UNIVERSITETI POLITEKNIK
FAKULTETI I INXHINIERISË ELEKTRIKE
DEPARTAMENTI I AUTOMATIKËS

DISERTACION

PËR MBROJTJEN E GRADËS SHKENCORE

DOKTOR

"*Metoda programimi dhe aplikime për sistemet e kompjuterizuara me ndjekje automatike të objekteve si dhe me komandim në distancë*"

Përgatitur nga
MSc. Ing Genti PROGRI

Udhëheqës shkencor
Prof. Dr. Petrika MARANGO

Tiranë, 2016

FALENDERIM

Së pari dëshiroj të falenderoj autoritetet e Departamentit të Automatizimit të Fakultetit të Inxhinerisë Elektrike të Universitetit Politeknik që mundësuan realizimin e suksesshëm të këtij punimi të doktoraturës.

Para së gjithash, dua të shpreh mirënjohjen time të thellë dhe falenderimin për udhëheqësin tim Prof. Dr. Petrika Maranga për bashkëpunimin miqësor e shkencor, për udhëheqjen, mbështetjen, këshillat dhe sugjerimet konstruktive gjatë gjithë kohës.

Falenderim të veçantë i shpreh mikut tim Prof. Dr. Genci Capi për disa ide profesionale që ka ofruar.

Falënderoj gjithashtu të gjithë ata që ndihmuan që ky punim të përfundojë me sukses.

Së fundi, falënderoj familjen time për mbështetjen e vazhdueshme e të pakursyer, inkurajimin, durimin, përkrahjen morale e materiale dhe për mirëkuptim gjatë gjithë kohës së kryerjes së punimit.

Disertacioni

paraqitur nga: MSc. Genti PROGRI

për marrjen e gradës shkencore

DOKTOR

TEMA

"***Metoda programimi dhe aplikime për sistemet e kompjuterizuara me ndjekje automatike të objekteve si dhe me komandim në distancë***"

Udhëheqësi shkencor: Prof. Dr. Petrika MARANGO

Mbrohet me datën ____. ____. _____ para jurisë:

1. _____ Kryetar

2. _____ Anëtar (Oponent)

3. _____ Anëtar (Oponent)

4. _____ Anëtar

5. _____ Anëtar

PERMBAJTJA

Indeksi i figurave

Lista e shkurtimeve

I/O	>	Input - Output
LPT	>	Line Printer Terminal
COM	>	Communication Port
USB	>	Universal Serial Bus
NN	>	Neural Network
GA	>	Genetic Algorithm
GUI	>	Graphical user interface
RCM	>	Robot Command Module
DE	>	Depth Estimation
RNA	>	Rrjet nervor artificial
BAM	>	Bluetooth Adapter Module

HYRJE

1.1 Formulimi i objektit.

Realizimi i komandimit prej kompjuterit, të paisjeve me elemente motorike me hapa si dhe e paisjeve robotike, për kryerjen e proceseve të caktuara është një detyrë e rëndësishme. Sistemet komanduese kompjuterike (hardware & software) së bashku me paisjet e tjera lëvizëse si dhe me paisjet sensorë I/O, përbëjnë ato që ne i quajmë paisje robotike. Në atë që përkufizuam si një paisje robotike, sistemet komanduese kompjuterike zënë një vend të rëndësishëm sepse përbëjnë atë që do ta quanim 'trurin' e sistemit robotik. Një sistem komandues kompjuterik do ta përfytyronim të përbërë nga dy pjesë të përshkruara në vijim, të lidhura ngushtë me njëra- tjetrën.

- Pjesa fizike që përfshin kompjuterin, mikrokontrollerin dhe sensorët (I/O). Kjo njihet si pjesa hardware.
- Pjesa e programeve që përpunojnë vlerat e marra nga sensorët (input) si dhe realizojnë komandimet e motorëve. Kjo njihet si pjesa software.

Sistemi komandues kompjuterik emërtohet dhe rregullator i sistemit të kontrollit në terminologjinë e kontrollit automatik të proceseve.
Për sa i takon pjesës hardware dhe pjesës software ato janë të lidhura ngushtësisht me njëra- tjetrën dhe funksionaliteti është rrjedhojë e sinkronizimit të tyre.

1.2 Objektivat përfundimtare që kërkohen të arrihen

Qëllimi kryesor i këtij punimi është të tregojë komandimin e robotit prej kompjuterit, me ndihmën e kamerave, në kontrollin e distancës prej një objekti në lëvizje. Për realizimin e kësaj detyre duhet të zhvillohen disa hapa që janë :

- Mënyrat e komandimit të motorëve me hapa prej kompjuterit, në funksion të portës I/O ku lidhet indirekt paisja motorike, si dhe metoda programimi në mjedise të ndryshme programimi.
- Lokalizim i objekteve që vëzhgohen në imazhet e marra prej kamerave.
- Ndjekja e objekteve në fushën e imazhit të kamerave.
- Përcaktimi i distancës së objekteve prej planit të kamerave.
- Komandim i sistemit robotik në ndjekje të objektit në lëvizje.

Nisur nga qëllimi kryesor i përshkruar më sipër, objektivat që kërkohen të arrihen në këtë punim përshkruhen në disa hapa në vijim :

- Realizimi i komandimit të lëvizjes dhe shpejtësisë së motorit me hapa duke shfrytëzuar portat I/O të kompjuterit, të emërtuara LPT, COM, USB. Gjithashtu do të tregohet realizimi i komandimit nga kompjuteri në distancë pa tel (wireless) të motorit me hapa.

- Realizimi teorik dhe praktik i pozicionimit, ndjekjes dhe vlerësimit të distancës së një objekti të shënjuar prej një sistemi kontrolli të përbërë prej një kamere dhe lazerit pikësor, dy kamerash dhe prej një kamere referuar vlerësimit të thellësisë (perspektivës).

- Realizimi i kontrollit të distancës së robotit prej një objekti në lëvizje me ndihmën e kameras duke shfrytëzuar metoda klasike të ndjekjes përcaktuar nga shpejtësia e lëvizjes së objektit.

- Realizimi i kontrollit të distancës së robotit prej një objekti në lëvizje me ndihmën e kameras duke shfrytëzuar rrjetat neurale (NN) të optimizuara me algoritmat gjenetike (GA).

1.3 Struktura e disertacionit

Komandim i motorit me hapa. Motorët me hapa. Parametrat e motorave me hapa EM259/2019033-1 dhe 28BYJ-48 5VDC. Komandimi i motorave me hapa prej kompjuterit nëpërmjet portës LPT. Porta paralele LPT në PC. Skema elektronike. Programet e realizuara në Visual Basic, Matlab për komandimin prej portës LPT	Jan-13 Feb-13
Komandimi i motorave me hapa prej kompjuterit nëpërmjet portës USB.Porta USB në PC. Skema elektronike Programet e realizuara në Visual Basic për komandimin prej portës USB.	May-13 Jun-13
Komandimi nëpërmjet PIC18F4550-I/P Microchip Microcontroller & ULN2803	Jul-13
Komandimi i motorit me hapa në distancë me valë nga kompjuteri	Aug-13
Programet e realizuara në C++ dhe Visual Studio 6 për komandimin në distancë, me valë	Sep-13
Disa metoda optike të vlerësimit të distancës së objekteve prej sistemeve të kontrollit. Metoda e vlerësimit të distancës së objekteve nëpërmjet një sistemi i përbërë prej një kamere dhe lazerit pikësor.. Algoritmika dhe programet e realizuara Visual Studio 6 (VB).	Oct-13
Metoda e vlerësimit të distancës së objekteve nëpërmjet një sistemi prej dy kamerash.. Algoritmika dhe programet e realizuara Visual Studio 6 (VB) dhe Matlab R2013b Metoda e vlerësimit të distancës së objekteve prej një kamere referuar vlerësimit të thellësisë (Depth Estimation).	Nov-13
Algoritmika dhe programet e realizuara në Matlab R2013b	Dec-13
Sisteme të thjeshta robotike të komanduara nga kompjuteri në ndjekjen sipas një drejtimi të një objekti të paracaktuar.	Jan-14 Feb-14 Mar-14
Sisteme moderne robotësh të programueshëm. Roboti model iRobot Create.	Feb-14 Mar-14
Specifikime dhe libraria e komandave në mjedisin Matlab.	Apr-14 May-14 Jun-14
Specifikime dhe libraria e komandave në Visual Studio .Net (VB). Komunikimet me kabëll (wire) dhe me valë (wireless and bluetooth)..	May-14 Jun-14 Jul-14
Specifikime dhe libraria e komandave në Visual Studio 6 (VB). Komunikimet me kabëll (wire) dhe me valë (wireless and bluetooth)..	Aug-14 Sep-14 Oct-14
Specifikime dhe libraria e komandave në Basic4Androide (Java). Komunikimet me kabëll (wire) dhe me valë (wireless and bluetooth)..	Oct-14 Nov-14
Përdorimi i sistemit robotik iRobot Create në fusha të ndryshme sociale.	Dec-14
Ndjekja e një objekti prej iRobot Create sipas metodave klasike të ndjekjes përcaktuar nga shpejtësia e lëvizjes së objektit.	Jan-15
Teoria e algoritmatve neuronikë (ANN)..	Feb-15 Mar-15
Teoria e algoritmatve gjenetikë (GA)..	Mar-15 Apr-15
Përdorimi i algoritmave neuronikë (ANN) në realizimin e orientimit dhe kontrollit të distancës së iRobot Create prej një objekti në lëvizje...	May-15 Jun-15
Përdorimi i algoritmave gjenetikë (GA) në realizimin e orientimit dhe kontrollit të distancës së iRobot Create prej një objekti në lëvizje. Simulimi i kontrollit të distancës së iRobot Create prej një objekti qëllim në Matlab R2013b.	Jun-15 Jul-15 Aug-15
Realizimi ekperimental i kontrollit të distancës së iRobot Create prej një objekti qëllim në Matlab R2013b	Jul-15 Aug-15 Sep-15 Oct-15 Nov-15
Programet e simulimeve dhe ato në kohë reale Matlab R2013b	Sep- Oct- Nov- Dec-

	15 15 15 15
Prezantimi i posterave	Dec-15
Shkrimi i materialeve dhe sistemii i tyre	Aug-13 Dec-15

KAPITULLI II

Hyrje

Në këtë kapitull do të përshkruhen disa mënyra komandimi të drejpërdrejta të një dhe dy motorave me hapa, prej kompjuterit nëpërmjet portave paralele LPT, portave seriale COM, portave USB dhe me valë.
Realizimi i këtyre komandimeve bëhet me ndihmën e qarqeve te integruar ULN2003A, ULN2803A të cilët ndërmjetësojnë lidhjen me motorin me hapa që do të komandojmë.
Gjithashtu do të përshkruhet mënyra e komandimi të shpejtësisë së lëvizjes dhe e pozicionit të motorit me hapa, prej kompjuterit. Kjo pjesë në thelb përmbledh komandimin bazë të një motori me hapa, i përdoruar gjerësisht në robotë, si elementi bazë i robotëve.

Në pjesën e fundit të kapitullit jepen algoritmikat e përdorura dhe programet e realizuara në gjuhët e programimit Visual Studio .Net (VB) [5], Visual Studio 6 (VB) [4], Visual Studio C# [6], Matlab R2013b [14] për komandimin e motorit me hapa. Programet do të sigurojnë të gjitha lëvizjet e mundëshme të motorit me hapa (psh gjysëm hap para, gjysëm hap prapa, hap para, hap prapa, rrotullim me një kënd të përcaktuar etj.)

2.1 Komandimi i motorit me hapa.

Në praktikë problemet inxhinerike janë të vështira të zgjidhen gjithmonë me metodat klasike. Procese të ndryshme kontrollohen nëpërmejt kompjuterave. Kjo bëhet për disa arsye, ku ndër më kryesoret mund të përmendim :

- Një kontroll i tillë siguron drejtimin e proceseve industriale me një algoritëm të parametrizuar, gjë që siguron hapësirë në proceset e kontrollit.
- Me të njëjtin sistem kontrolli të komjuterizuar mund të sigurosh kontroll për disa procese industriale të kategorizuara në një drejtim (programe të ndryshme për procese të ndryshme).
- Një kontroll i tillë siguron drejtimin e proceseve industriale me përpikmëri të programuar duke rritur shkallën e saktësisë.

Në këtë kuadër për të ndërtuar këto sisteme të ndërlikuara kontrolli, nevojitet analiza e kontrolli të elementeve përbërës të tyre. Kjo është arsyeja që në këtë pjesë po japim një mënyrë të realizimit të kontrollit të motorit me hapa prej kompjuterit. Për të realizua këtë qëllim të parë është e nevojshme të japim skemën bazë të kontrollit prej kompjuterit të motori me hapa Fig 2.1 nëpërmjet një qarku të integruar që do të përbëjë modulin e kontrollit (device control).

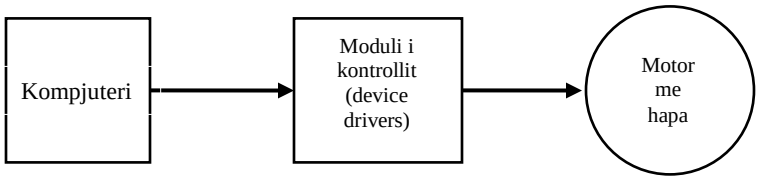


Fig 2.1. *Skema bazë për kontrollit e motorit me hapa*

Komunikimi me motorin do të realizohet nëpërmjet modulit të kontrollit i cili lidhet me kompjuterin nëpërmjet portës LPT ose USB.

Elementët që do të na nevojiten për kryerjen e detyrës së shtruar janë paraqitur më poshtë :

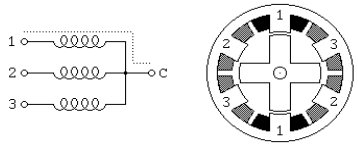
- **Hardware (pjesët fizike)**
  a. Motor me hapa njëpolar me 5 përcjellësa (5-wire Unipolar stepper motor)
  b. Qarku i integruar ULN 2003 A që do jetë qarku komandues për motorin me hapa.
  c. Fishë mashkull për portën LPT model DB-25 pin.
  d. Stripboard
  e. PC me portë paralele LPT

- **Software (programe)**
  a. Visual Basic 6
  b. Visual Basic Express Editions 2010 (.Net)
  c. Visual C#

2.1.1 Motori me hapa

Në teori një motor me hapa është një paisje shumë e thjeshtë. Një motor me hapa në fakt është një paisje elektromekanike e cila konverton sinjalet pulsante elektrike në lëvizje diskrete mekanike rrotulluese. Lëvizja rrotulluese e rotorit të motorit me hapa është e përcaktuar nga sekuencat e sinjave elektrike që zbatohen në motor [1]. Sekuencat e impulseve të zbatuara do të përcaktojnë dhe mënyrën e rrotullimit të rotorit të motorit. Shpejtësia e zbatimit të tyre do të përcaktojë dhe shpejtësinë e rrotullimit të motorit, brenda disa kufijve që lidhen me parametrat fizikë të paisjes. Në formë simbolike mund të themi se këta motorë rrotullohen vetëm me një hap në çdo çast kohe. Nisur nga kjo veti, motorët me hapa kanë një përdorim shumë të madh, duke u përdorur në të gjitha lëvizjet ku kërkohet kontroll të pozicionit pra kur kërkohen rrotullime me kënde të përcaktuara [3]. Dallohen disa tipe motorësh me hapa. Shkutimisht disa prej modeleve të type po i paraqesim skematikisht më poshtë [3].

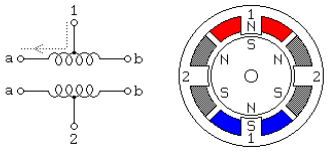
a. Variable Reluctance Motor



Motori i mësipërm ka tre pështjella me fundet e tyre të lidhura së bashku. Pozicioni i tyre është me sfazim 30^0 siç tregohet në figurën e mësipërme. Dhënia e tensionit në të tre pështjellat në mënyrë sekuenciale do të siguroje rrotullimin e rotorit të motorit në mënyrë të vazhdueshme.

Pështjella 1 : 10010010010010010010010010
Pështjella 2 : 01001001001001001001001001
Pështjella 3 : 00100100100100100100100100

b. Unipolar Motor



Në këtë tip motori prej secilës pështjellë është nxjerrë prej meseve të tyre daljet 1 dhe 2 siç tregohen në figurën e mësipërme. Pikat 1 dhe 2 do të jenë pikat me potencial pozitiv që do të lidhen me burimin e ushqimit DC. Për të siguruar fushën magnetike në pështjella është e nevojshme që dy skajet e sejcilës prej dy përshtjellave të tokëzohen. Ky motor rrotullohet 30^0 për çdo hap. Dy sekuenca kontrolli për motorin e mësipërm jepen si më poshtë :

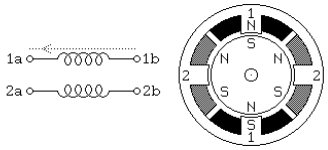
Bobina 1a : 10001000100010001000100010001
Bobina 1b : 00100010001000100010001000100
Bobina 2a : 01000100010001000100010001000
Bobina 2b : 00010001000100010001000100010
time --->

Bobina 1a : 11001100110011001100110011001
Bobina 1b : 00110011001100110011001100110
Bobina 2a : 01100110011001100110011001100
Bobina 2b : 10011001100110011001100110011
time --->

Kombinimi i dy sekuencave do të sigurojë rrotullimin e motorit me gjysëm hapi.

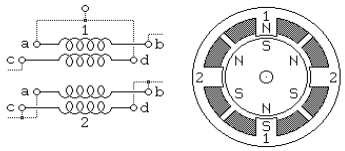
Bobina 1a : 11000001110000011100000111
Bobina 1b : 00011100000111000001110000
Bobina 2a : 01110000011100000111000001
Bobina 2b : 00000111000001110000011100

c. Bipolar Motor



Motori në figurën e mësipërme ka të njëjtin ndërtim si dhe motori Unipolar me dy pështjella por pa dalje prej mesit të tyre. Qarku elektrik për ushqimin e pështjellave të motorit është më i ndërlikuar, duke përdorur një qark kontrolli H-Bridge për çdo përshtjellë. Në vazhdim sekuencat me shenjat + dhe − tregojnë polaritetin e tensionit të zbatuar në pështjella prej qarkut të kontrollit H-Bridge.

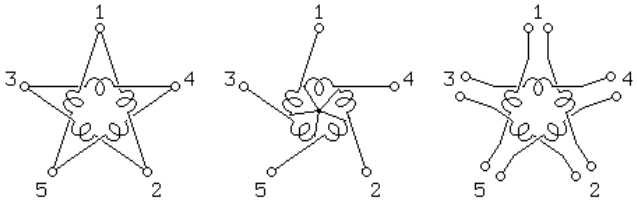
d. Bifilar Motor



Motori në figurën e mësipërme ka të njëjtin ndërtim si dhe motori unipolar. Tek ky motor pështjellat janë te lidhura në paralel siç tregohet në figurë. Në këtë rast do të kemi tetë

përcjellësa. Dy përcjellësa të çdo bobine janë të lidhur në seri dhe pika e lidhjes është përdorur si dalje duke përdorur motorin bifilar si një unipolar, ndërsa për ta përdorur motorin bifilar si një motor bipolar, telat e secilës bobinë do të lidhen në seri ose paralel.

e. Multiphase Motor



Pështjellat e këtij motori janë të lidhura siç tregohet në figurën e mësipërme. Ato janë të lidhura dy e nga dy në seri sipas skemave trekëndësh, yll ose në yll të hapur në skaje. Tensioni që zbatohet në pështjella jepet nëpërmjet qarkut të kontrollit H-Bridge.

Në Fig 2.2 jepet skema principiale e ndërtimit të motorit me hapa me katër pole ose dhëmbë në stator dhe dy pole në rrotor. Rrotori dhe statori janë ndërtuar prej materiali çelik. Statori ka dy çifte pështjellash (bobinash) të lidhura në seri siç tregohet në figurën e mëposhtëme [7].

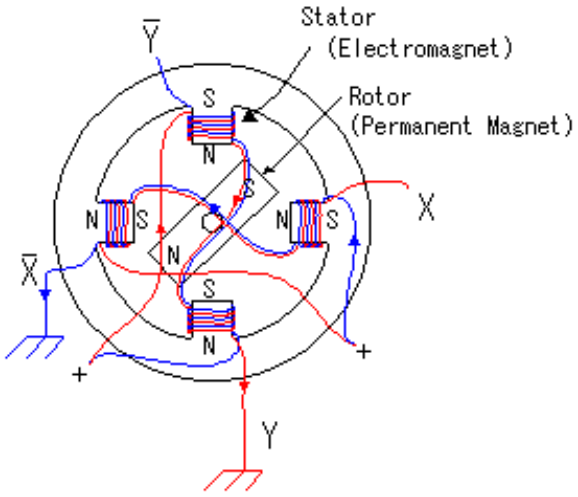


Fig 2.2. *Skema principiale e motorit me hapa*.

Secili çift pështjellash përbën një fazë të motorit me hapa. Parimi i thjeshtuar i punës i këtyre motorave shpjegohet në Fig 2.3.

