. Espinosa Horacio, Macedonia David, Same Adam, Storch David, Wroblewski Norbert, „*Robotic Control with Bluetooth Wireless Communication*”, Northwestern University, 2007.
9. S. Furui, „*Cepstral analysis technique for automatic speaker verification*”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(2), 2007, pp. 254-272.
10. Glass J.R., „*A probabilistic framework for segment-based speech recognition*”, *Computer Speech and Language*, vol. 17, no. 2, pp. 37–152, Apr. 2007.
11. Grange S., „*Vision-based sensor fusion for active interfaces*”, Diploma report, Microengineering Department, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2005.
12. Hoptroff, R., „*Bluetooth remote robot control*”, Elektor Electronics, 2008.
13. Hrnčár M., „*Aspects of voice-command creation in Human-Machine Interface*”, Dissertation project, Žilina, 2007.
14. Higgins A.L., Bahler L., Porter J., „*Speaker verification using randomized phrase prompting*”, *Digital Signal Processing*, Vol I, 2005, pag. 89-112.
15. Huang X., Acero A., Hon H., „*Spoken Language processing*”, Prentice Hall PTR, New Jersey, 2005.
16. Jaquier Cyril, Drapel Kevin, Ijspeert Auke, Crespi Alessandro, Andres Upegui, „*Using Bluetooth to Control a YaMoR Modular Robot*”, 2008.
17. Jianhua T., Jian Yu, Lixing Huang, Fangzhou Liu, Huibin Jia, Meng Zhang, „*The WISTON Text to Speech System for Blizzard 2008*”, Institute of Automation, Chinese Academy of Sciences, China, 2008.
18. Karnik N. N., Mendel J. M., „*O introducere in Sisteme Logica de tip Fuzzy 2*”, Technical Report, University of Southern California, 2007.
19. Kwok L., „*A technique for the integration of multiple parallel feature streams in the Sistem-4 speech recognition system*”, Master's Thesis (Sun Labs TR-2003-0341), Harvard University, Cambridge, MA, June 2005.