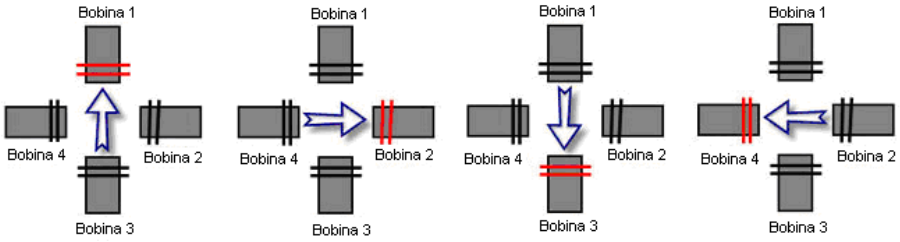


Fig 2.3. *Rotullimi i magnetit të përhershëm (rotorit) me hap të plotë.*

Në Fig 2.3-1 rotori ka orientimin e treguar si rezultat i fushës magnetike të krijuar kur në bobinën 1 kalon rrymë e vazhduar. Pas ndërprerjes së impulsit në bobinën 1 dhe lëshimit në bobinën 2, rotori do të marrë pozicionin e ri të treguar në Fig 2.3-2. Më tej rotori kalon në pozicionin drejtuar bobinës 3 treguar në Fig 2.3-3 dhe më tej bobinës 4 treguar në Fig 2.3-4.
Në tabelën e mëposhtme jepet shpërndarja e inpulseve në bobina për të siguruar rrotullimin me hapa.

Impulsi i zbatuar	Bobina 1	Bobina 2	Bobina 3	Bobina 4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Një motor i shpjeguar si më sipër do të funksiononte me katër hapa. Për të rritur saktësinë e pozicionit një motor i tillë do të mund të funksiononte edhe me gjysëm hapi. Në këto kushte një motor që funksionon me 4 hapa ose 90 gradë për hap, në rastin e rrotullimit me gjysëm hapi do të realizonte një rrotullim të plotë pas 8 hapash ose 45 gradë për hap. Për të siguruar rrotullimin me gjysëm hapi, tensionet nëpër bobina në momente të caktuara të kohës do të jepeshin siç tregohet në Fig 2.4.

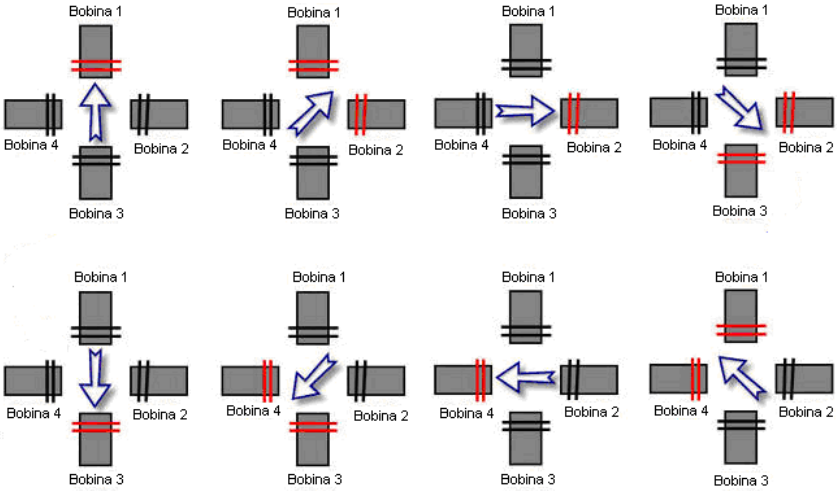


Fig 2.4. *Rotullimi i magnetit të përhershëm (rotorit) me ½ hapi.*

Në tabelën e mëposhtme jepet shpërndarja e impulseve në bobina për të siguruar rrotullimin me ½ hap.

Impulsi i zbatuar	Bobina 1	Bobina 2	Bobina 3	Bobina 4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

Përveç rrotullimeve me gjysëm hapi dhe me hap të plotë, më këtë motor mund të realizohet edhe një rrotullim hap i plotë me pozicionim midis dy bobinave. Kjo lëvizje realizohet atëhere kur tensioni zbatohet njëherësh në dy bobina të njëpasnjëshme si në Fig 2.5.

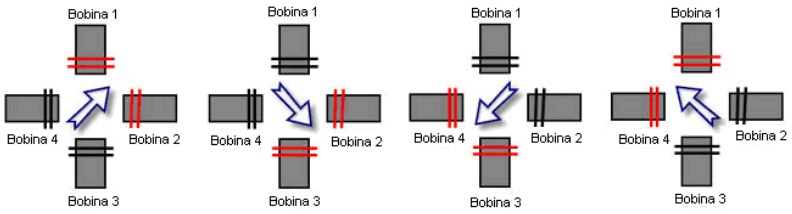


Fig 2.5. *Rotullimi i magnetit të përhershëm (rotorit) me hap të plotë me pozicionim midis dy bobinave.*

Në tabelën e mëposhtme jepet shpërndarja e impulseve në bobina për të siguruar rrotullimin me hap të plotë me pozicionim midis dy boboinave.

Impulsi i zbatuar	Bobina 1	Bobina 2	Bobina 3	Bobina 4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Në mënyrë të përmbledhur llogaritja e këndit të rrotullimit të rrotorit te motorit do të bëhej në këtë mënyrë :

$$\theta_{rrot} = \frac{360^0}{S}$$

$$S = m \cdot N_r \qquad (2.1)$$

$ku: m =$ numri i fazave

$N_r =$ numri i çift dhëmbëve të rrotorit

Për motorin e marrë në trajtimin teorik llogaritja e këndit të rotullimit do të bëhej si në vijim :

$$m = 2$$
$$N_r = 2 \qquad (2.2)$$
$$S = m \cdot N_r$$
$$\theta_{rrot} = \frac{360^0}{S} = 90^0 \text{ për hap}$$

Përgjithësisht këndi i rrotullimit në motorët me hapa të sotëm është 5^0.

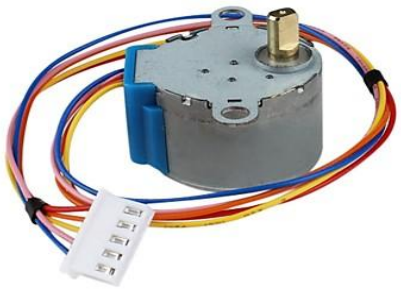
2.1.2. Parametrat e motorave me hapa EM259/2019033-1 dhe 28BYJ-48 5VDC.

Motori që ne do të përdorim në trajtimet e mëposhtëme do të jetë një motor Uniolar [1].
Për kryerjen e provave u përdor një motor printeri Lx300+ (bipolar motor) i cili është motor me 4 përcjellësa dalës (dy bobina të pavarura) dhe motori me hapa model 28BYJ-48 5VDC [15]. Pas modifikimit të motorit të printerit Lx300+, duke nxjerrë pikat e mesit të secilës prej bobinave u sigurua motori me hapa i cili skematikisht është përdorur në skemat elektronike të dhëna në vijim.

Parametrat e motorit me hapa model EM259/EM221/EM211/2019033-1 janë :

Rated voltage : 12VDC
Number of Phase 4
Speed Variation Ratio 1/64
Stride Angle 5.625° /64
Frequency 100Hz
DC resistance 50Ω±7%(25℃)
Idle In-traction Frequency > 600Hz
Idle Out-traction Frequency > 1000Hz
In-traction Torque >34.3mN.m(120Hz)
Self-positioning Torque >34.3mN.m
Friction torque 600-1200 gf.cm
Pull in torque 300 gf.cm
Insulation grade A

Parametrat e motorit me hapa model 28BYJ-48 5VDC janë :



Rated voltage : 5VDC
Number of Phase 4
Speed Variation Ratio 1/64
Stride Angle 5.625° /64
Frequency 100Hz
DC resistance 50Ω±7%(25℃)
Idle In-traction Frequency > 600Hz
Idle Out-traction Frequency > 1000Hz
In-traction Torque >34.3mN.m(120Hz)
Self-positioning Torque >34.3mN.m
Friction torque 600-1200 gf.cm
Pull in torque 300 gf.cm
Insulation grade A

2.2. Komandimi i motorave me hapa prej kompjuterit nëpërmjet portës LPT

Realizimi i komandimit të motorit me hapa prej kompjuterit nëpërmjet portës paralele LPT jepet skematikisht nëpërmjet Fig 2.6 [16].

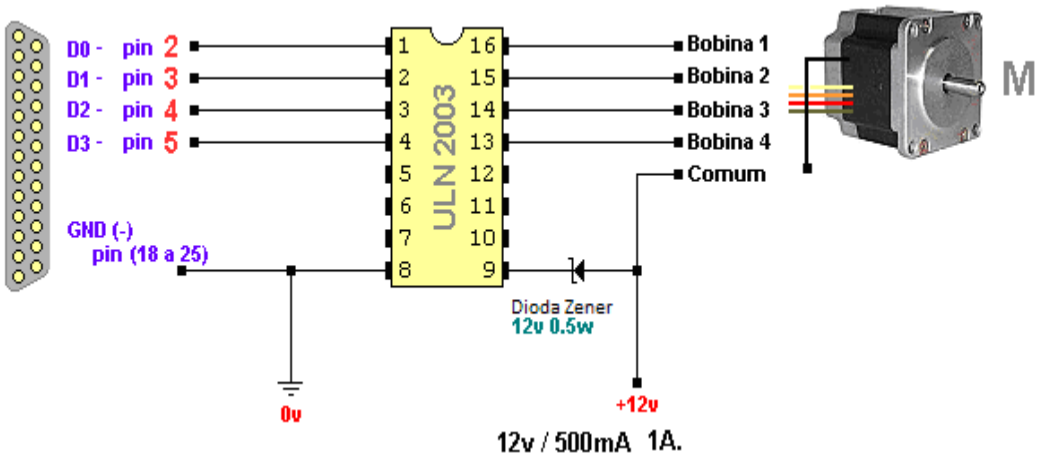


Fig 2.6. *Skema e bazë e komandimit të motorit nëpërmjet modulit të ULN2003*

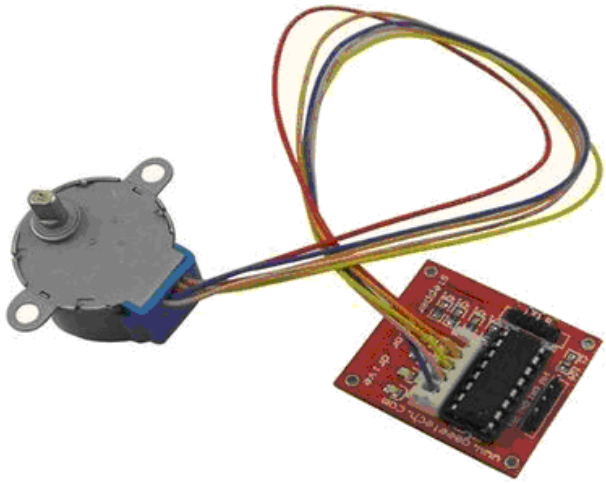


Fig 2.7. *Komandimi i* motorit *28BYJ-48 nëpërmjet modulit të ULN2003*

2.2.1. Porta paralele LPT në PC.

Një portë paralele është një socket në kompjutera që mundëson lidhjen e tyre me një numër të madh paisjesh si printera, skanera, modema dhe disa web kamera. Në kohët e sotme, kjo portë në bordet kryesore nuk po montohet duke siguruar uljen e kostos së paisjeve. Porta në fjalë ka 25 kontakte (pins) dhe qëllimi i tyre shpjegohet në Fig 2.8 [17].

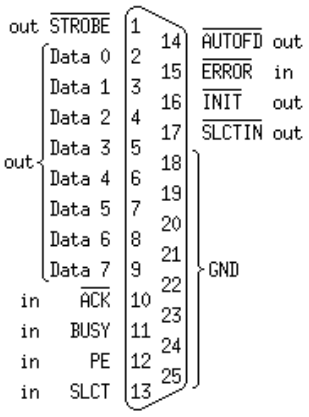


Fig 2.8. *Porta DB - 25 pin*

Shpjegimi më i hollësishëm i kontakteve të kësaj porte jepet në tabelën e mëposhtëme.

Pin-i	Emri	Përshkrimi
1	Strobe	Gjendja e këtij pin-i është me nivel (1) kur nuk kemi dërgim të dhënash në portë. Kur prej kompjuterit dërgohet 1 byte informacion në portë, gjendja e këtij pin-i kalon në (0).
2 - 9	D0 - D7	8 pin-et ku jepen datat e dërguara nga kompjuteri.
10	nAck	Dërgim sinjali për pranim informacioni nga printeri për në kompjuter.
11	Busy	Kur printeri është duke trajtuar të dhënat, gjendja e këtij pin-i është (1) . Gjendja e tij do të ndryshojë atëhere kur të dhënat do të kenë përfunduar së trajtuari.
12	Paper Out	Nëpërmjet këtij pin-i printeri i tregon kompjuterit nëse letra në printer është apo mungon.
13	Select	Nëpërmjet këtij pin-i tregohet gjendja gati ose jo gati e paisjes së lidhur në portë.
14	Autofeed	Nëpërmjet këtij pin-i kompjuteri dërgon sinjal për tërheqjen e letrës brenda printerit.
15	Error	Nqs printeri ka ndonjë problem atëhere pin-i 15 kalon prej nivelit (1) në (0) prej printerit duke siguruar që në kompjuter të identifikohet gabimi.
16	Initalize	Ky pin merr vlerën (0) sa herë që një punë e re do të shkojë në printer.
17	Select-In	Pin 17 përdoret nga kompjuteri për të shkëputur printerin prej tij nëpërmjet komunikimit.
18-25	Ground	Pin-e të tjera të pa përdorshme për rastet në trajtim.

2.2.2. Skema elektronike.

ULN2003 është një qark monolit për tensione dhe rryma të mëdha të siguruara nga skemat Darlington Fig 2.9. Ai është i ndërtuar nga shtatë çifte Darlington me emiter të përbashkët. Çdo çift

Darlington është i përbërë nga dy tranzistorë dypolarë. ULN2003 i përket familjes së ULN200X të qarqeve të integruar (ICs). ULN2003 punon me tension + 5 TTL (Transistor-Transistor Logic), nga paisja logjike CMOS. Ky qark i integruar ka një përdorim të gjerë si një rele e komanduar. Përdorimin më të rëndësishëm e ka për komandimin e motorëve me hapa.

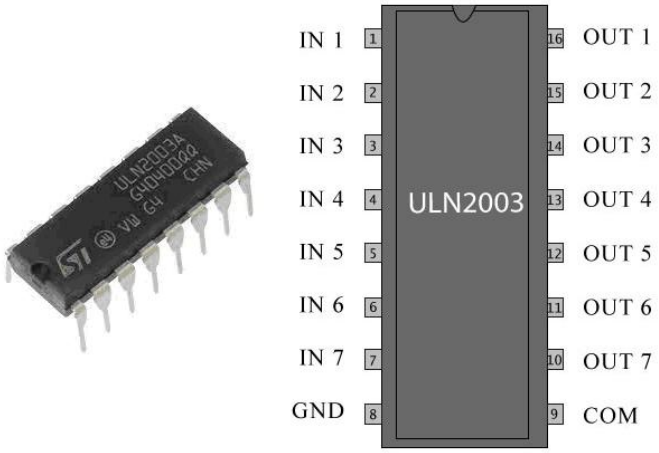


Fig 2.9. *Qarku ULN2003.*

Në Fig 2.10 jepet skema elektronike e ndërtimit të brendshëm për 1H1D të qarkut ULN2003.

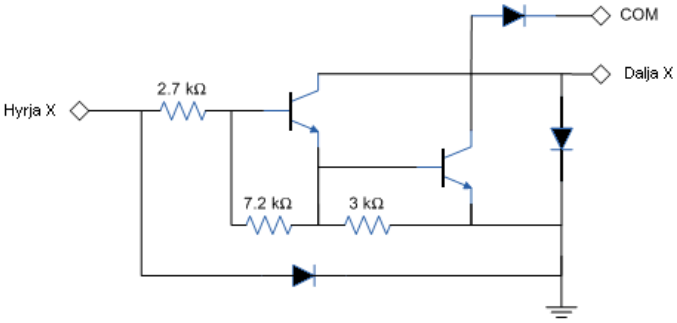


Fig 2.10. *Skema e detajuar për 1H1D në qarkun ULN2003.*

Në skemën e Fig 2.11 kompjuteri nuk është paraqitur por në vend të tij jepet vetëm porta LPT DB-25 pins. Daljet prej saj shkojnë në qarkun e integruar ULN2003A dhe më tej në motorin me hapa.

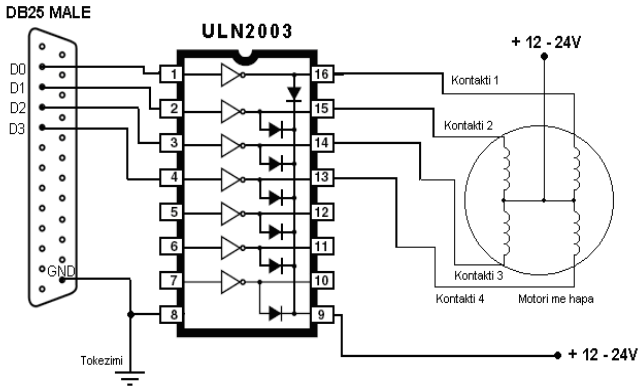


Fig 2.11. *Skema e detajuar e lidhjes së elementëve për realizimin e komandimit nga porta LPT .*

Pin-i D0 i portës së kompjuterit lidhet me pin-in (1) tek ULN2003.
Pin-i D1 i lidhet me pin-in (2) tek ULN2003.
Pin-i D2 i lidhet me pin-in (3) tek ULN2003.
Pin-i D3 i lidhet me pin-in (4) tek ULN2003.

Nga ana e motorit me hapa, skema e lidhjes realizohet nga pin-et 16, 15, 14, 13 me katër fijet e motorit duke respektuar rreshtimin e tyre. Toka e sistemit lidhet në pin-in 8. Ushqimi +12 V – +24 V jepet në pin-in 9.
Në Fig 2.12 jepet mënyra e realizimit të komandimit të dy motorëve me hapa duke shfrytëzuar të njëjtën portë LPT të kompjuterit. Kjo do të thotë se me një 8 bit data (1 byte) përcaktojmë një gjendje të të dy motorëve.

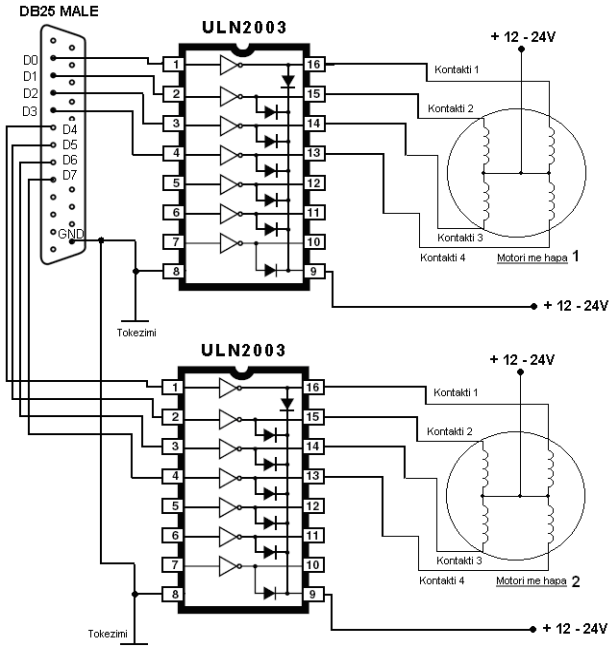


Fig 2.12. *Skema e detajuar e lidhjes së elementëve për realizimin e komandimit të 2 motorëve me të njejtën portë LPT*

2.2.3. Programet e realizuara në Visual Basic, Matlab për komandimin prej portës LPT

Përdorimi i portës paralele LPT shpjeguar ne paragrafin 2.2, është një mënyrë e thjeshtë për të realizuar komunikimet me pasisje të ndryshme. Ky komunikim ka qenë i drejtpërdrejtë në fillim (në versionet e para të sistemeve operative Windows) duke u realizuar me komandat *inportb() dhe outportb()*. Në versionet e sotme të sistemit operativ Windows, komunikimi realizohet nëpërmjet librarisë *inpout32.dll*. Shfrytëzimi i kësaj librarie u bë mënyra më e e mirë dhe e pagabueshme për të komunikuar me portën paralele LPT.

Përdorimi i *inpout32.dll* në tre gjuhët e programimit *C# , Visual Basic 6 dhe Visual Basic .Net* jepet si më poshtë.

C#:

```
private class PortAccess
{
   [DllImport("inpout32.dll", EntryPoint="Out32")]
   public static extern void Output(int address, int value);
}
```

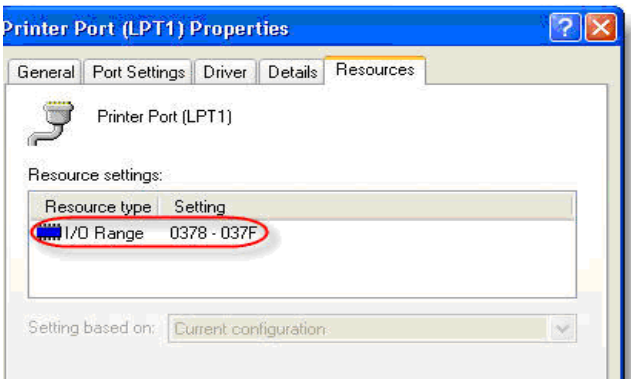
Visual Basic 6 :

```
' Inp dhe Out deklarimet per porten I/O duke perdorur inpout32.dll.

Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal portaddress As Integer) As Integer

Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal portaddress As Integer, ByVal value As Integer)
```

Visual Basic .Net:

```
Private Class PortAccess
   Public Declare Sub Output Lib "inpout32.dll" Alias "Out32" (ByVal address As Integer, ByVal value As Integer)
End Class
```

Metoda *PortAccess.Output* merr dy parametra, *adresen (address)* dhe *vleren (value)* . Njohja e adresës së portës paralele LPT bëhet në Control Panel të Windows_it. Adresa e kësaj porte përcaktohet në dritaren e vetive në Windows dhënë në vijim :



Sic shihet në figurën e mësipërme porta LPT ka adresën 0378-037F. Vlera në hexadecimal '0x378', në decimal do të jetë '888'. Nqs do të përdornim portën LPT2 me adresë në hexadecimal '0x278', vlera e saj në decimal do të qe '632' .
Forma e dërgimit të informacioneve binare apo decimale në portë do të jetë nëpërmjet thirrjes së funksioneve si më poshtë :

C#:

```
PortAccess.Output(888, 255);
```

Visual Basic 6 dhe Visual Basic.Net :

```
PortAccess.Output(888, 255)
```

Në funksionet e mësipërme në pjesën e vlerës jepet numri decimal '*255*' i cili në sistemin binary do të shprehej '*1111 1111*'.Dërgimi i '*1111 1111*' do të thotë se gjendjet e portave D0-D7, janë me +5 V (gjendja e tyre '1'). Kalimi i tyre në gjendjet '0' bëhet duke thirrur funksionet :

C#:

```
PortAccess.Output(888, 0);
```

Visual Basic 6 dhe Visual Basic.Net :

```
PortAccess.Output(888, 0)
```

Për të kryer lëvizjen e motorit me hap të plotë kodet që do të përdoreshin për këtë rast do të qenë si më poshtë :

C#:

```
PortAccess.Output(888, 1);          // 1 decimal = 0001 binary. This will set D0 to high
System.Threading.Thread.Sleep(100); // delay
PortAccess.Output(888, 2);          // 2 decimal = 0010 binary. This will set D1 to high
System.Threading.Thread.Sleep(100); // delay
PortAccess.Output(888, 4);          // 4 decimal = 0100 binary. This will set D2 to high
System.Threading.Thread.Sleep(100); // delay
PortAccess.Output(888, 8);          // 8 decimal = 1000 binary. This will set D3 to high
```

Visual Basic 6:

```
Output(888, 1)                  ' 1 decimal = 0001 binary. This will set D0 to high
Sleep(100)                      ' delay
Output(888, 2)                   ' 2 decimal = 0010 binary. This will set D1 to high
Sleep(100)                      ' delay
Output(888, 4)                  ' 4 decimal = 0100 binary. This will set D2 to high
Sleep(100)                      ' delay
```

Visual Basic.Net :

```
PortAccess.Output(888, 1)           ' 1 decimal = 0001 binary. This will set D0 to high
System.Threading.Thread.Sleep(100)  ' delay
PortAccess.Output(888, 2)           ' 2 decimal = 0010 binary. This will set D1 to high
System.Threading.Thread.Sleep(100)  ' delay
PortAccess.Output(888, 4)           ' 4 decimal = 0100 binary. This will set D2 to high
System.Threading.Thread.Sleep(100)  ' delay
PortAccess.Output(888, 8)           ' 8 decimal = 1000 binary. This will set D3 to high
```

Për të realizuar dërgimin e informacioneve binare në drejtim të bllokut të kontrollit të motorit, duhet të realizohet kalimi i këtij informacioni prej formës binare në atë decimale. Për të realizuar këtë konvertim përdoret funksioni i mëposhtëm i shkruar për Viasual Basic 6 dhe Visual Basic .Net.

```
Public Function BinToDec(BinNum As String) As String
  Dim i          As Integer
  Dim DecNum     As Long

  On Error GoTo ErrorHandler

'  Cikli binar
  For i = Len(BinNum) To 1 Step -1
    ' Kontrollo stringun per karaktere jo te rregult
    If Asc(Mid(BinNum, i, 1)) < 48 Or _
       Asc(Mid(BinNum, i, 1)) > 49 Then
       DecNum = ""
       Err.Raise 1002, "BinToDec", "Invalid Input"
    End If
     If Mid(BinNum, i, 1) And 1 Then
        DecNum = DecNum + 2 ^ (Len(BinNum) - i)
     End If
  Next i
  BinToDec = DecNum

  Exit Function

ErrorHandler:
  Msgbox "Messazh gabimi ne konvertim :" & " " & Err.Number  &  " " & Err.Desctription
End Function
```

Pamja e një programimi të thjeshtë në Visual Basic 6 për kontrollin e motorit me hapa jepet më poshtë.

Në faqen kryesore të këtij programi demostrues, paraqiten disa mundësi të këtij kontrolli kompjuterik mbi motorin me hapa si psh :

- Rrotullim me frekunca të ndryshme me hap të plotë.
- Rrotullim me frekunca të ndryshme me gjysëm hapi.
- Ndalim të menjëhershëm për cdo gjendje të tij.
- Rrotullime në kahun orar ose në atë jo orar.

Mbështetur në këto ide mund të ndërtohen programe të tjera më të sofistikuara për të realizuar detyra të caktuara të cilat përfshijnë elementet e trajuar në këtë pjesë.

Për të komunikuar me portën LPT prej programit Matlab R2013b [14], për dërgimin e komandave në motor sipas skemës elektronike të përshkruar më sipër, ekzekutohen instruksionet e mëposhtme :

```
dio = digitalio('parallel','LPT1');      % --> Krijohet një object
lines = addline(dio,0:7,0,'out');        % --> Inicializohen pinet 2-9
putvalue(lines,infordata);               % --> Dërgohet 'data' në portë
```

Instruksioni : `dio = digitalio('parallel','LPT1');` krijon një objekt për portë paralele që adresohet tek LPT1.
Instruksioni : `lines = addline(dio,0:7,0,'out');` inicializon pinet 2 deri 9 si pine dalëse, me gjendjet e sinjaleve në vlerat 0 V ose 5 V që korespondojnë me gjendjet binare 0 dhe 1.
Instruksioni : `putvalue(lines, infordata);` kryen dërgimin e paketës `infordata` në portën paralele.

Datat e dërguara `infordata` mund të dërgohen në kodin decimal ose binar si në shembullin e dhënë më poshtë :

```
putvalue(dio,8)
putvalue(dio,[0 0 0 1 0 0 0 0])
```

Kështu për të ndaluar rrotullimin e motorit instruksionet që do të përdoren jepen si më poshtë :

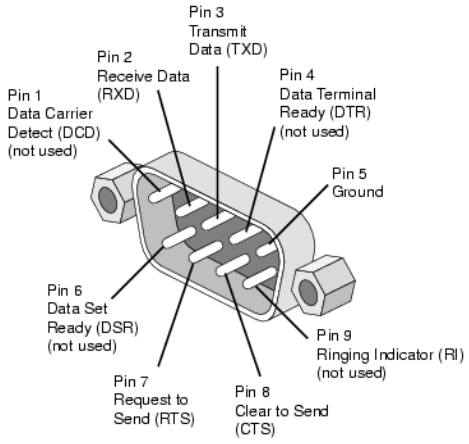
```
dio = digitalio('parallel','LPT1');
lines = addline(dio,0:7,0,'out');
putvalue(lines,0);
```

Në Shtojcën-1 po japim programet e realizuara për komandimin e shpejtesisë dhe pozicionit të motorit me hapa prej portë LPT.

2.3. Komandimi i motorave me hapa prej kompjuterit nëpërmjet portës COM

2.3.1. Porta seriale COM në PC.

Një portë seriale është një ndërfaqe fizike në kompjutera që mundëson lidhjen e tyre me një numër të madh paisjesh si printera, skanera, modema. Transferimi i datave nëpërmjet kësaj porte realizohet në hyrje dhe dalje në mënyrë seriale që do të thotë se në çdo çast kohe transmetohet një informacion prej një biti. Në kohët e sotme, kjo portë në bordet kryesore nuk po montohet duke siguruar uljen e kostos së paisjeve. Porta në fjalë ka 9 kontakte (pins) dhe qëllimi i tyre shpjegohet më poshtë.



2.3.2. Skema elektronike.

Në Fig 2.13 jepet mënyra e realizimit të komandimit të motorit me hapa me 6 përcjellësa prej portës seriale RS232 nëpërmjet modulit M2-ENGINE & ULN2803. Qarku ULN2803 siguron rryma më të mëdha në dalje nëpërmjet tranzistorëve të lidhur sipas skemës Darlington.

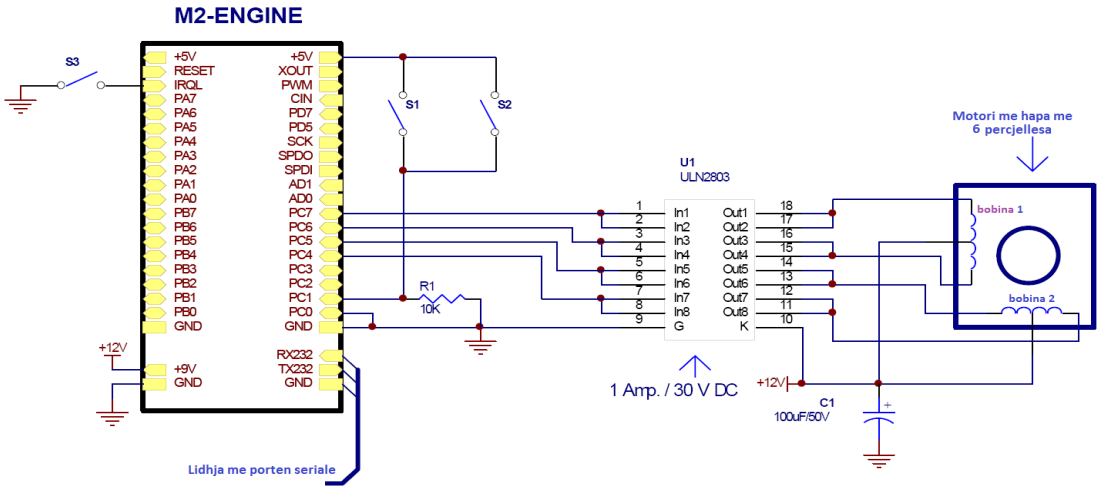


Fig 2.13. *Skema e detajuar e lidhjes së elementëve për realizimin e komandimit nga porta COM (RS232).*

Moduli M2-ENGINE përfaqëson një qark i cili converton sinjalin e transmetuar nëpërmjet RS232 në format për qarkun ULN2803. Në Fig 2.14 jepet moduli M2-ENGINE së bashku me elementet e tij [8].

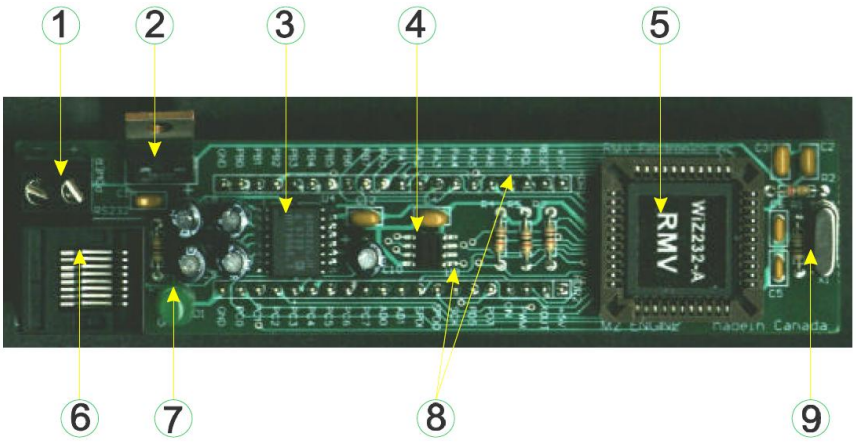


Fig 2.14. *Moduli M2-ENGINE.*

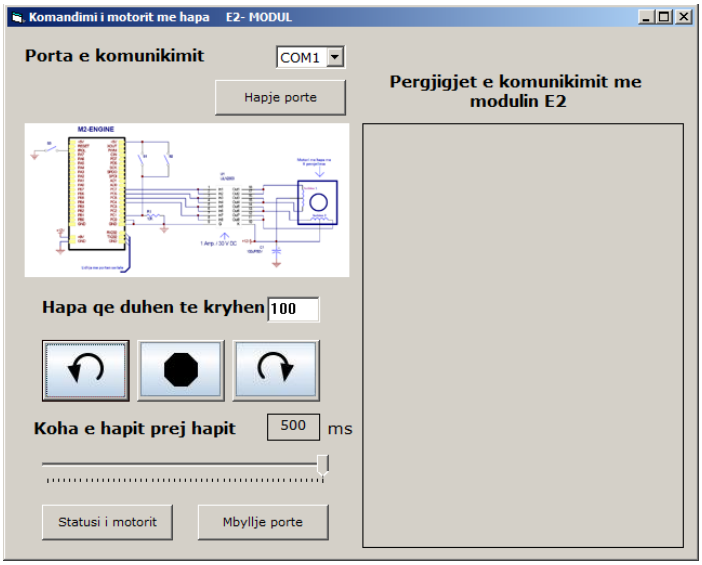
1- RS232 RJ45 connector
2- Power ON LED
3- Connection Headers
4- Time Base Quartz Crystal
5- Board Power Supply
6- 5V DC Voltage Regulator
7- RS232 -TTL Interface IC
8- Analog to Digital Converter
9- WiZ232 Microcontroller

Specifikimet për M2-ENGINE jepen si më poshtë :

- Serial Interface: EIA RS232 , 9600 to 115200 Bauds
- Serial Port Connection: RJ45, 8-positions, 8-contacts

- Digital I/O: 24 bits, TTL compatible
- Serial Peripheral Interface Port: 3 wires (Serial Clock, Data Input, Data Output)
- Analog To Digital Converter: 2 channels, 0-5 VDC, 10-bit
- Power Requirements: 7.5 to 12 VDC,100mA
- Dimensions (L,W,H): 28.1 x 117.1 x 29 (mm)

2.3.3. Programet e realizuara në Visual Basic për komandimin prej portës COM



Në Shtojcën-2 po japim programet e realizuara për komandimin e shpejtësisë dhe pozicionit të motorit me hapa prej portës COM.

2.4. Komandimi i motorave me hapa prej kompjuterit nëpërmjet portës USB

Për të realizuar komandimin e motorit Unipolar me hapa me 6 përcjellësa prej portës USB të kompjuterit ndërmjet disa mënyrave që ekzistojnë, do të analizohen dy mënyra të komandimit të motorit me hapa nëpërmjet USB :

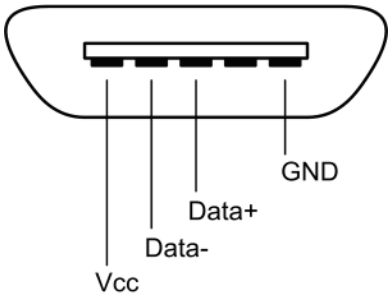
- Komandimi nëpërmjet moduli Stepper-Bee
- Komandimi nëpërmjet PIC18F4550-I/P Microchip Microcontroller & ULN2803

Komandimi nëpërmjet moduli Stepper-Bee përshkruhet në paragrafin në vijim.
Komandimi nëpërmjet PIC18F4550-I/P Microchip Microcontroller & ULN2803 do të jepet në kapitullin pasardhës përdorur në zbatime praktike.

2.4.1. Porta USB në PC.

Një portë USB (Universal Serial Bus) është një socket në kompjutera që mundëson lidhjen e tyre me një numër të madh paisjesh. USB ka patur për qëllim zëvendësimin e shumë llojeve të portave seriale dhe paralele, gjithashtu mund të lidhë pajisje periferike të kompjuterit si miun kompjuterik, tastierën kompjuterike, aparate fotografikë dixhitalë, printues, lojra mediatike personale etj. Për shumë nga këto pajisje, USB është bërë metodë standarde lidhjeje. USB është projektuar për

kompjutera personalë, por është bërë e zakonshme edhe në pajisje të tjera si smart phone, videolojrat etj. Posaçërisht USB HardDisk dhe USB Stick janë ndër më të pëlqyerit për bartjen e të dhënave në mënyrë të shpejtë dhe të pa komplikuar. Me disa nga USB-Stick mund të startohet edhe Sistemi Operues. Modeli më i ri është USB 2.0 e cila është më e shpejtë dhe moderne! Një USB-1.1-stick lexon të dhënat me një shpejtësi prej 0,8-0,9 MB/s kurse shkruan me një shpejtësi 0,6-0,8 MB/s, kurse USB 2.0 transferojnë të dhëna deri 480 Mbit/s. Porta në fjalë ka 4 kontakte (pins) dhe qëllimi i tyre shpjegohet më poshtë.



2.4.2. Skema elektronike.

Moduli Stepper-Bee që jepet në Fig 2.15, mund të lidhet njëkohësisht me dy motorë me të cilët komunikon në mënyrë të pavarur. Kjo do të thotë se gjatë kohës së komunikimit me motorin e parë duke kryer lëvizje rrotulluese me hapa në intervale kohore të përcaktuara, mund të komunikohet me motorin e dytë i cili mund të kryejë gjithashtu lëvizje rrotulluese me hapa të pavarura nga i pari dhe në intervale kohore të ndryshme me të parin. Mundësia e komandimit të dy motorëve në mënyrë të pavarur nga njeri-tjetri bën të mundur që moduli Stepper-Bee të ketë një përdorim shumë të madh në aplikacione të shumta që lidhen me komandimin e motorëve me hapa [20].

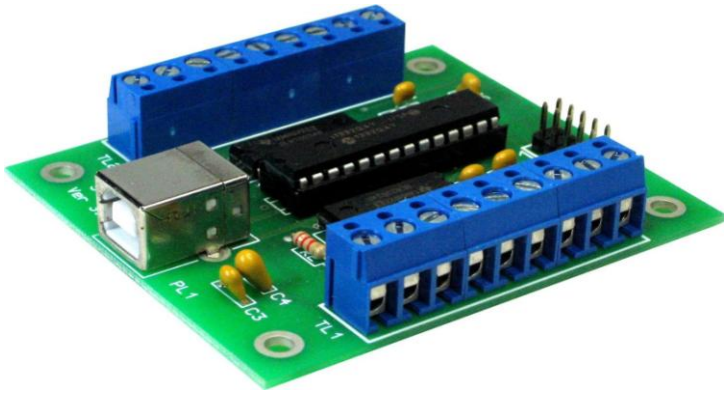


Fig 2.15. *Moduli Stepper Bee.*

Skema bazë e komandimit të motorëve Uniolarë me hapa nëpërmjet modulit Stepper-Bee jepet në Fig 2.16.

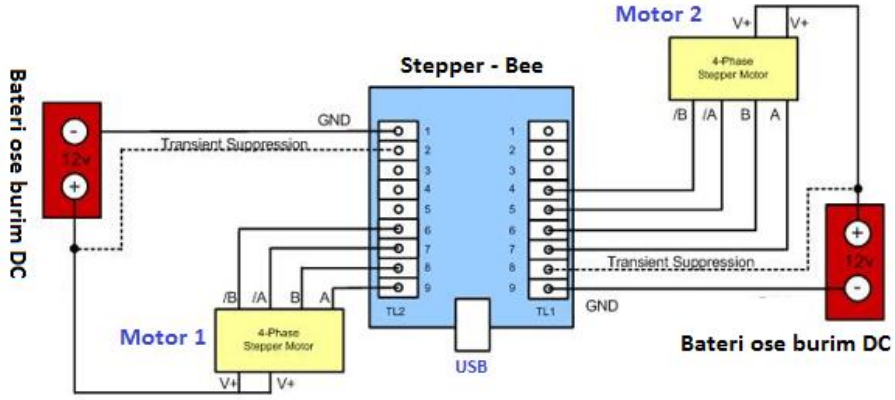


Fig 2.16. *Moduli Stepper-Bee dhe lidhja me dy motorë me hapa.*

2.4.3. Programet e realizuara në Visual Basic për komandimin prej portës USB.

Moduli Stepper-Bee vjen me një mundësi për të gjithë ata përdorues që kërkojnë një zgjidhje të shpejtë duke e përdorur atë në projektet e automatizimit. Ky modul ka një ndërfaqës midis përdoruesit (programuesit) dhe modulit i cili e lehtëson punën e komandimit të motorëve. Ndërfaqësi është një dynamic link library (dll) që emërtohet 'stp.dll', e cila përmbledh një numër funksionesh dhe komandash që krijojnë mundësinë e një komandimi profesional të motorëve me hapa. Dll siguron një ndërfaqe të thjeshtë dhe të shpejtë duke eleminuar problemet e komunikimeve me portën USB, probleme që shfaqen shpesh në dërgimin dhe marrjen e të dhënave në sinkron me paisjen komunikuese.