20. Lamere P., Kwok P., Walker W., Gouvea E., Singh R., Raj B., si P. Wolf, „*Design of the CMU Sistem-4 decoder*”, in *Proceedings of the 8th European Conference on Speech Communication and Technology*, Geneve, Switzerland, Sept. 2003, pp. 981–1184.
21. Langner, B., Black, A. „*Creating a Database of Speech In Noise For Unit Selection Synthesis*”, 5th ISCA Speech Synthesis Workshop, Pittsburgh, USA, 2007.
22. Langner, B., Black, A. „*An Examination of Speech In Noise and its Effect on Understandability for Natural and Synthetic Speech*”, Carnegie Mellon University, Language Technologies Institute, Technical Report CMU-LTI-04-187, 2007.
23. Langner, B., Black, A. „*Improving the Understandability of Speech Synthesis by Modeling Speech In Noise*”, ICASSP 2005, Philadelphia, USA, 2008
24. Langner, B., Black, A. „*Using Speech in Noise to Improve Understandability for Elderly Listeners*”, ASRU 2005, San Juan, Puerto Rico, 2008.
25. Lee K.F., Hon H. W., si Reddy R., „*An overview of the SISTEM speech recognition system*”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 1, pp. 35–45, Ian. 2005.
26. Li X. X., Zhao Y., Pi X., Liang L.H., si V. Nefian A., „*Audio-visual continuous speech recognition using a coupled hidden Markov model*” in *Proceedings of the 7th International Conference on Spoken Language Processing*, Denver, CO, Sept. 2008, pp. 213–216.
27. Matsui T., Furui S., „*Similarity normalization method for speaker verification based on a posteriori probability*”, *Proceedings of the ESCA Workshop on Automatic Speaker Recognition, Identification and Verification*, 2004, p. 59-62.
28. Mayer H., Schilling K. Mayer H. and Harmo P. Halme A, „*Mobile robot control via Bluetooth technology*”, Zurcher Hochschule Winterthur, IEEE, 2008.
29. Melin P., Urias J., Solano D., Lopez M., Castillo O., „*Recunoastere vocala cu retele neuronale, Logica Fuzzy de tip 2 si algoritmi genetici*”, August 2006, Engineering Letters.
30. Melin P., Gonzalez F., Martinez G., „*Pattern Recognition Using Modular Neural Networks and Genetic Algorithms*”, *Proceedings of IC-AI'04*, Las Vegas, USA, 2004, pp. 77-83.
31. Placeway P., Chen S., Eskenazi M., Jain U., Parikh V., Raj B., Ravishankar M., Rosenfeld R., Seymore K., Siegler M., Stern R., si Thayer E., „*The 1996 HUB-4 Sistem-3 system*”, in *Proceedings of the DARPA Speech Recognition Workshop*. Chantilly, VA: DARPA, Feb. 2007.
32. Ravishankar M.K., „*Efficient algorithms for speech recognition*”, PhD Thesis (CMU Technical Report CS-96-143), Carnegie Mellon University, Pittsburgh, PA, 2006.
33. Raux, A., „*Automated Lexical Adaptation and Speaker Clustering based on Pronunciation Habits for Non-Native Speech Recognition*”, *INTERSPEECH (ICSLP) 2008*, Jeju Island, Korea.
34. Seneff S., Hurley E., Lau R., Pao C., Schmid P., Zue P., „*Galaxy-II: A reference architecture for conversational system development*”, *ICSLP98*, Sydney, Australia, 2008.
35. Sidenmark, M., „*Blue Id Access System using Bluetooth*”, University of Karlskrona, 2006.
36. Singh R., Warmuth M., Raj B., si Lamere P., „*Classification of speech applications*”, in *Proceedings of the 8th European Conference on Speech Communication and Technology*, Geneve, Switzerland, Sept. 2005, pp. 1073–1776.
37. Terrence W. Fong, Chuck Thorpe, Betty Glass, „*PDA driver – A handheld system for remote driving*”, *IEEE International Conference on Advanced Robotics*, July 2006.
38. Terrence W. Fong, Chuck Thorpe, C. Baur, „*Advanced interfaces for vehicle teleoperation: collaborative control, sensor fusion displays and web-based tools*”, *Autonomous Robots*, July 2006.
39. Yamagishi J., Zen H., Wu Y-Y., Toda T., Tokuda K., „*The HTS-2008 System: Yet Another Evaluation of the Speaker-Adaptive HMM-based Speech Synthesis System in The 2008 Blizzard Challenge*”, HTS working group: Nagoya Institute of Technology, Japan / Nara Institute of Science and Technology, Japan / University of Edinburgh, UK, 2008.

George POPESCU – Comandă vocală Kephera III

40. Yapanel H. Umit, Hansen H.L. John, „*Towards an intelligent acoustic front end for automatic speech recognition: built-in speaker normalization*”, Center for Robust Speech Systems, Department of Electrical Engineering, University of Texas at Dallas, USA, May 2008.
41. Young S., „*The HTK hidden Markov model toolkit: Design and philosophy*” Cambridge University Engineering Department, UK, Tech. Rep. CUED/FINFENG/TR152, Sept. 2004.
42. Young S., Russell N.H., si Russell J.H.S., „*Token passing: A simple conceptual model for connected speech recognition systems*”, Cambridge University Engineering Dept, UK, Tech. Rep. CUED/F-INFENG/TR38, 2007.
43. Zijing Lin, Meng, M.Q.-H., Wanming Chen, Huawei Liang, Xin Liu, „*Design of a PDA-based telerobotic system*”, ROBIO, IEEE International Conference on Robotics and Biomimetics, 2007.
44. X-Voice: <http://xvoice.sourceforge.net/>
45. G-Voice: <http://www.cse.ogi.edu/~omega/gnome/gvoice/>
46. ISIP: <http://www.isip.msstate.edu/project/speech/>
47. IBM viaVoice: http://www-4.ibm.com/software/speech/dev/sdk_linux.html
48. Vocalis: <http://www.vocalisspeechware.com/>
49. PDAriver: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.1234>
50. *Mobile Autonomous Robot Software (MARS)*, Self-Composing Adaptive Programming Environment, Final Report, SAIC, Littleton, CO, 2006.
51. *Kephera III User Manual*, version 2.2, K-Team, Switzerland, March 2008, info@k-team.com, www.k-team.com.