Funksionet bazë për të realizuar komandimet nëpërmjet stp.dll po i japim në vijim :

```
InitStp()
RunMotor1(steps, interval, direction, outputs)
RunMotor2(steps, interval, direction, outputs)
StopMotor1(outputs)
StopMotor2(outputs)
GetCurrentStatus(M1Active, M2Active, M1Steps, M2Steps, Inputs)
```

Deklarimet në Visual Basic 6.0 dhe Visual Basic.Net janë si më poshtë :

```
Declare Function InitStp Lib "stp.dll" () As Integer

Declare Function RunMotor1 Lib "stp.dll" ( ByVal steps As Integer,
ByVal interval As Integer,
ByVal direction As Integer,
ByVal outputs As Integer,
) As Boolean

Declare Function RunMotor2 Lib "stp.dll" ( ByVal steps As Integer,
ByVal interval As Integer,
ByVal direction As Integer,
ByVal outputs As Integer,
) As Boolean
```

```
Declare Function StopMotor1 Lib "stp.dll" ( ByVal outputs As Integer, ) As Boolean
Declare Function StopMotor2 Lib "stp.dll" ( ByVal outputs As Integer, ) As Boolean

Declare Function SetStepMode Lib "stp.dll" ( ByVal M1Mode As Integer,
ByVal M2Mode As Integer,
) As Boolean

Declare Function GetStatus Lib "stp.dll" ( ByRef M1Active As Integer,
ByRef M2Active As Integer,
ByRef M1Steps As Integer,
ByRef M2Steps As Integer,
ByRef Inputs As Integer,
) As Boolean
```

Para se të përdorim ndonjë nga funksionet e kontrollit të motorit në fillim kalojmë StepperBee në gjendje gati për punë duke përdorur funksionin InitStp().
Vënien në punë të motorit 1 dhe 2 e bëjmë nëpërmjet funksionit RunMotor1 dhe RunMotor2, të cilët kanë katër parametra që do të përcaktojnë mënyrën e punës së motorit. Parametrat e funksionit po përshkruhen më poshtë :

Steps - numër i plotë në diapazonin nga 1 deri 16000 e cila lidhet me numrin e hapave që kërkojmë të kryhen.

Interval - numër i plotë në diapazonin nga 1 deri 16000 e cila lidhet me hohën e ekzekutimit të hapit nga hapi.

Direction - numër i plotë me vlerat 0 ose 1. Zero nënkupton lëvizjen sipas një kahu dhe 1 lëvizjen në kahun e kundërt.

Outputs - numër i plotë me vlerat nga 0 deri 7 e cila nënkupton gjendjet 'on/off' të switching outputs të lidhura me motorin 1 ose 2. Kështu n.q.s Outputs është 5 (e cila në sistemin binar është 00000101) kjo do të thotë se daljet 1 dhe 3 janë 'on'.

Për të kuptuar me shumë funksionimin e këtij funksioni po japim një shembull. Për të rrotulluar motorin sipas drejtimit para me 400 hapa, ku hapi nga hapi bëhet çdo 50 ms, me switchet me gjendje off , komanda që duhet të jepet është :

```
RunMotor1(400, 50, 0, 0)
```

Kur kërkohet që njëkohësisht të kryhet dhe lëvizja e motorit tjetër me 300 hapa, me interval hapi prej hapit 30 ms, sipas drejtimit prapa dhe me switche me gjendjen off, komanda do të jetë :

```
RunMotor2(300, 30, 1, 0)
```

Lëvizjet e motorave me hapa janë të pavarur nga njeri tjetri duke bërë të mundur një rritje të mundësisë për realizimin cilësor të kontrolleve në projektet e automatizimit.

Funksioni GetCurrentStatus() function mund të thirret në çdo moment dhe nëpërmjet tij mund të përcaktohet gjendja e secilit prej motorëve me hapa si dhe gjendja e sinjaleve dixhitale në hyrje të motorëve. Ky funksion ka pesë parametra të cilët përshkruajnë gjendjet e motorëve. Thirrja e tij përshkruhet më poshtë :

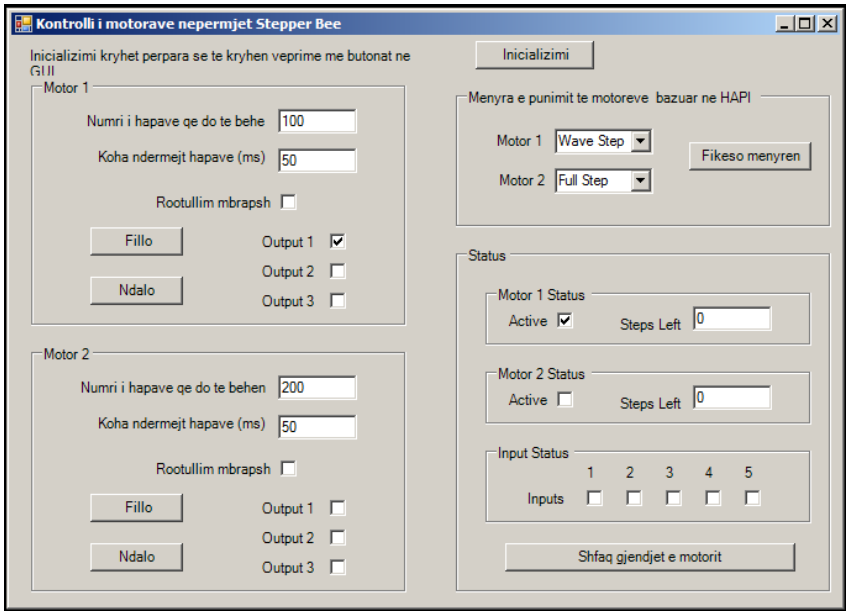
```
Dim M1Active, M2Active, M1Steps, M2Steps, Inputs As Integer
GetCurrentStatus(M1Active, M2Active, M1Steps, M2Steps, Inputs)
```

Pas ekzekutimit të funksionit të mësipërm parametrat marrin vlerat që përshkruajnë gjëndjen.

M1Active merr vlerën 1 kur motori 1 është aktivizuar në lëvizje, 0 për motorin 1 të ndaluar.

M2Active	merr vlerën 1 kur motori 2 është aktivizuar në lëvizje, 0 për motorin 2 të ndaluar.
M1Steps	numër i plotë në diapazonin 0 - 16000 që përcakton numrin e hapave që motori 1 ka përfunduar deri në atë moment.
M2Steps	numër i plotë në diapazonin 0 - 16000 që përcakton numrin e hapave që motori 2 ka përfunduar deri në atë moment.
Inputs	numër i plotë në diapazonin 0 - 31 dhe lidhet me gjendjen e switcheve on/off.

Më poshtë po paraqesim GUI e ndërtuar për komandimin e motorave 1 dhe 2 në gjuhën e programimit Visual Basic.Net.

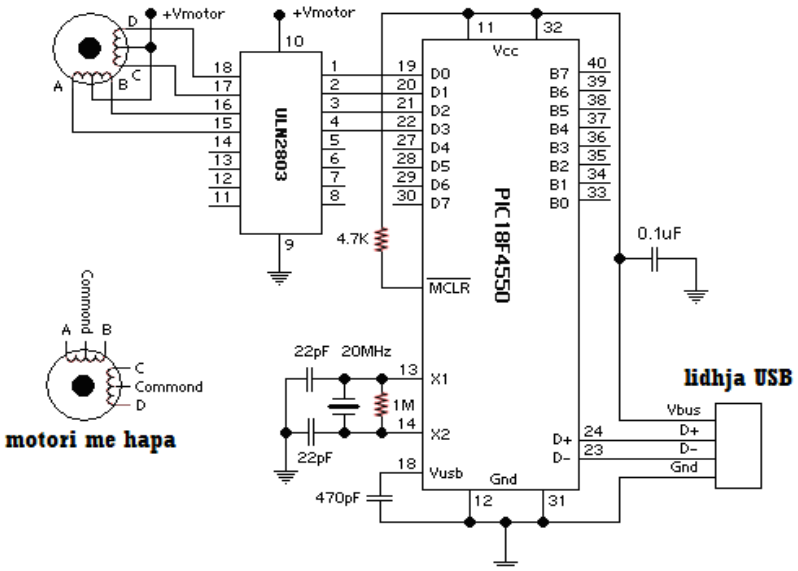


Në Shtojcën-3 po japim programet e realizuara për komandimin e shpejtësisë dhe pozicionit të motorit me hapa prej portë USB.

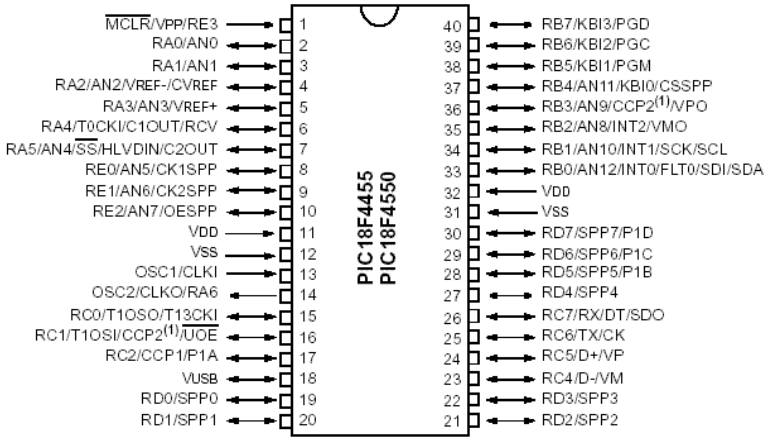
2.4.4. Komandimi nëpërmjet PIC18F4550-I/P Microchip Microcontroller & ULN2803

Komandimi i motorit me hapa realizohet nëpërmjet PIC18F4550-I/P i cili vë në lëvizje motorin me hapa nëpërmjet ULN2803. Në skemat e mëposhtëme jepet PIC18F4550 dhe emërtimet e kontakteve të tij.
Hapat që ndiqen në komandimin e motorit me hapa jepen në përshkrimin e mëposhtëm.
Programohet PIC18F4550-I/P nëpërmjet programit MicroCode Studio – PicBasic Pro V 3.0.7.0. Me ndihmën e aplikacionit EasyHID USB Wizard i ndërtuar për të punuar me USB 2.0, pjesë e programit PicBasic Pro, gjenerohen dy programe provizorë të kompiluar : i pari për kompjuterin si program PC Host i kompiluar edhe për Microsoft Visual Basic 6.0 dhe i dyti për paisjen (device) PIC18F4550-I/P të cilin e shkarkojmë nëpërmjet USB të kompiluar prej Compiler Proton310. Gjatë ndërtimit të programeve futet në konfigurimet e tyre për të personalizuar programet emri i aplikacionit (ProductID) si dhe një Unique ID (VendorID) për të siguruar personalizimin e programeve të kompjuterit dhe paisjes (PIC18F4550-I/P). Programi në kompjuter (PC Host) përmban një librari dinamike mcHID.dll, funksionet e të cilës si lidhja (connection), dërgimi i

datave (send data), marrja e datave (receive data) dhe shkëputja e lidhjes (disconnection), mund të shfrytëzohen prej aplikacionit GUI të ndërtuar në Microsoft Visual Basic 6.0 prej të cilit dërgohen komandat për rrotullimin e motorit me hapa.



(a)



(b)

(a) *Skema pricipiale elektronike e komandimit të motorit me hapa prej mikrokontrollerit.*
(b) Emërtimet e kontakteve për PIC18F4450

Në vazhdim po japim programin për PIC18F4550-I/P të ndërtuar në MicroCode Studio – PicBasic Pro V 3.0.7.0.

```
' select MCU and clock speed
Device = 18F4550
XTAL = 48
Declare PORTB_PULLUPS 1
' descriptor file, located in \inc\usb_18 - a copy
' is located in the same folder as this file
USB_DESCRIPTOR = "STEPUSBDESC.inc"
```

```
' USB Buffer...
Symbol USBBufferSizeMax = 8
Symbol USBBufferSizeTX  = 8
Symbol USBBufferSizeRX  = 8
Dim    USBBuffer[USBBufferSizeMax] As Byte

' some useful flags...
Dim PP0 As Byte SYSTEM        ' USBPOLL status return
Symbol CARRY_FLAG = STATUS.0  ' high if microcontroller does not have control over the
buffer
Symbol ATTACHED_STATE = 6     ' is USB attached

' ************************************************************
' * main program loop - remember, you must keep the USB      *
' * connection alive with a call to USBPoll, USBIn or USBOut *
' * every couple of milliseconds or so                       *
' ************************************************************
TRISD = %00000000
PORTD = %00000001
GoSub MakeSleep
PORTD = %00000011
GoSub MakeSleep
PORTD = %00000111
GoSub MakeSleep
PORTD = %00001111
GoSub MakeSleep
PORTD = %00000111
GoSub MakeSleep
PORTD = %00000011
GoSub MakeSleep
PORTD = %00000001
GoSub MakeSleep

' ************************************************************

GoSub AttachToUSB
ProgramLoop:
   USBIn 1, USBBuffer, USBBufferSizeRX, ProgramLoop
   PORTD = USBBuffer[0]
   GoTo ProgramLoop

' ************************************************************
' * receive data from the USB bus                            *
' ************************************************************
DoUSBIn:
   USBIn 1, USBBuffer, USBBufferSizeRX, DoUSBIn
   Return

' ************************************************************
' * transmit data                                            *
' ************************************************************
DoUSBOut:
   USBOut 1, USBBuffer, USBBufferSizeTX, DoUSBOut
   Return

' ************************************************************
' * wait for USB interface to attach                         *
' ************************************************************
AttachToUSB:
   Repeat
      USBPoll
   Until PP0 = ATTACHED_STATE
   Return

' ************************************************************
' * wait after command                                       *
```

```
' ************************************************************
MakeSleep:
 For i=0 To 255 Step 1
     For j=0 To 255 Step 1
         For k=0 To 10 Step 1
         Next
     Next
 Next
Return
```

Aplikacioni GUI i ndërtuar në Microsoft Visual Basic 6.0, i cili paraqitet në figurën e mëposhtme, prej të cilit dërgohen komandat për rrotullimin e motorit me hapa para dhe prapa si dhe mund të ndryshohet shpejtësia e rrotullimi. Nëpërmjet D0, D1, D2, D3 që identifikojnë katër portat e PIC18F4550-I/P ku janë lidhur katër portat e ULN2803, kryhet dërgimi i sinjaleve në motor. Parametri 'Interval' te një Timer Control në Visual Basic siguron një shpejtësi të ndryshueshme në rrotullimin e motorit me hapa. Dërgimi i sinjaleve në portat D0, D1, D2, D3 siguron rrotullimin para ose prapa të motorit ndërsa koha e dërgimit të këtyre sinjaleve përcakton dhe shpejtësinë e tij.

GUI për komandimin e motorit me hapa nëpërmjet PIC18F4450-I/P

Në programin e ndërtuar përshkruhen funksionet e komunikimeve si më poshtë :

- Lidhja nëpërmjet portës USB me PIC18F4550 > ConnectToHID (Me.hwnd)
- Ndërprerja e komunikimit me PIC18F4550 > DisconnectFromHID
- Dërgimi i datave në portë për rrotullimin e motorit me hapa :
  BufferOut(0) = 0
  BufferOut(1) = 10
  hidWriteEx VendorID, ProductID, BufferOut(0)
- Ndalimi i rrotullimit realizohet pas thirrjes dhe ekzekutimit të instruksioneve :
  BufferOut(0) = 0
  hidWriteEx VendorID, ProductID, BufferOut(0)

Programin e detajuar po e japim në Shtojcën 3.

2.5. Komandimi i motorave me hapa prej kompjuterit në distancë

Detyra e komandimit të motorit me hapa në distancë është një detyrë e shtruar prej kohësh. Realizimi i saj është bërë në mënyra të ndryshme të lidhura me nivelin e teknologjisë. Në kohët e sotme ky problem është bërë më i thjeshtë nisur nga zhvillimet e mëdha teknologjike dhe ulja e kostove të paisjeve që do të përdoren. Për të realizuar komandimin e një motori me hapa në distancë zgjidhjet që ofrohen janë rrjedhojë e kërkesave të proceseve industriale dhe zhvillimeve shoqërore.

Në këtë pjesë do të parashtroj dy mënyra komandimi që lidhen me kontrollin në distancë midis panelit kompjuterik të komandimit dhe bllokut që merr komandat tek i cili ndodhet dhe motori me hapa.

- Komandimi i motorit me hapa në distancë me valë (wireless) nga kompjuteri.
- Komandimi i motorit me hapa në distancë në LAN dhe WEB nga kompjuteri (trajtimi i këtij problemi nuk është pjesë e këtij materiali por do të jetë një punë e së ardhmes).

2.5.1. Komandimi i motorit me hapa në distancë me valë (wireless) nga kompjuteri.

Realizimi i këtij komandimi do të bëhet bazuar në bllok diagramën e dhënë në Fig 2.17 . Në bllok diagramë jepen të gjithë elementet që nevojiten për të realizuar detyrën e shtruar.

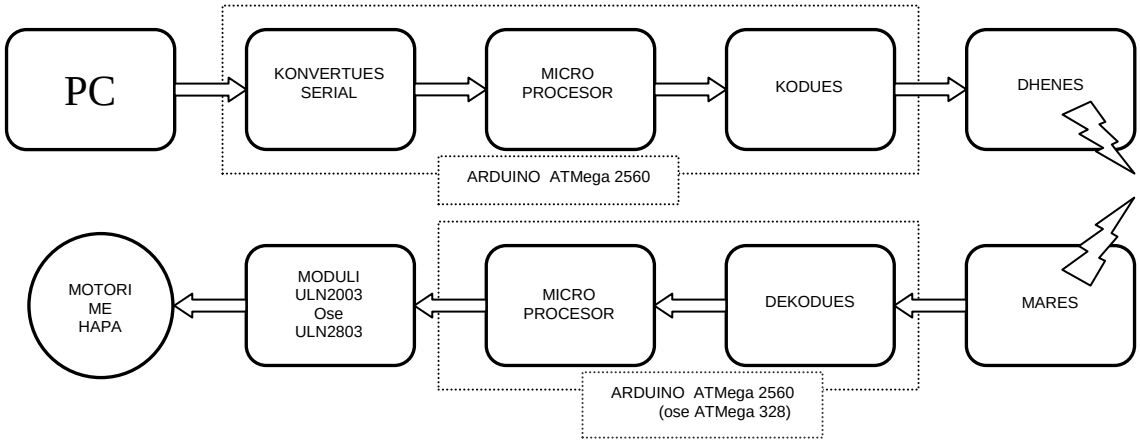


Fig 2.17. *Bllok diagrama e komandimit me vale te motorit me hapa* .

Nga kompjuteri (PC) nëpërmjet portës seriale, informacioni (komandat) jepet në hyrjen seriale të microcontrollerit ATMega 2560. Programi i ndërtuar për microcontroller ATMega2560 lexon të dhënat prej portës seriale dhe i kalon të koduara drejt dhënësit valor. Sinjali i transmetuar merret prej marrësit dhe kalon në hyrjen seriale të mikrokontrollerit të dytë ATMega2560 (ose microcontroller ATMega328) në bllokun e poshtëm në Fig 2.17. Prej microcontroller ATMega2560 (ose microcontroller ATMega328), ku është ekzekutuar programi që përpunon informacionet e marra nga porta hyrëse, sinjalet jepen në modulin ULN2003 (ose ULN2803) që përbën driverin e motorit me hapa. Këto sinjale elektrike ne dalje te modulit ULN2003 (ose ULN2803) vënë në lëvizje motorin me hapa.

Në Fig 2.18 jepen shpjegime mbi portat (pins) e ATMega2560 [19].

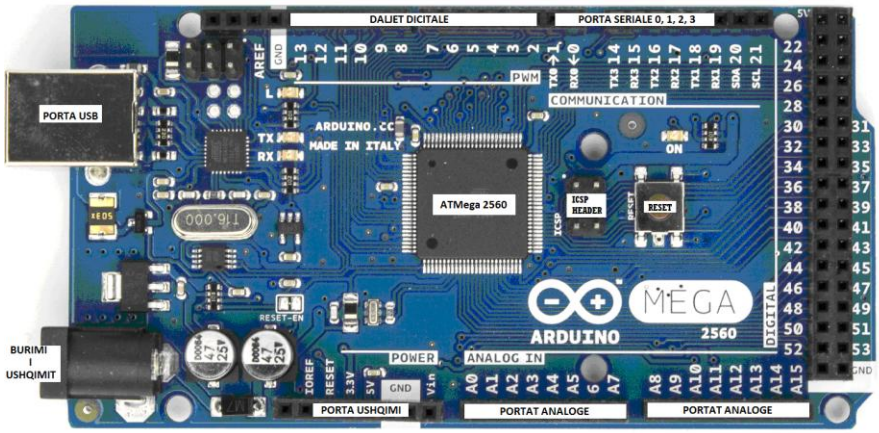


Fig 2.18. *Shpjegime për portat e ATMega2560.*

Shpjegime për pinet e burimit të ushqimit :

- V_{in} : *Kur ushqimi i paisjes bëhet nga jashtë nëpërmjet një konvertuesi* $AC->DC$ *përdoret kontakti* V_{in} *për të zbatuar tensionin*.
- $5V$: Arduino ATMega përdor tensionin prej $5V$ për të ushqyer të gjitha pjesët e veta. Tensioni i $5V$ mund të sigurohet nëpërmjet portës USB ose prej një gjeneratori tensioni ku për këtë rast tensioni zbatohet në V_{in} .
- $3.3V$: Në Arduino ATMega mund të zbatohet një tension $3.3V$ prej një gjeneratori tensioni. Në këtë rast rrymat maksimale të punës janë të rendit $50mA$.
- GND : Pika me potencial zero.

Arduino ATMega2560 ka 51 pine (pins) hyrëse dhe dalëse digitale që shërbejnë për të siguruar komunikimet me paisje të tjera (nga këto 15 pine mund të shërbejnë si dalje PWM (Pulse-Width Modulation)), 16 hyrje analoge, 4 UARTs (hardware serial ports), 1 element 16 MHz crystal oscillator, 1 USB connection, 1 power jack, 1 ICSP header dhe 1 buron reset.

Secili prej pineve tyre mund të këtë dy nivele tensionesh $0\ V$ (gjendja '0') dhe $5\ V$ (gjendja '1'). Rezistencat elektrike të hyrjes së tyre janë në vlerat $20-50\ \Omega$. Rrymat maksimale që lejohen në këto kontakte (pins) janë të rendit $40mA$. Më poshtë po japim specifikimet e mikrokontroller ATMega2560 :

Microcontroller ATmega2560
Operating Voltage 5V
Input Voltage (recommended) 7-12V
Input Voltage (limits) 6-20V
Digital I/O Pins 54 (of which 15 provide PWM output)
Analog Input Pins 16
DC Current per I/O Pin 40 mA
DC Current for 3.3V Pin 50 mA
Flash Memory 256 KB of which 8 KB used by bootloader
SRAM 8 KB
EEPROM 4 KB
Clock Speed 16 MHz

Në Fig 2.19 jepen paisjet dhënëse dhe marrëse model STT-433 RF Transmitter/Receiver [18].

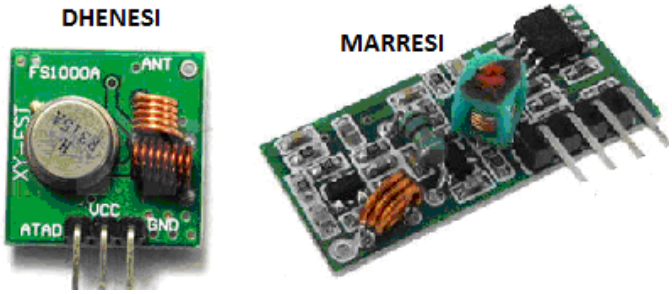


Fig 2.19. *Dhënësi dhe marrësi model STT-433 RF Transmitter/Receiver.*

Dhënës-Marrësi model STT-433 RF është një paisje me një përdorim jo të vogël dhe me një kosto të ulët. Dhënës-Marrësi STT-433 RF është një transmetues ideal për aplikacione ndërlidhjeje të tipit remote control. Për të vënë në punë këtë transmetues është i nevojshëm një burim ushqimi me tension në kufijtë $1.5-12\,V$. Në përbërje të tij gjendet një SAW-stabilized oscillator, për kontrollin sa më të saktë të frekuencave të sinjaleve.

Karakteristikat e STT-433 RF Transmitter/Receiver jepen si më poshtë :

Parametrat	**Simboli**	**Min.**	**Tipi**	**Max.**	**Njesia**
Operating Voltage	*Vcc*	*4.5*	*5.0*	*5.5*	*VDC*
Operating Current	*Icc*	-	*3.5*	*4.5*	*mA*
Reception Bandwidth BW	*rx*	-	*1.0*	-	*MHz*
Center Frequency	*Fc*	-	*433.92*	*MHz*	*MHz*
Sensitivity	-	-	*-105*	-	*dBm*
Max Data Rate	-	300	*1k*	*3K*	*Kbit/s*
Turn On Time	-	-	*25*	-	*ms*
Operating Temperature	*Top*	-10	-	*+60*	*°C*

Për të shpjeguar pinet e STT-433 RF Transmitter/Receiver po japim skematikisht emërtimet e tyre në Fig 2.20 dhe Fig 2.21.

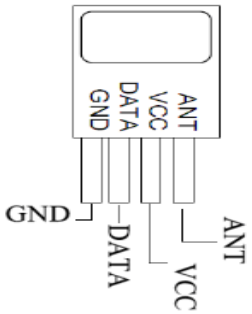


Fig 2.20. *STT-433 RF Transmitter.*

Në Fig 2.20, DATA jep kontaktin ku dërgohet sinjali pas kodimit të kryer në daljen e mikrokontrollerit Tx0.

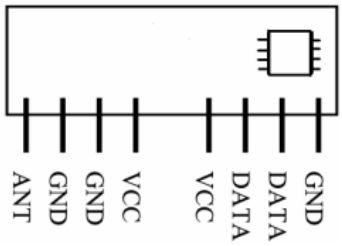


Fig 2.21. *STT-433 RF Receiver.*

Në Fig 2.21, DATA jep kontaktin ku merret sinjali dhe më pas jepet në hyrjen Rx0 të mikrokontrollerit ku bëhet dhe dekodimi.

Në Fig 2.22 jepet moduli ULN2003 me dioda led që shërben si modul ndërmjetës midis mikrokontroller ATMega dhe motorit me hapa. Gjatë dhënies së impulseve të ndryshme në hyrje, blloku i diodave led (A B C D) shpreh visualisht nivelet e tyre në pinet 1N1, 1N2, 1N3 dhe 1N4. I njëjti modul ndërtohet edhe me CIs ULN2803, duke zëvendësuar ULN2003 me ULN2803 [15][16].

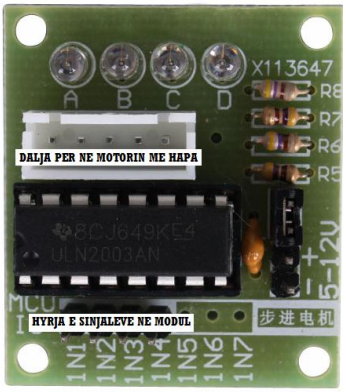


Fig 2.22. *Moduli ULN2003.*

Pasi kemi përshkruar të gjithë modulet që do të përdorim në kryerjen e komandimit me valë të motorit me hapa prej kompjuterit, jemi gati të japim dhe skemën e përgjithshme të punës. Në Fig 2.23 dhe Fig 2.24 jepen dy skema funksionale të komandimit me blloqet IC të motorit me hapa.

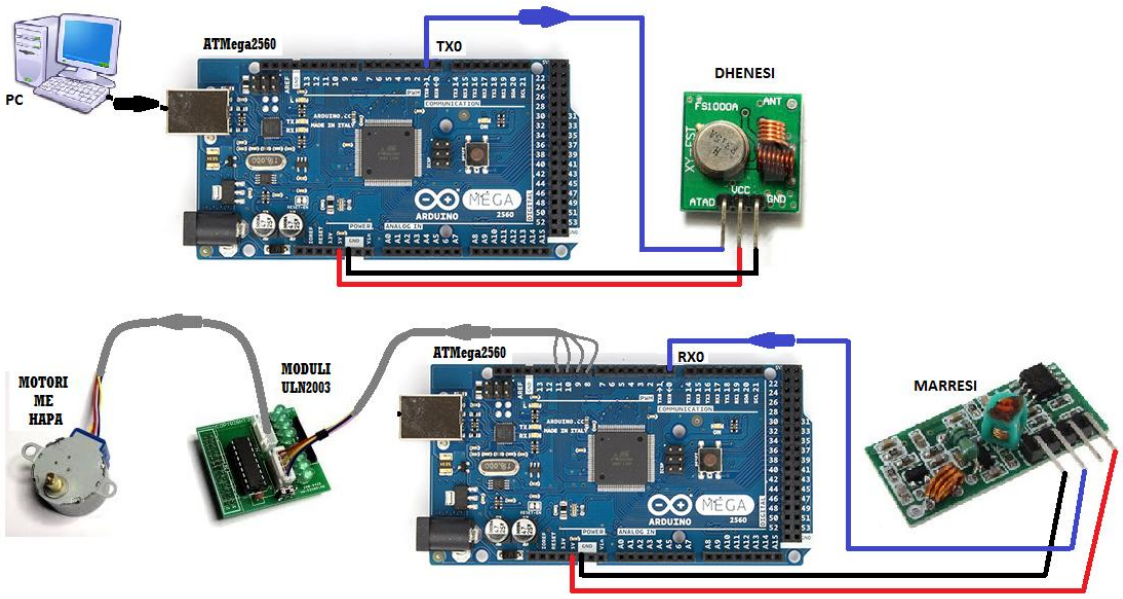


Fig 2.23. *Skema e komandimit me valë të motorit me hapa duke përdorur dy qarqe ATMega2560.*

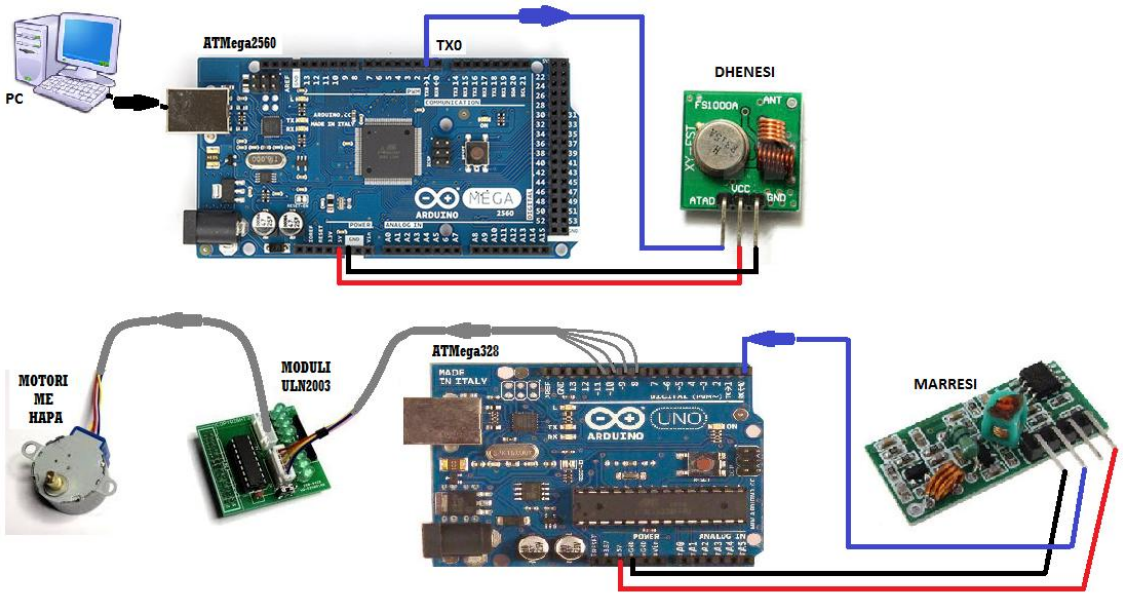


Fig 2.24. *Skema e komandimit me valë të motorit me hapa duke përdorur ATMega2560 dhe ATMega328.*

2.6. Programet e realizuara në C++ dhe Visual Studio 6 për komandimin në distancë, me valë

Programet që realizojnë komandimin me valë të motorit me hapa përbëhen prej tre pjesësh. Programi i parë, realizuar në Visual Basic 6.0 që përbën dhe GUI (Fig 2.25), siguron mundësinë e dërgimit të komandave drej mikrokontrollerit ATMega2560 nëpërmjet portës seriale të kompjuterit (COM 1-9). Pas zgjedhjes së portës seriale, përcaktohet bound rate për portën në vlerat (9600, N, 8, 1), që përcaktojnë shpejtësinë e komunikimit të kompjuterit (nëpërmjet portës seriale) me

mikrokontrollerin ATMega2560 (nëpërmjet portës seriale 0 në mikrokontroller). Instruksionet që dërgojnë komandat (për rrotullim të motorit sipas një drejtimi të caktuar me shpejtësi të ndryshueshme) drejt portës së mikrokontrollerit ATMega2560 janë :

```
CommObject.Output = "+"      ' (+) karakteri për rritje shpejtësie rrotullimi për motorin ( x 2)
CommObject.Output = "-"      ' (-)  karakteri për ulje shpejtësie rrotullimi për motorin ( / 2)
CommObject.Output = "R"      ' (R) karakteri për rrotullim orar
CommObject.Output = "L"      ' (L) karakteri për rrotullim anti-orar
CommObject.Output = "0"      ' (0) karakteri për resetim gjendjesh
```

Karakteret komanda të dërguara nga GUI, lexohen prej programit të startuar në mikrokontrollerit ATMega2560.

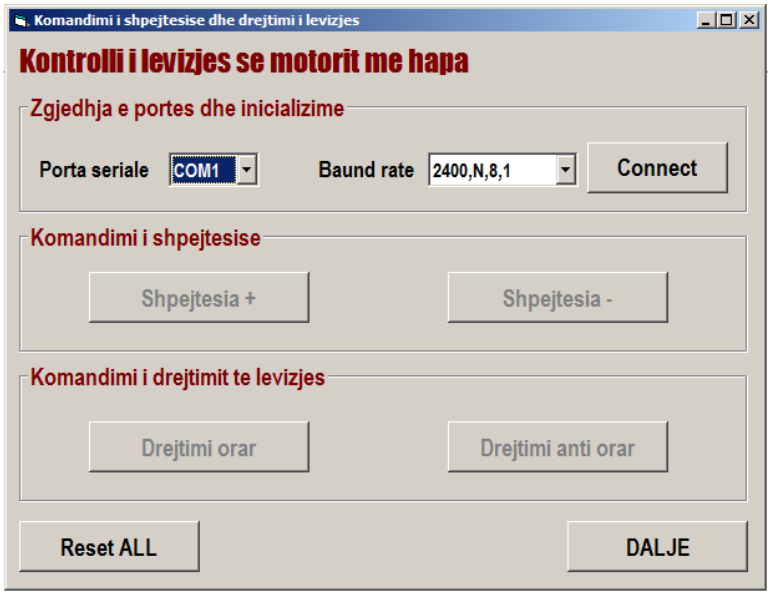


Fig 2.25. *GUI ne VisualBasic 6.0.*

Programi në mikrokontrollerin ATMega2560, i ndërtuar në C++, lexon datat e hyra në portën seriale 0 dhe në bazë të tyre, sipas programit, dërgon në portën seriale 1, të lidhur me transmetuesin, komandat e koduara. Instruksionet që kryejnë këtë detyrë jepen si më poshtë :

```
void loop() {
// lexoj nga porta seriale 0, dergoj data ne porten seriale 1:
if (Serial.available()) {
char inByte = Serial.read();     (LEXIMI I KARAKTEREVE KOMANDA PREJ SERIAL 0)

// Ndertoj protokollin e komunikimit per dergimin e komandave
if(inByte=='l'||inByte=='L') {                      // drejtimi antiorar
        writeUInt(100);          (DERGIMI I KOMANDAVE TE KODUARA DREJT SERIAL 1)
        blink(4);
        }
    else if(inByte=='r'||inByte=='R') {             // drejtimi orar
        writeUInt(101);          (DERGIMI I KOMANDAVE TE KODUARA DREJT SERIAL 1)
        blink(4);
        }
    else if(inByte=='+') {                          // rritje e shpejtesise se rrotullimit
        writeUInt(102);          (DERGIMI I KOMANDAVE TE KODUARA DREJT SERIAL 1)
        blink(6);
```

```
        }
    else if(inByte=='-') {                          // ulja e shpejtesise se rrotullimit
        writeUInt(103);          (DERGIMI I KOMANDAVE TE KODUARA DREJT SERIAL 1)
        blink(6);
        }
    else if(inByte=='0'){              // Reset gjendjet
        writeUInt(104);          (DERGIMI I KOMANDAVE TE KODUARA DREJT SERIAL 1)
        blink(8);
        }
    }
}
```

Pas transmetimit prej RF-Transiter, informacion nëpërmjet RF-Receiver hyn në mikrokontroller ATMega2560 (ose ATMega328) në portën seriale 0. Programi në mikrokontrollerin ATMega2560 (ose ATMega328), i ndërtuar në C++, lexon datat e hyra në portën seriale 0 dhe në bazë të tyre, sipas programit, dërgon në pinet [8, 9, 10, 11] impulset e duhura.

```
void loop() {
// print the string when a newline arrives:
rfDataCommand=readUInt(true);                       (LEXIMI I KARAKTEREVE KOMANDA PREJ SERIAL 0)

//kontroll i shpejtesise
if(rfDataCommand==100){
    blink(2);                                       // pulso Led dy here
    stepperDirection=true;                          // drejtimi anti-orar
    }
else if(rfDataCommand==101){
    blink(3);                                       // Pulso Led tre here
    stepperDirection=false;                         // drejtimi orar
    }
else if(rfDataCommand==102){
    blink(4);                                       // Pulso Led kater here
    if(speedDefaind<2048) speedDefaind=speedDefaind*2;
    stringComplete = true;
    }
else if(rfDataCommand==103){
    blink(5);                                       // Pulso Led pese here
    if(speedDefaind>4) speedDefaind=speedDefaind/2;
    stringComplete = true;
    }
else if(rfDataCommand==104){
    blink(6);                                       // Pulso Led gjashte here
    speedDefaind=32;
    stepperDirection = false;
    stringComplete = true;
    }

if(stringComplete){
    OCR1A=(156624/speedDefaind); (NDRYSHIMI I FREKUENCES SE DHENIES SE SINJALEVE NE PINET 8,9,10,11)
    stringComplete = false;
    }
}
// timer 1 interuppt service routine
ISR(TIMER1_COMPA_vect){
    if ((count2 == 0) || (count2 == 1)) {
    count2 = 16;
    }
count2>>=1;
                                                        (DERGIMI I IMPULSEVE NE PINET 8,9,10,11)
if(stepperDirection==false) {
        for (count = 3; count >= 0; count--) {
        digitalWrite(motorPins[3 - count], count2>>count&0x01);
        }
```

```
    }
else if(stepperDirection) {
        for (count = 3; count >= 0; count--) {
        digitalWrite(motorPins[count], count2>>count&0x01);
        }
    }
}
```

Gjendjet në pinet [8, 9, 10, 11] përcillen drejt modulit ULN2003 (ose ULN2803) prej nga më tej në motorin me hapa.
Në Shtojcën-4 po japim tre programet e realizuara për komandimin valor të motorit me hapa, për komandimin e shpejtësisë dhe pozicionit të tij.

Si përfundim mund të themi se :

- Nëpërmjet materialeve të këtij kapitulli u treguan disa mënyra se si mund të kryhet komandimi i motorit me hapa prej kompjuterit, si një prej elementëve kryesorë në paisjet e komanduara dhe robotike, nëpërmjet portave I/O.

- Mënyrat e komunikimeve na tregojnë përparësitë apo problematikat e tyre. Një prej parametrave që specifikon këto mënyra komunikimi lidhet me shpejtësinë e komunikimeve në dërgimin dhe marrjen e të dhënave. Në të gjitha rastet e komunikimeve me motorin me hapa, shpejtësia e dërgimit të komandave (frekuenca e veprimeve komanduese) është jo më e madhe se parametri i frekuencës së motorit me hapa.

KAPITULLI III

Hyrje

Robotistët po punojnë për realizimin e robotëve autonomë të lëvizshëm që mund të performojnë detyra të dobishme në mjediset njerëzore. Këto mjedise janë një sfidë e konsiderueshme për shkak të kompleksitetit të tyre.
Navigimi i robotëve në mjediset njerëzorë është shqyrtuar edhe më parë. Në pjesën më të madhe të teknikave që përdoren, robotët llogaritin distancën dhe zhvendosjen kundrejt objekteve të caktuara, të cila përcaktohen nëpërmjet sensorëve lazër, termike ose ultrasonikë, për të naviguar në mjedise të mbyllura. Megjithatë, disavantazhet e sensorëve lazer, termike dhe ultrasonikë qëndrojnë në faktin që vlerat e marra nga sensorët kërkohen për çdo njësi distance, që do të thotë se për të krijuar një ide të plotë të mjedisit përreth, duhen përdorur një numër i madh sensorësh. Për më tepër, për të arritur llogaritje të sakta, ato do të duhet të vendosen pingul me shënjestrën.
Kohët e fundit, navigimi i robotëve me ndihmën e pamjes vizuale ka tërhequr shumë studiues. Metodat e ndërtuara kanë përfshirë përdorimin e sinjaleve të marra prej sensorëve bazuar në Stereo Vision, Monocular Vision si dhe në kombinimet e pamjeve vizuale me informacionet e sensorëve të tjerë. Metodat gjithashtu ndryshojnë në mënyrën se si ato merren me informacionin e përkohshëm, që nga përpunimi i pamjeve (frame) individuale deri në llogaritjet, rrjedhojë e krahasimeve të shumë pamjeve vizuale të njëpasnjëshme [21].

3.1. Disa metoda optike të vlerësimit të distancës së objekteve prej sistemeve të kontrollit.

Në procesin e kontrollit të pozicionit dhe të shpejtësisë në rastin e ndjekjes së objekteve në sistemet robotike, një vend të rëndësishëm zë përcaktimi i distancës dhe zhvendosjes së sistemit të kontrollit prej objektit në vëzhgim. Ekzistojnë tre teknika tipike për matjen e distancës (position measurement systems -PMS) [27] së një roboti prej një objekti të paracaktuar.
Këto teknika për vlerësimin e PMS-së janë :

- trekëndëshave (triangulation)
- analizimi i skenës (scene analysis)
- afërsia (proximity)

Teknika e trekëndëshave (triangulation) përdor vetitë gjeometrike të trekëndëshit për të llogaritur vendndodhjen e objekteve. Teknika më e njohur është Global Positioning System (GPS). Megjithatë, GPS-i, duke qenë i varur nga sateliti, karakterizohet nga një problem i cili nuk e lejon të përcaktojë me saktësi vendndodhjen e objekteve brenda një ndërtese. Një teknikë e llogaritjes së afërsisë (proximity) përcakton se kur një objekt është afër një vendi të njohur, dhe prania e objektit mund të "ndihet" me anë të analizave për fenomene të ndryshme fizike që shfaqen. Disa teknika të famshme janë duke zbuluar kontaktin fizik ose duke monitoruar Access Point-ët e telefonave celular [28, 29]. Teknika e analizimit të skenës (scene analysis) përdor vecoritë nga vëzhgimi i skenës nga një pikë e caktuar dhe nxjerr konkluzione për vendndodhjen e sistemit të vëzhgimit dhe të objektit që ndiqet. Disa teknika të njohura janë :

- sistemi i radarëve
- sistemi i lokalizimit të pamjeve vizuale

Në teknikën e lokalizimit në ambjente të brendshme, rrezet infrared [28], ultrasonike [33], laser range finder [31, 32], RFID [30] dhe radari janë teknikat më të njohura wireless. Teknologjia infrared përdoret gjerësisht për të lokalizuar vendndodhje të objekteve, por transmetimi i sinjalit dhe

kërkesa lidhen me fushë pamjen që ka sistemi vëzhgues me objektin. Lokalizimi ultrasonik përdor një teknikë që njihet me emrin "time-of-flight" për të përftuar vendndodhjen e objektit në vëzhgim. Megjithatë, përdorimi i ultratingujve kërkon një infrastrukturë të zhvilluar në mënyrë që teknika të jetë e saktë dhe efektive. Matja e distancës me anë të lazerave bëhet duke matur kohën që i duhet dritës së lazerit të reflektohet mbrapsht nga shënjestra dhe të kthehet te dërguesi. Duke qenë se laser range finder është një aparat matës shumë i shpejtë dhe i saktë, përdoret gjërësisht në shumë aplikacione. Në [31, 32], Subramanian dhe Barawid propozuan një sistem drejtuesi automjetesh autonom të bazuar në një laser rangefinder. Laser rangefinder-i u përdor për të përftuar informacionin për distancën e objekteve në ambient që më pas do të përdorej për të naviguar dhe për të shmangur pengesat. Në [33], Thrun solli një metodë navigimi të pavarur të bazuar në një algoritëm të vecantë. Këto punime konfirmojnë që laser rangefinders-at janë të performancave të larta dhe të matjeve të sakta. Megjithatë, perfomanca e tyre e lartë bazohet në hardwaret me kosto të lartë. Lokalizimi i bazuar në RFID, përdor ndjekjen e RF-ve me një lexues me antenë që të lokalizojë objekte, por gjetja e secilës RF mund të funksionojë në distancë afërsisht 4 deri në 6 metra. Për të përmirësuar saktësinë e ulët në lokalizimin e pozicionit, teknologjia e njohur SpotON përdor disa algoritma, të bazuar në analizën e forcës së sinjaleve të radios për ndjijimin e vendndodhjeve 3D. Megjithatë një sistem i tillë nuk është akoma i plotë. Një sistem RADAR, i bazuar në RF, përdor përshtatësin e network-ut 802.11, për të matur fuqinë e sinjaleve në një numër të madh stacionesh bazë. Fatkeqësisht, pjesa më e madhe e rasteve nuk sjell një saktësi të dëshiruar. Në lokalizimin e robotëve kundrejt objekteve, në ambiente të brendshme, pjesa më e madhe e këtyre teknikave valore (wireless) përdoren për të performuar skanime të pengesave statike rreth robotëve, dhe pozicioni llogaritet duke i lidhur ato skanime me një hartë metrike të ambientit. Në ambiente dinamike vecorite e dalluara ndonjëherë nuk janë mjaftueshëm për të vlerësuar lokalizimin.
Siç dihet për të realizuar gjetjen e distancave dhe llogaritjen e zhvendosjeve, ka shumë mënyra bazuar ne fizikën optike, në funksion të paisjeve që përdoren. Një klasifikim i metodave optike i bazuar dhe në dukuritë fizike që shfrytëzohen do të qe si më poshtë [26]:

- Intensity-Based Sensors.
- Triangulation Sensors
- Time-of-Flight Sensors
- Confocal Sensors
- Interferometric Sensors
- Multiple-Wavelength and Scanning Interferometry
- Frequency Modulated Continuous Wave Time-of-Flight

Në këtë punim ne do të merremi vetëm me metodat optike dhe për të qenë të saktë me tre prej tyre, për realizimin e idesë përfundimtare të navigimit të robotëve drejt objekteve të paracaktuar në mjedise të ndryshme.

Në materialin në vazhdim do të tregohet se si bëhet ky vlerësim i distancës dhe zhvendosjes nëpërmjet tre teknikave pa sensorë, të cilat në formë të përmbledhur po i japim më poshtë :

- Një sistemi të përbërë prej kompjuterit, një kamere (Monocular Vision) dhe një gjeneruesi lazeri pikësor. Imazhi i marë nga kamerat përpunohen duke përcaktuar nëpërmjet një algoritmi të thjeshtë pozicionin e rezes së dritës (lazer) mbi objektin që vëzhgojmë. Nëpërmjet këtij pozicionimi mundemi të përcaktojmë më tej distancën e objektit prej sistemit të kontrollit.

- Një sistemi të përbërë prej kompjuterit dhe dy kamerash (Stereo Vision).
  Llogaritjet për përcaktimin e distancës mbështeten në parimet e funksionimit të sistemit të kamerave si dhe në përdorimin e koncepteve gjeometrike. Objekti të cilit i përcaktohet distanca në këtë rast është një trup i targuar d.m.th një trup ne një shenjë të dallueshme nga mjedisi përreth ose mbi të cilin godet një reze drite lazer duke siguruar shënjimin e objektit.

- Një sistemi të përbërë prej kompjuterit dhe një kamere (Monocular Vision) ku nëpërmjet përpunimit të imazhit dhe analizave gjeometrike duke shfrytëzuar principet e prespektivës, llogaritet distanca e një objekti prej planit të kamerës. Kjo njihet si metoda e vlerësimit të distancës së objekteve prej një kamere referuar vlerësimit të thellësisë (Depth Estimation).

Përpunimet e të dhënave kryhen nëpërmjet programeve në kompjuter të ndërtuar në mjediset Visual Studio 6.0 (Visual Basic), Visual Studio.Net si dhe në Matlab R2013b.
Në këtë kapitull do të paraqiten metodat dhe rezultatet e aritura prej realizimeve eksperimentale për sejcilën prej tyre.

3.2. Metoda e vlerësimit të distancës së objekteve nëpërmjet një sistemi i përbërë prej një kamere dhe lazerit pikësor.

Përdorimi i teknikave të dhëna më sipër bëhet me një qëllim të vetëm; evidentimin dhe shmangien e pengesave që dalin gjatë lëvizjes së sistemeve robotikë. Në vijim do të shtjellohet se si distanca e një objekti mund të përcaktohet me ndihmën e një kamere, lazeri pikësor dhe kompjuterit. Pas aktivizimit të lazerit Fig 3.1, mbi objektin të cilit do t'i përcaktojmë largësinë nga sistemi i kontrollit projektohet njolla e rrezes së lazerit (objekti ndodhet në fushën e pamjes së kamerës). Matematika që do të përdoret është e thjeshtë kështu që kjo teknikë funksionon plotësisht. Modeli llogaritës është paraqitur në figurën e mëposhtëme.

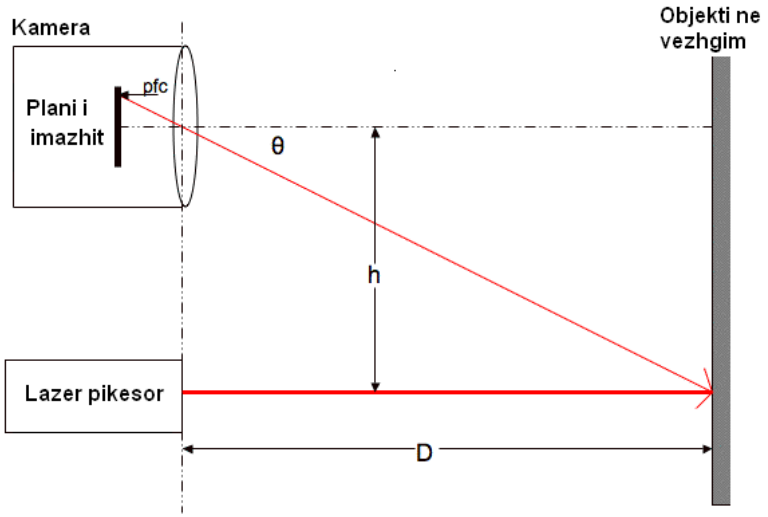


Fig 3.1. *Imazhi i objekteve prej një lenteje*

Kjo rreze lazer është në mënyre ideale paralele me boshtin optik të kameras. Supozohet se ne pamjen e marrë në kamera pika e lazerit është pika më e shdritshme. Nëpërmjet një algoritmi të

thjeshtë të shprehur në njërën prej gjuhëve të programimit Visual Studio 6.0 (Visual Basic), Visual Studio.Net, MatLab R2013b, bëjmë të mundur përcaktimin e njollës së lazerit ne imazhin e marrë. Sa më afër qendrës se imazhit të jetë njolla e lazerit, aq më larg ndodhet objekti që vëzhgohet. Distanca midis kameras dhe lazerit pikësor është e njohur dhe zgjidhet prej nesh (h). Distanca e panjohur D përcaktohet si më poshtë :

$$D = \frac{h}{\tan(\theta)}$$

Për të përcaktuar distancën e panjohur është e kuptueshme që duhet të njohim këndin θ. Përcaktimi i tij bëhet nëpërmjet formulave të mëposhtme.

$$tg(\theta) = \frac{h}{D}$$
$$\theta = pfc \cdot rpc + r_0$$

Përfundimisht mund të themi se shprehja llogaritëse përfundimtare do të jetë si në vijon [22], [23]:

$$pfc \cdot rpc + r_0 = arctg(\frac{h}{D})$$

ku :

pfc - pika prej qendrës deri tek projeksioni i lazerit

rpc - radianë për pixel

r_o - vlerë korrigjuese për mbulimin e gabimit

Nga termat e mësipërme për këndin θ ato që duhet të përcaktohen janë pfc, rpc dhe r_o. Madhësia pfc përcaktohet duke analizuar figurën e marë prej kameras. Vlerësohet numri i pikave prej qendrës së kameras deri në shenjën e bërë prej rrezeve lazer. Dy madhësitë e tjera përcaktohen në mënyrë eksperimentale duke shfrytëzuar distanca të njohura.

pfc (pixel)	distancë e zgjedhur (cm)
105	29
100	28.5
65	58.3
57	73
49	90
45	110
42	128
40	158
37	189
35	218

Nisur nga vlera të njohura të distancave krijohen sisteme ekuacionesh algjebrikë linearë nga ku përcaktojmë madhësinë e rpc dhe r_o, për çdo çift vlerash të distancës së njohur si më poshtë:

$$pfc^{(1)} \cdot rpc + r_0 = arctg(\frac{h^{(1)}}{D})$$

$$pfc^{(2)} \cdot rpc + r_0 = arctg(\frac{h^{(2)}}{D})$$

$$\dots$$

$$pfc^{(n-1)} \cdot rpc + r_0 = arctg(\frac{h^{(n-1)}}{D})$$

$$pfc^{(n)} \cdot rpc + r_0 = arctg(\frac{h^{(n)}}{D})$$

Për të patur vlera sa më të sakta me gabime minimale, llogaritjet e madhësive rpc dhe r_o bëhen disa herë për distanca të ndyshme.

$$rpc = \frac{arctg(\frac{h^{(2)}}{D}) - arctg(\frac{h^{(1)}}{D})}{pfc^{(2)} - pfc^{(1)}}$$

$$r_0 = arctg(\frac{h^{(1)}}{D}) - pfc^{(1)} \cdot \frac{arctg(\frac{h^{(2)}}{D}) - arctg(\frac{h^{(1)}}{D})}{pfc^{(2)} - pfc^{(1)}}$$

Më tej llogaritet një vlerë mesatare për këto madhësi [25].

Provat eksperimentale janë kryer me paisjet e mëposhtme :

- Kamera HAMY c-200 Wireless Camera
- Lazer Pikësor Sin 2013908 Wavelength 650 nm±10 Max Output <5mW

Bazuar në paisjet e përdorura dhe nisur nga formulat e mësipërme bëhet llogaritja e madhësive rpc dhe r_o sipas metodës së treguar më sipër.

Vlerat e llogaritura janë :

rpc = -0.0565 rad

r_o = 0.002426 rad / pixel

Duke përdorur rezultatet e llogaritjeve të mësipërme nëpërmjet formulës $D = \frac{h}{\tan(pfc \cdot rpc + r_0)}$ bëhet e mundur llogaritja e distancës së kërkuar h.

$$h = D \cdot \tan(pfc \cdot rpc + r_0)$$

Në tabelën e mëposhtëme jepen distancat e llogaritura nga sistemi i kontrollit të distancës duke zgjedhur të njëjtat vlera të *pfc* si në tabelën e mësipërme.

pfc (pixel)	distancë e llogaritur (cm)	distancë e zgjedhur (cm)	gabimi relativ %
105	29.4	29	1.36
100	29.2	28.5	2.40
65	60.0	58.3	2.83
57	76	73	3.95
49	92	90	2.17
45	114.3	110	3.76
42	125.2	128	-2.24
40	164	158	3.66
37	192	189	1.56
35	210	218	-3.81

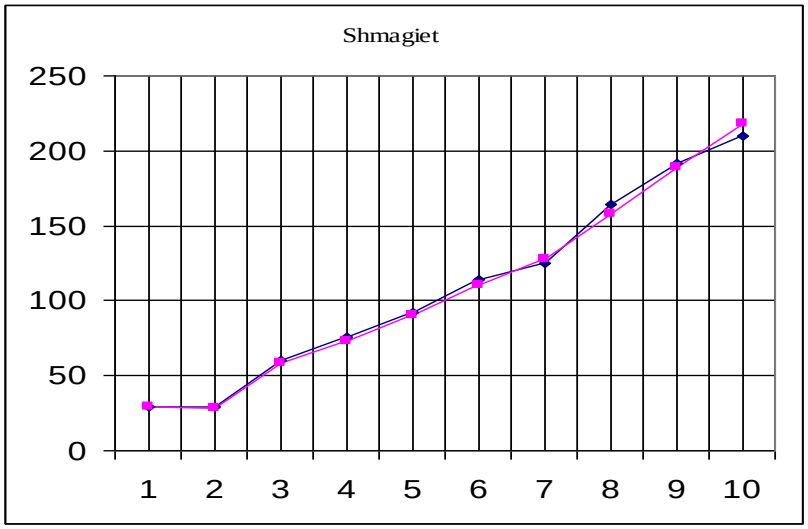


Fig 3.2. *Grafiku i shmagjeve midis vlerave të matura dhe atyre të zgjedhura për të njëjtën pfc.*

Në tabelën e mësipërme, kolona e fundit shpreh gabimin e realizuar prej sistemit të kontrollit të distancës, krahasuar me vlerat e matura paraprakisht. Kjo tregon se gabimi nuk kalon vlerat e 4 %.
Në Fig 3.2 jepet shmangia midis vlerave të matura dhe atyre të zgjedhura për të njëjtën *pfc* .
Për realizimin e tabelave të mësipërme u përdorën programet e ndërtuara në gjuhët e programimit Visual Studio 6.0 (Visual Basic), Visual Studio.Net si dhe në MatLab R2013b.
Sistemi i realizuar sipas përshkrimit të mësipërm krijon avantazhin e dukshëm të thjeshtësisë së tij si dhe të saktësisë relative të kërkuar. Kufizimi i këtij sistemi për përcaktimin e distancës është problemi i pozicionimit majtas apo djathtas i objektit. Që ky sistem të japë vlerësime pozicioni jo vetëm sipas një drejtimi lëvizjeje është e nevojshme të modifikohen algoritmat e tij, pjesë kjo e një pune në të ardhmen.

3.3. Algoritmika dhe programet e realizuara Visual Studio 6 (VB)

Për të realizuar përcaktimin e distancës së një objekti të shënjestruar, prej një sistemi kamerash & lazer të përshkruar në paragrafin e mësipërm, po japim algoritmin dhe programin e ndërtuar në Visual Studio 6 (VB).
Hapi i parë që kryhet është kalibrimi i sistemit të përcaktimit të distancës. Kjo do të realizohet në këtë mënyrë :

1. Ekzekutohet programi dhe pasi në GUI shfaqet imazhi i marrë nga kamera, ndizet lazeri pikësor. Sigurohemi që pika e ndricimit të lazerit godet në objektin tonë në shqyrtim.
2. Vendosim sistemin kamera lazer (këto paisje janë të vendosura mbi trupin e robotit) në një distancë të përcaktuar (të njohur). Në textbox ngjitur me butonin 'Kalibrimi', vendosim vlerën e distancës së njohur.
3. Shypim butonin 'Kalibrimi' i cili ekzekuton kodet e mëposhtme dhe nëpërmjet tyre bëhet llogaritja e variablit Koeficienti_Sys_Lazer_Kamera i cili do të shfrytëzohet më tej në llogaritjet e distancave të panjohura. Llogaritjen e madhësisë Koeficienti_Sys_Lazer_Kamera mund ta bëjmë bazuar në një matje kalibruese të vetme (siç jepet në kodet e programit) :

```
Private Sub cmdKalibrim_Click()
   Dim DistPikeLazerNgaQendra  As Integer
   Set Image2.Picture = dspPreview(1).Snap
   DistPikeLazerNgaQendra = DistanceKalibrimiPrejQendres(Image2)
   Koeficienti_Sys_Lazer_Kamera = DistPikeLazerNgaQendra * CDbl(txtDist.Text)
End Sub

Private Function DistanceKalibrimiPrejQendres(ByVal picColor As PictureBox) As Integer
   Dim PozicionLazerit As LAZER_DISTANCE
   Const pixR As Integer = 3
   Const pixG As Integer = 2
   Const pixB As Integer = 1
   Dim bitmap_info                 As BITMAPINFO
   Dim pixels()                    As Byte
   Dim bytes_per_scanLine          As Long
   Dim pad_per_scanLine            As Integer
   Dim X                           As Integer
   Dim Y                           As Integer
   Dim lartesia, gjeresia          As Integer
   Dim rreshti                     As Integer
   Dim kolona                      As Integer
   Dim max_rreshti                 As Integer
   Dim max_kolona                  As Integer
   Dim max_rreshtiR                As Integer

   Dim koeficient_radian_pixel             As Double
   Dim koeficient_kompesimi_gabimi         As Double


   ' Inicializim per bitmap_info bitmap .
   With bitmap_info.bmiHeader
      .biSize = 40
```

```
        .biWidth = picColor.ScaleWidth
        .biHeight = -picColor.ScaleHeight
        .biPlanes = 1
        .biBitCount = 32
        .biCompression = BI_RGB
        bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \ 32) * 4)
        pad_per_scanLine = bytes_per_scanLine - (((.biWidth * .biBitCount) + 7) \ 8)
        .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
    End With

    max_rreshtiR = 0

    ' Ngarkim bitmap datash
    ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To picColor.ScaleHeight)
    GetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info,
DIB_RGB_COLORS

    lartesia = picColor.Height / Screen.TwipsPerPixelY
    gjeresia = picColor.Width / Screen.TwipsPerPixelX

    ' Kodet per procesimin e imazhit
    For rreshti = 1 To lartesia - 1 Step 10
        For kolona = 1 To gjeresia - 1 Step 10
            If pixels(pixR, kolona, rreshti) > max_rreshtiR Then
                max_rreshtiR = pixels(pixR, kolona, rreshti)        ‘ Variabli që mban vlerën më të
                max_rreshti = rreshti                               ‘ madhe te matrices pixels(3,..,..)
                max_kolona = kolona                                 ‘ duke siguruan gjetjen e koordinatave
            End If                                                  ‘ te pikes lazer ne image.
        Next
    Next

    ' Llogarit distancen prej pikes lazer deri ne qender te pamjes
    Dim x_gender, y_gender

    x_gender = CInt(gjeresia / 2)
    y_gender = CInt(lartesia / 2)

    DistanceKalibrimiPrejQendres = Sqr((y_gender - max_rreshti) ^ 2 + (x_gender - max_kolona) ^
2)
    End Function
```

Llogaritjen e madhësisë Koeficienti_Sys_Lazer_Kamera mund ta bejmë edhe disa herë për disa kalibrime të kryera për distanca të ndryshme, duke përfituar disa vlera për madhësitë :

Koeficienti_Sys_Lazer_Kamera1
Koeficienti_Sys_Lazer_Kamera2
...
Koeficienti_Sys_Lazer_KameraN

Më tej mund të llogaritim një vlerë mesatare për Koeficienti_Sys_Lazer_Kamera, nisur nga formula llogaritëse për mesataren aritmetike. Një rrugë e tretë është llogaritja e madhësive rpc dhe r_0 sipas vlerësimeve të bëra në paragrafin 3.2.

4. Pas përfundimit të kalibrimit, procesi i vlerësimit të distancës kryhet automatikisht në çdo interval kohor prej 500 ms. Imazhi i marrë prej kamerës që do të përpunohet kalohet në objektin Image2 prej instruksionit Set Image2.Picture = dspPreview(1).Snap. Objekti dspPreview është ai që lidhet drejpërdrejt me paisjen e kamerës duke na përcjellë imazhet e saj. Subrutina Timer1_Timer siguron thirrjen e përsëritur të funksionit GjetjeDistancePrejObjektit që shërben për të llogaritur distancën e kërkuar.

```
Private Sub Timer1_Timer()
   Dim DistLazer As LAZER_DISTANCE

   ' Set Image1.Picture = dspPreview(0).Snap
   ' DistLazer = GjetjeDistancePrejObjektit(Image1)
  '  valRreshti(0).Caption = DistLazer.NrRreshtitLazerit
  '  valKolona(0).Caption = DistLazer.NrKolonesLazerit
  '  valDistanca(0).Caption = DistLazer.distanca

   Set Image2.Picture = dspPreview(1).Snap
   DistLazer = GjetjeDistancePrejObjektit(Image2)
   valRreshti(1).Caption = DistLazer.NrRreshtitLazerit
   valKolona(1).Caption = DistLazer.NrKolonesLazerit
   valDistanca(1).Caption = DistLazer.distanca
End Sub

Private Function GjetjeDistancePrejObjektit(ByVal picColor As PictureBox) As LAZER_DISTANCE
   Dim PozicionLazerit As LAZER_DISTANCE
   Const pixR As Integer = 3
   Const pixG As Integer = 2
   Const pixB As Integer = 1
   Dim bitmap_info             As BITMAPINFO
   Dim pixels()                As Byte
   Dim bytes_per_scanLine      As Long
   Dim pad_per_scanLine        As Integer
   Dim X                                   As Integer
   Dim Y                                   As Integer
   Dim lartesia, gjeresia                  As Integer
   Dim rreshti                 As Integer
   Dim kolona                  As Integer
   Dim max_rreshti                         As Integer
   Dim max_kolona                          As Integer
   Dim max_rreshtiR                        As Integer
   Dim koeficient_radian_pixel             As Double
   Dim koeficient_kompesimi_gabimi  As Double
   Dim pixels_prej_qendres                 As Double
   Dim distanca                As Double

   ' Prepare the bitmap description.
   With bitmap_info.bmiHeader
      .biSize = 40
      .biWidth = picColor.ScaleWidth
      ' Use negative height to scan top-down.
      .biHeight = -picColor.ScaleHeight
      .biPlanes = 1
      .biBitCount = 32
      .biCompression = BI_RGB
      bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \ 32) * 4)
      pad_per_scanLine = bytes_per_scanLine - (((.biWidth * .biBitCount) + 7) \ 8)
      .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
```

```
    End With

    max_rreshtiR = 0

    ' Load the bitmap's data.
    ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To picColor.ScaleHeight)
    GetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info,
DIB_RGB_COLORS

    lartesia = picColor.Height / Screen.TwipsPerPixelY
    gjeresia = picColor.Width / Screen.TwipsPerPixelX

    ' Kodet per procesimin e imazhit
    For rreshti = 1 To lartesia - 1 Step 10
        For kolona = 1 To gjeresia - 1 Step 10
            If pixels(pixR, kolona, rreshti) > max_rreshtiR Then
                max_rreshtiR = pixels(pixR, kolona, rreshti)        ' Variabli që mban vlerën më të
                max_rreshti = rreshti                                ' madhe te matrices pixels(3,..,..)
                max_kolona = kolona                                  ' duke siguruan gjetjen e koordinatave
            End If                                                   ' te pikes lazer ne image.
        Next
    Next

    max_rreshti = max_rreshti + 5
    max_kolona = max_kolona + 5

    ' llogaritja e distaces prej pikes lazer deri ne qendra e faqes se imazhit
    Dim x_gender, y_gender

    x_gender = CInt(gjeresia / 2)
    y_gender = CInt(lartesia / 2)

    pixels_prej_qendres = Sqr((y_gender - max_rreshti) ^ 2 + (x_gender - max_kolona) ^ 2)

    ' Llogaritja e distances ne baze te madhesive 'pixels_prej_qendres' dhe kalibrimit
'Koeficienti_Sys_Lazer_Kamera'
    distanca = Koeficienti_Sys_Lazer_Kamera / pixels_prej_qendres

    ' Percakto pozicionin e rreshtit dhe kolones se pikes lazer.
    PozicionLazerit.NrRreshtitLazerit = max_rreshti
    PozicionLazerit.NrKolonesLazerit = max_kolona

    ' Percakto distancen e bojektit nga kamera
    PozicionLazerit.distanca = distanca

    If max_kolona > 0 And rreshti > 0 Then

        ' Vizato nje vije te kuqe vertikale qe kalon ne piken lazer
        If max_rreshti > 50 Then
            For rreshti = max_rreshti - 50 To max_rreshti
                pixels(pixR, max_kolona, rreshti) = 255
            Next
        End If

        For rreshti = lartesia / 2 - 50 To lartesia / 2 + 50
            pixels(pixR, x_gender, rreshti) = 127
        Next

        ' Vizato nje vije te kuqe horizontale qe kalon ne piken lazer
```

```
        If max_kolona > 50 Then
           For kolona = max_kolona - 50 To max_kolona - 1
              pixels(pixR, kolona, max_rreshti) = 255
           Next
        End If

        For kolona = gjeresia / 2 - 50 To gjeresia / 2 + 50
           pixels(pixR, kolona, y_gender) = 127
        Next

        ' Shfaq imazhin e targuar.
        SetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info,
DIB_RGB_COLORS
        picColor.Picture = picColor.Image
     End If
     GjetjeDistancePrejObjektit = PozicionLazerit
End Function
```

Në Shtojcën-5 po japim programin për llogaritjen e distancës së sistemit kamera & lazer prej një objekti të caktuar.

3.4. Metoda e vlerësimit të distancës së objekteve nëpërmjet një sistemi prej dy kamerash

Modeli i pranuar për kamerën.

Në vlerësimin e distancës së objekteve nga një sistem prej dy kamerash, do të bazohemi në parimin e punës së këtyre aparateve . Ne jemi gjithashtu duke supozuar një model të përsosur lidhur me lenten, duke pranuar një lente të hollë . për të lehtësuar llogaritjen lidhur me imazhin dixhital të prodhuar nga kamera. Metoda që përdoret mbështetet në faktin që nje objekt me lartësi h_0 do të ketë imazhin e tij djathtas lentes ne madhësinë h_I . Kur objekti origjinal zmadhohet me madhësinë δ , imazhi i tij do të zmadhohet me $f(\delta)$. Prej Fig 3.3, shohim që : $h_0^{'} = h_0 + \delta$ dhe $h_I^{'} = h_I + f(\delta)$. Duke u mbështetur në një gjeometri të thjeshtë mund të shkruajmë : $\frac{h_0}{h_I} = \frac{R-f}{f}$ dhe $\frac{h_0^{'}}{h_I^{'}} = \frac{R-f}{f}$

Prej ekuacionit të fundit mund të shkruajmë :

$$h_I^{'} = h_0^{'} \cdot \frac{f}{R-f} = (h_0 + \delta) \cdot \frac{f}{R-f} = h_I + \delta \cdot \frac{f}{R-f} \qquad (1)$$

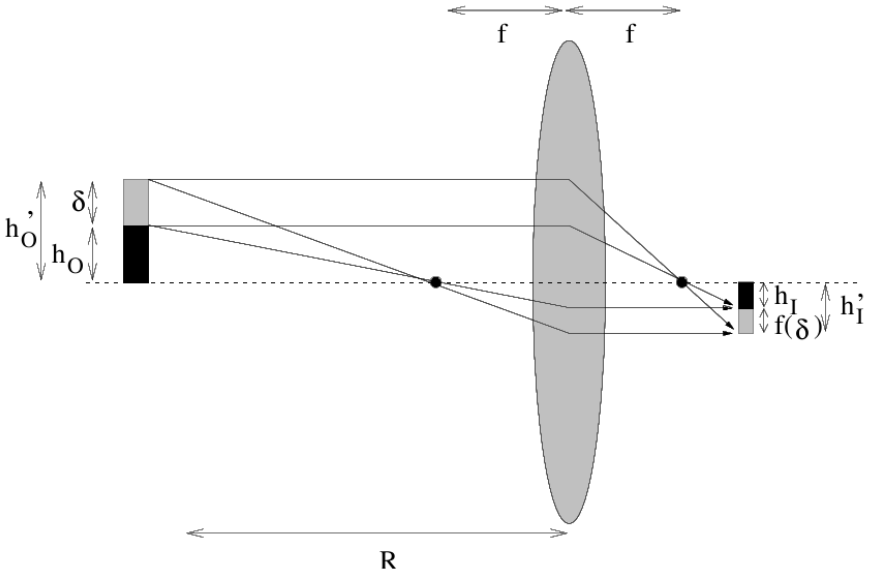


Fig 3.3. *Imazhi i objekteve prej një lenteje*

Duke mbajtur objektin në të njëjtën distancë, madhësia $\frac{f}{R-f}$ mbetet e pandryshueshme. Kjo do të thotë se varësia e ndryshimit të madhësisë së imazhit në funksion të ndryshimit të madhësisë së objektit është lineare. Nisur nga ky përfundim mund të themi se përcaktimi i distancës së objektit prej kameras do të mund të realizohet nëpërmjet konvertimit të numrit të pikave që numërohen në imazhin e objektit sipas një varësie lineare (shënojmë me ρ koeficientin konvertues). Në vlerësimin e distancës kërkohet njohja e disa madhësive ndërmjet të cilave është njohja e distacës midis dy kamerave si dhe këndi i pamjes së kamerave përcaktuar nga shprehja $\theta = \arctan\left(\frac{D}{2 \cdot f}\right)$ ku f largësia vatrore dhe D diametri i fushës së kameras. Formula e fundit na lejon të përcaktojmë këndin e pamjes së kamerës në plan.

Vlerësimi i distancës së objektit nga një sistem prej dy kamerash.

Për të vlerësuar distancën e objekteve nga një sistem kontrolli të përbërë nga dy kamera, algoritmi që do të përdoret mbështetet në vlerësimet e mësipërme. Objekti mund të jetë i pozicionuar në lidhje me kamerat sipas mundësive të dhëna në vijim :

- Ndërmjet të dy kamerave.
- Majtas ose djathtas ndaj të dy kamerave.
- Përballë kamerës së majtë ose të djathtë.

Për të gjitha këto pozicionime të objektit ajo që shihet është që objekti gjithmonë gjendet në zonën e përbashkët të të dy kamerave. Analiza për rastet kur objekti ndodhet jashtë kësaj zone përbën një tjetër rast që do të trajtohet në një material tjetër si punë e së ardhmes.

Vlerësimi i distancës së objektit të pozicionuar ndërmjet dy kamerave.

Duke u mbështetur në Fig 3.4, llogaritja e distancës së objektit (R) do të kryhet si më poshtë.

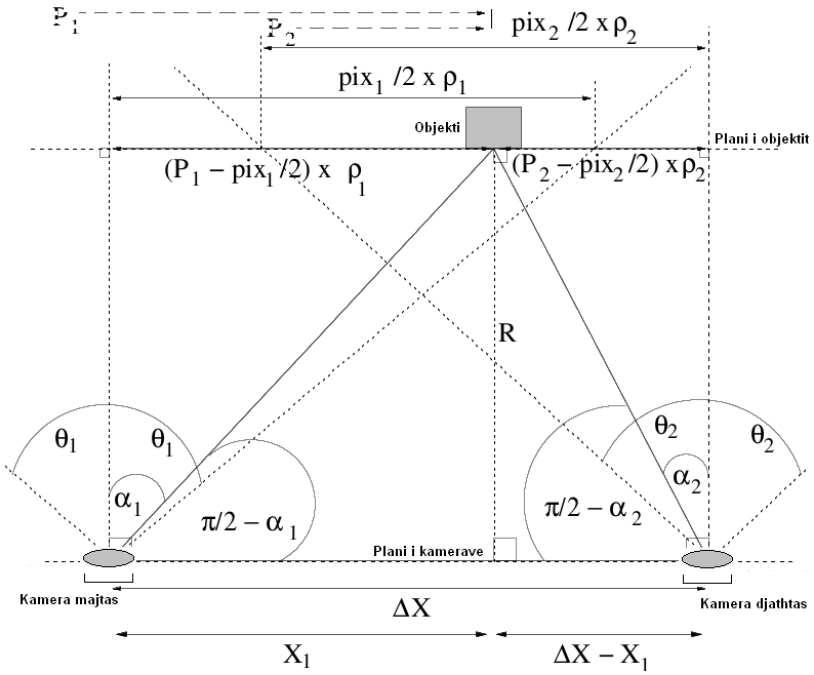


Fig 3.4. *Objekti i pozicionuar ndërmejet dy kamerave në zonën e përbashkët*

Përcaktimi i madhësive θ_1 dhe α_1 do të kryhet nga shprehjet :

$$\tan(\theta_1) = \frac{\frac{pix_1}{2} \cdot \rho_1}{R} \qquad \text{dhe} \qquad \tan(\alpha_1) = \frac{\left(P_1 - \frac{pix_1}{2}\right) \cdot \rho_1}{R} \qquad (2)$$

Në mënyrë të ngjashme do të përcaktohen madhësitë θ_2 dhe α_2.

$$\tan(\theta_2) = \frac{\frac{pix_2}{2} \cdot \rho_2}{R} \qquad \text{dhe} \qquad \tan(\alpha_2) = \frac{\left(\frac{pix_2}{2} - P_2\right) \cdot \rho_2}{R} \qquad (3)$$

ku :

pix_1 përcakton numrin total të pikave në të gjithë zonën që kontrollon kamera majtas,

ρ_1 koeficientin konvertues për kameran majtas

P_1 përcakton numrin total të pikave prej kufirit majtas deri në objekti (kamera majtas).

pix_2 përcakton numrin total të pikave në të gjithë zonën që kontrollon kamera djathtas,

ρ_2 koeficientin konvertues për kamerën majtas

P_2 përcakton numrin total të pikave prej kufirit majtas deri në objekti (kamera djathtas).

Prej ekuacioneve (2) dhe (3) përcaktojmë :

$$\alpha_1 = \arctan\left(\frac{P_1 - \frac{pix_1}{2}}{\frac{pix_1}{2}} \cdot \tan(\theta_1)\right) \qquad \text{dhe} \qquad \alpha_2 = \arctan\left(\frac{\frac{pix_2}{2} - P_2}{\frac{pix_2}{2}} \cdot \tan(\theta_2)\right) \qquad (4)$$

Prej fig 2, mund të shkruajmë :

për kamerën majtas : $\tan(\frac{\pi}{2} - \alpha_1) = \frac{R}{X_1}$ (5)

për kamerën djathtas : $\tan(\frac{\pi}{2} - \alpha_2) = \frac{R}{\Delta X - X_1}$ (6)

ku : ΔX përcakton distancën ndërmjet dy kamerave.

Duke u nisur nga ekuacionet (5) dhe (6) mund të përcaktojmë :

$$\frac{\tan(\frac{\pi}{2} - \alpha_1)}{\tan(\frac{\pi}{2} - \alpha_2)} = \frac{\Delta X - X_1}{X_1} \qquad (7)$$

Duke e zgjidhur ekuacionin (7) sipas X_1 llogaritim :

$$X_1 = \frac{\tan(\frac{\pi}{2} - \alpha_2)}{\tan(\frac{\pi}{2} - \alpha_1) + \tan(\frac{\pi}{2} - \alpha_2)} \cdot \Delta X \qquad (8)$$

Përfundimisht mund të përcaktojmë distancën e kërkuar të objektit prej kamerave prej ekuacioneve (5) dhe (8).

$$R = \frac{\tan(\frac{\pi}{2} - \alpha_1) \cdot \tan(\frac{\pi}{2} - \alpha_2)}{\tan(\frac{\pi}{2} - \alpha_1) + \tan(\frac{\pi}{2} - \alpha_2)} \cdot \Delta X \qquad (9)$$

ose

$$R = \frac{1}{\tan(\alpha_1) + \tan(\alpha_2)} \cdot \Delta X \qquad (9')$$

Përfundimisht duke shfrytëzuar formulat (4) dhe (9) jemi në gjendje të përcaktojmë distancën e objektit nga kamerat kur njohim madhësitë $\alpha_1, \alpha_2, \Delta X$.

Vlerësimi i distancës së objektit të pozicionuar majtas ose djathtas ndaj të dy kamerave.

Në të njëjtën mënyrë si u trajtua rasti i mësipërm do të trajtohet edhe rasti i dhënë në Fig 3.5.

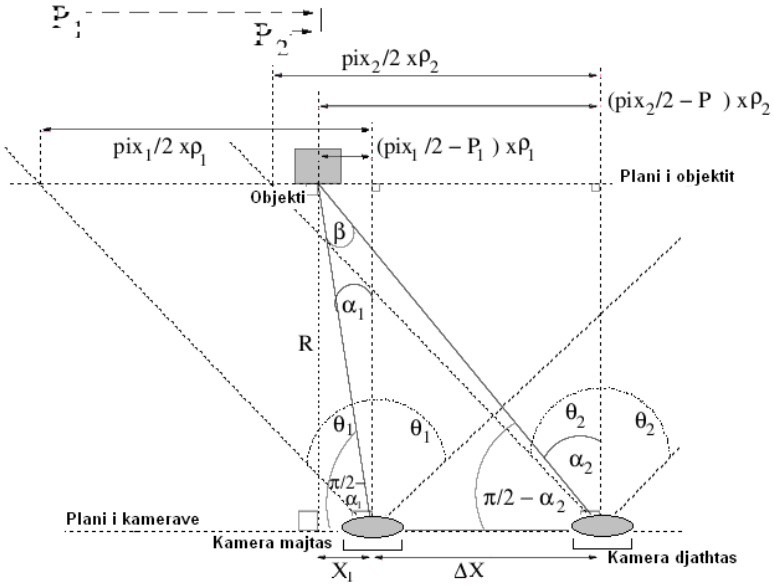


Fig 3.5. *Objekti i pozicionuar majtas të dy kamerave në zonën e përbashkët*

Këndi α_1, α_2 do të përcaktohen nga shprehjet e mëposhtëme :

$$\alpha_1 = \arctan\left(\frac{\frac{pix_1}{2} - P_1}{\frac{pix_1}{2}} \cdot \tan(\theta_1)\right) \qquad \text{dhe} \qquad \alpha_2 = \arctan\left(\frac{\frac{pix_2}{2} - P_2}{\frac{pix_2}{2}} \cdot \tan(\theta_2)\right) \qquad (10)$$

Nga fig 3 mund të përcaktojmë : $\beta = \pi - \left[\left(\alpha_1 + \frac{\pi}{2}\right) + \left(\frac{\pi}{2} - \alpha_2\right)\right] = \alpha_2 - \alpha_1$

Prej fig 3, mund të shkruajmë :

$$\tan(\frac{\pi}{2} - \alpha_1) = \frac{R}{X_1} \qquad (11) \qquad\qquad \tan(\frac{\pi}{2} - \alpha_2) = \frac{R}{\Delta X + X_1} \qquad (12)$$

ku : ΔX përcakton distancën ndërmjet dy kamerave.

Prej ekuacioneve (11) dhe (12) nxjerrim : $X_1 = \dfrac{\tan(\frac{\pi}{2} - \alpha_2)}{\tan(\frac{\pi}{2} - \alpha_1) - \tan(\frac{\pi}{2} - \alpha_2)} \cdot \Delta X \qquad (13)$

Përfundimisht mund të përcaktojmë distancën e kërkuar të objektit prej kamerave prej ekuacioneve (11) dhe (13).

$$R = \frac{\tan(\frac{\pi}{2} - \alpha_1) \cdot \tan(\frac{\pi}{2} - \alpha_2)}{\tan(\frac{\pi}{2} - \alpha_1) - \tan(\frac{\pi}{2} - \alpha_2)} \cdot \Delta X \qquad (14)$$

ose

$$R = \frac{1}{\tan(\alpha_2) - \tan(\alpha_1)} \cdot \Delta X \qquad (14')$$

Përcaktimi i distancës për rastin kur objekti ndodhet në të djathtë të kameras së djathtë, do të jetë sipas ekuacionit :

$$R = \frac{\tan(\frac{\pi}{2} - \alpha_1) \cdot \tan(\frac{\pi}{2} - \alpha_2)}{\tan(\frac{\pi}{2} - \alpha_2) - \tan(\frac{\pi}{2} - \alpha_1)} \cdot \Delta X \qquad (15)$$

ose

$$R = \frac{1}{\tan(\alpha_1) - \tan(\alpha_2)} \cdot \Delta X \qquad (15')$$

Vlerësimi i distancës së objektit të pozicionuar përballë kameras të majtë ose të djathtë.

Në Fig 3.6 jepet objekti i pozicionuar përballë me kameran e majtë në zonën e përbashkët.

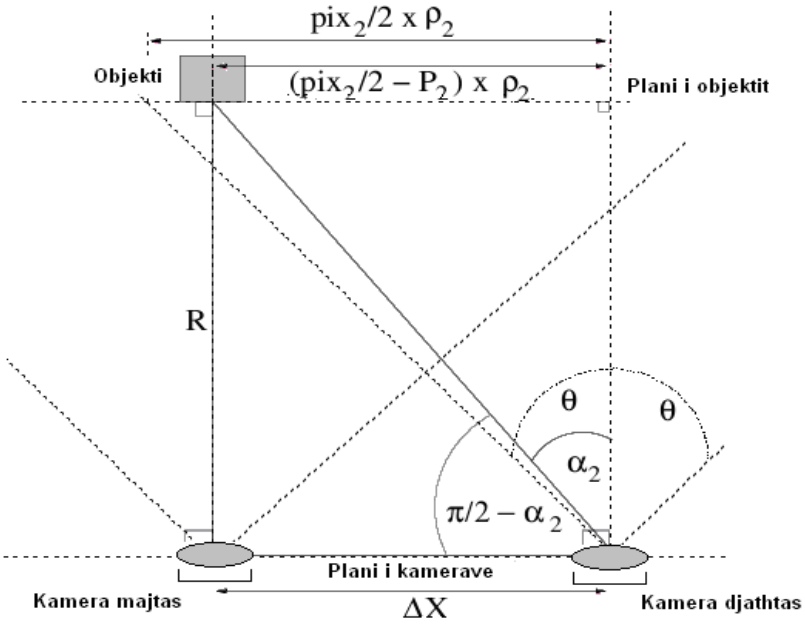


Fig 3.6. *Objekti i pozicionuar përballë me kamerën e majtë në zonën e përbashkët*

Për rastin kur objekti ndodhet përballë me kamerën e majtë ($\alpha_1 = 0$), llogaritjet e distancës do të bëhen si vijon :

$$R = \Delta X \cdot \tan(\frac{\pi}{2} - \alpha_2) \qquad (16)$$

Në mënyrë të ngjashme llogaritet distanca kur objekti ndodhet përballë me kamerën e djathtë ($\alpha_2 = 0$).

$$R = \Delta X \cdot \tan(\frac{\pi}{2} - \alpha_1) \qquad (17)$$

Realizimi eksperimental i vlerësimit të distancës së objektit prej planit të kamerave.

Në vlerësimin eksperimental të distancës së objektit nga plani i kamerave, do të na nevojitet një çift kamerash si në Fig 3.7, për të cilat është e domosdoshme të njihen parametrat si largësia vatrore f dhe diametri i lentes D. Objekti për të cilin do të bëhen vlerësimet e distancës do të jetë i dallueshëm nga mjedisi në sfond. Për të realizuar dallueshmërinë midis objektit dhe pjesës tjetër në sfond është e nevojshme që objekti të ketë një ngjyrë të vecantë nga pjesa tjetër e mjedisit.
Programi që realizon vlerësimin e distancës së objektit nga plani i kamerave është ndërtuar në disa mjedise programimi, Matlab R2013b, Visual Basic .Net dheVisual Basic 6.0.

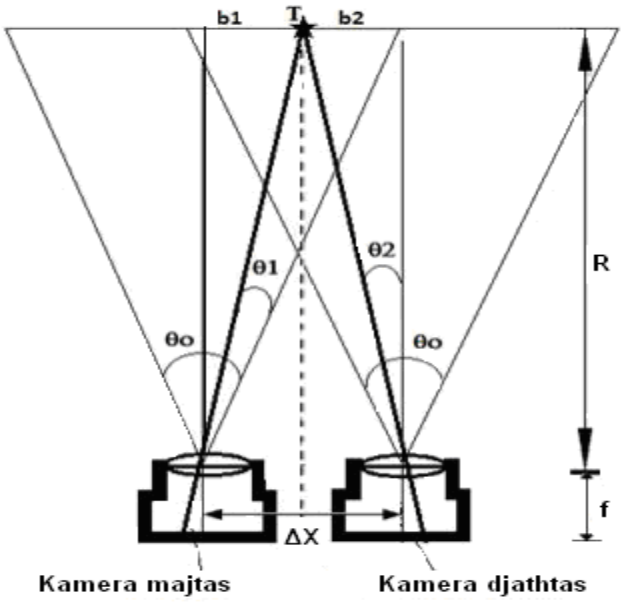


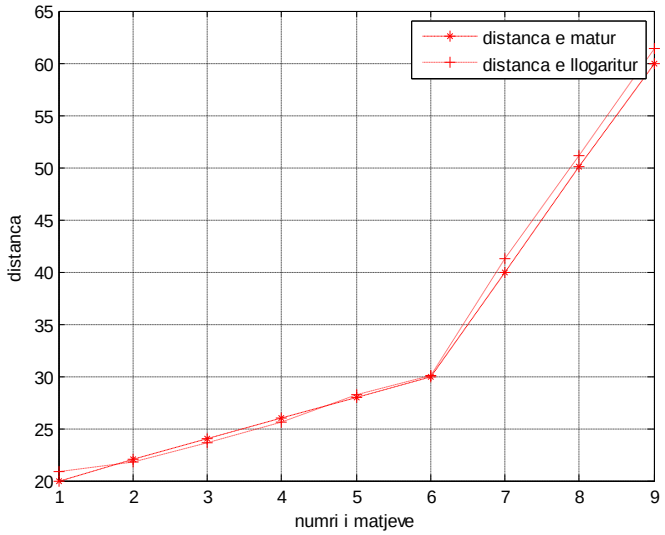
Fig 3.7. *Skema eksperimentale*

Rezultatet e matjeve të kryera jepen në tabelën e mëposhtëme. Realizimi i tyre është kryer për distanca të njohura të planit të kamerave nga plani i objektit.

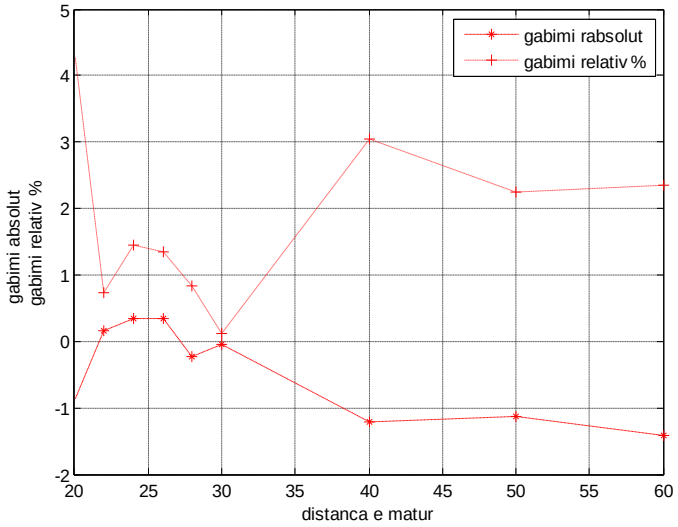
Kamerat të përdorura janë PcCamera HAMY c-200 Wireless Camera me këto të dhëna : $D = 1cm$, $f = 3cm, \Delta X = 11cm$

Nr	Distanca e matur (në cm)	Distanca e llogaritur (në cm)	Shmangia (në cm)	Gabimi relativ në % (vlera relative e tyre)
1	20	20.8545	-0,8545	4,2725
2	22	21.8396	0,1604	0,7291

3	24	23.6540	0,3460	1,4417
4	26	25.6523	0,3477	1,3373
5	28	28.2352	-0,2352	0,8400
6	30	30.0365	-0,0365	0,1217
7	40	41.2145	-1,2145	3,0363
8	50	51.1256	-1,1256	2,2512
9	60	61.4125	-1,4125	2,3542



Grafiku 3.8. *Distancat e matura dhe të llogaritura*



Grafiku 3.9. *Gabimet absolute dhe relative në varësi të distancës së matur*

Llogaritjet e kryera janë për rastin kur objekti ndodhet në pozicionin përballë kamerave, në mes të distancës ndërmjet dy kamerave.

Si përfundim mund të themi se :

- Nëpërmjet kësaj metode tregohet se si mund të vlerësohet distanca e një objekti prej një cifti kamerash.
- Metoda siguron rezultate të sakta për sa kohë njihen parametrat e kamerave si largësia vatrore f dhe diametri i lentes D.
- Kjo metodë përdoret vetem në rastin kur sistemi shihet në plan dhe jo për rastin 3D.
- Nisur nga tabela dhe grafikët e mësipërm mund të themi se kur distanca me objektin zvogëlohet, gabimi në vlerësim rritet (vlera 20 cm me gabim relativ 4.27%). Nga tabelat e rezultateve shihet edhe që me rritjen e distances midis kamerave dhe objektit rritet dhe masa e gabimit ne vlerësimin e distancës (vlerat per distancën 50 dhe 60 cm në tabelë).

3.5. Algoritmika dhe programet e realizuara Visual Studio 6 (VB) dhe Matlab R2013

Visual Basic 6.0

Më poshtë po japim një funksion llogaritës të distances dhe disa pamje prej programit të realizuar në Visual Basic 6.0

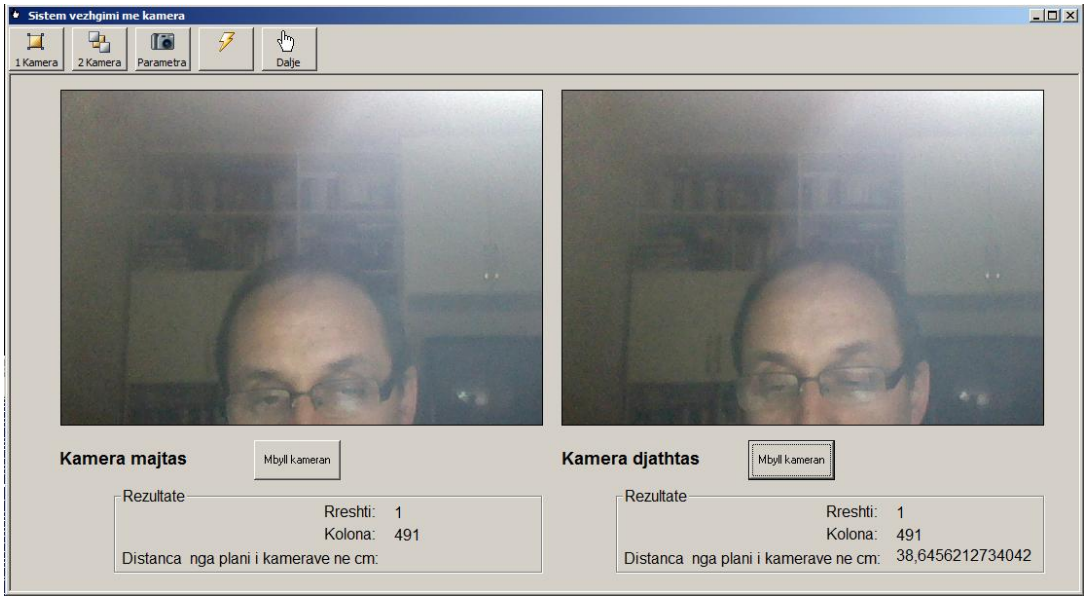


Fig 3.10. *GUI-ja në Visual Basic 6.0 për llogaritjen e distancës*

```
Public Function VleresoDistancenObjektit(PLeft As Double, PRight As Double, imageSize As Long) As Double
   Const PI As Double = 3.14159265358979
   Dim teta1, teta2              As Single
   Dim alpha1, alpha2   As Single
   Dim numurues          As Double
   Dim emerues           As Double
   Dim object_distance As Double

   ' llogaritje e kendeve teta1, teta2
   teta1 = Atn(gDiametriMajtas / (gDistVatroreMajtas * 2))              ' gjysem kendi i pamjes se kameras se majte
```

```
   teta2 = Atn(gDiametriDjathtas / (gDistVatroreDjathtas * 2))          ' gjysem kendi i pamjes se kameras se djathte

   ' objekti gjendet midis lenteve te majte dhe te djathte
   If ((PLeft > (imageSize / 2)) And (PRight < imageSize / 2)) Then
      'Rasti i pare
      alpha1 = Atn(((PLeft - (imageSize / 2)) / (imageSize / 2)) * Tan(teta1))
      alpha2 = Atn((((imageSize / 2) - PRight) / (imageSize / 2)) * Tan(teta2))
      numurues = DistancaNdermjet2Kamerave
      emerues = Tan(alpha1) + Tan(alpha2)
      object_distance = numurues / emerues
   End If

   If ((PLeft > (imageSize / 2)) And (PRight > (imageSize / 2)) Or (PLeft < (imageSize / 2)) And (PRight < (imageSize / 2))) Then
          alpha1 = Atn((((imageSize / 2) - PLeft) / (imageSize / 2)) * Tan(teta1))
          alpha2 = Atn(((PRight - (imageSize / 2)) / (imageSize / 2)) * Tan(teta2))
          If ((PLeft > (imageSize / 2)) And (PRight > (imageSize / 2))) Then
              object_distance = DistancaNdermjet2Kamerave / (Tan(alpha2) - Tan(alpha1))
          Else
              object_distance = DistancaNdermjet2Kamerave / (Tan(alpha1) - Tan(alpha2))
          End If
   End If

   If (PLeft = (imageSize / 2)) Then
      ' objekti perballe me lenten majtas
      alpha2 = Atn((((imageSize / 2) - PRight) / (imageSize / 2)) * Tan(teta2))
      object_distance = Tan((PI / 2) - alpha2) * DistancaNdermjet2Kamerave
   End If

   If (PRight = (imageSize / 2)) Then
     ' objekti perballe me lenten djathtas
      alpha1 = Atn(((PLeft - (imageSize / 2)) / (imageSize / 2)) * Tan(teta1))
      object_distance = Tan((PI / 2) - alpha1) * DistancaNdermjet2Kamerave
   End If

   VleresoDistancenObjektit = object_distance
End Function
```

Fig 3.11. *GUI-ja në Visual Basic 6.0 gjate llogaritjes së distancës*

Fig 3.12. *GUI-ja në Visual Basic 6.0, rezultatet*

Në Shtojcën-5 po japim programin për llogaritjen e distancës së sistemit të dy kamerave prej një objekti të caktuar.

Matlab R2013b

Për të realizuar gjetjen e distancës së një objekti të shënjestruar nëpërmjet dy kamerave në Matlab R2013b, do të shfrytëzoj programin që po e shpjegojmë në vijim.
Programi përbëhet nga dy module që emërtohen :

- Distaca2Kamera.m që përbman funksione dhe subrutina
- Distaca2Kamera.fig që përbën ndërfaqësin grafik për përdoruesin ose e njohur si GUI.

Funksionet e ndryshme që janë përdorur në këtë program po i përshkruajmë më poshtë :

`Function Distaca2Kamera_OpeningFcn(hObject, eventdata, handles, varargin)`
Ky funksion ekzekutohet para se dritarja e `Distaca2Kamera` të jetë bërë e dukshme. Ky funksion nuk rikthen ndonjë madhësi pas ekzekutimit. Gjatë ekzekutimit të këtij funksioni krijohen, inicializohen dhe startohen objektet për imazhet video të të dy kamerave, si psh për kamerën 1 instruksionet janë si më poshtë :

```
vid1 = videoinput('winvideo', 1);
% only capture one frame per trigger, we are not recording a video
vid1.FramesPerTrigger = 1;
% output would image in RGB color space
vid1.ReturnedColorspace = 'rgb';
% tell matlab to start the webcam on user request, not automatically
triggerconfig(vid1, 'manual');
% we need this to know the image height and width
vidRes = get(vid1, 'VideoResolution');
% image width
imWidth = vidRes(1);
% image height
imHeight = vidRes(2);
% number of bands of our image (should be 3 because it's RGB)
nBands = get(vid1, 'NumberOfBands');
% create an empty image container and show it on axPreview
hImageL = image(zeros(imHeight, imWidth, nBands), 'parent', handles.axPreview1);
% begin the webcam preview
preview(vid1, hImageL);
```

Pas përcaktimit të parametrave të kamerave si :

- largësia e vatrave ()
  ```
  focal_left = str2double(get(handles.edit1,'String'));
  focal_right = str2double(get(handles.edit3,'String'));
  ```
- diametri i lenteve të kamerave
  ```
  diam_left = str2double(get(handles.edit2,'String'));
  diam_right = str2double(get(handles.edit4,'String'));
  ```
- distanca midis kamerave
  ```
  cam_distance = str2double(get(handles.edit5,'String'));
  ```

thirret për tu ekzekutuar funksioni kryesor i cili kryen llogaritjet për distancën.

`Function pushbutton1_Callback(hObject, eventdata, handles)`
Ky funksion ekzekutohet pas shtypjes së butonit `LLOGARITJE`. Ky funksion siguron marrjen e imazheve nga të dy kamerat, ruajtjen e imazhit me emrat `'imgleft.jpg'` dhe `'imgright.jpg'`. Instruksionet që kryejnë këtë proces jepen më poshtë :

```
left_image_file ='imgleft.jpg';
FLeft = getframe(handles.axPreview1);
% image(FLeft.cdata);
ImageL = frame2im(FLeft);
% imwrite(FLeft.cdata, left_image_file);
imwrite(ImageL, left_image_file);
focal_left = str2double(get(handles.edit1,'String'));


right_image_file ='imgright.jpg';
FRight = getframe(handles.axPreview2);
% image(FRight.cdata);
ImageR = frame2im(FLeft);
% imwrite(FRight.cdata, right_image_file);
imwrite(ImageR, right_image_file);
focal_right = str2double(get(handles.edit3,'String'));
```

Imazhet e ruajtura thirren dhe më tej kryhet procesi i analizës për përcaktimin e distancës së një objekti, i cili zgjidhet në mënyrë manuale nga përdoruesi i programit nëpërmjet zgjedhjes me mouse të tij. Rezultati llogaritet, bazuar në teknikën e përshkruar në paragrafin e mësipërm, prej instruksioneve të mëposhtëme :

```
%Vlerat e qendres se objektit ne imazhin e majte dhe te djathte
PLeft = (object_left(1)+object_left(1)+object_left(3))/2
PRight = (xbegin + xend)/2

% llogaritja e distances dhe kthimi I vleres
teta1 = atan(diam_left/(focal_left*2)) % gjysem kendi i pamjes se kameres se majte
teta2 = atan(diam_right/(focal_right*2)) % gjysem kendi i pamjes se kameres se
djathte

% objekti eshte midis lentes se najte dhe te dhjathte
if ((PLeft  > (imageSize/2))&&(PRight < (imageSize/2)))
    disp('CASE 1!!!');
    alpha1 = atan(((PLeft - (imageSize/2))/(imageSize/2))*tan(teta1));
    alpha2 = atan((((imageSize/2) - PRight)/(imageSize/2))*tan(teta2));
    mone = tan((pi/2) - alpha1)*tan((pi/2) - alpha2)*cam_distance;
    mehane = tan((pi/2) - alpha1) + tan((pi/2) - alpha2);
    object_distance = mone /mehane
end
if ((PLeft  > (imageSize/2))&&(PRight > (imageSize/2))||(PLeft  <
(imageSize/2))&&(PRight < (imageSize/2)))
         alpha1 = atan((((imageSize/2) - PLeft)/(imageSize/2))*tan(teta1));
         alpha2 = atan(((PRight - (imageSize/2))/(imageSize/2))*tan(teta2));
         if ((PLeft  > (imageSize/2))&&(PRight > (imageSize/2)))
             % objekti eshte ne te majte te lentes se majte dhe te djathte
             disp('CASE 2!!!');
             object_distance = (sin((pi/2) - alpha1)*(sin((pi/2) -
alpha2))*cam_distance)/(sin(-alpha1 + alpha2))
         else
              %objekti eshte ne te djathte te lentes se majte dhe te djathte
             disp('CASE 3!!!');
             object_distance = (sin((pi/2) - alpha1)*(sin((pi/2) -
alpha2))*cam_distance)/(sin(-alpha2 + alpha1))
         end
end
if  (PLeft  == (imageSize/2))
    % objekti eshte perpara kameres se majte
    disp('CASE 4!!!\n');
    alpha2 = atan((((imageSize/2) - PRight)/(imageSize/2))*tan(teta2));
    object_distance = tan((pi/2) - alpha2)*cam_distance
end
if (PRight == (imageSize/2))
    %objekti eshte perpara kameres se djathte
    disp('CASE 5!!!\n');
```

```
        alpha1 = atan(((PLeft - (imageSize/2))/(imageSize/2))*tan(teta1));
        object_distance = tan((pi/2) - alpha1)*cam_distance
    end
```

`Function figure1_CloseRequestFcn(hObject, eventdata, handles)`
Ky funksion ekzekutohet kur aktivizohet mbyllja e GUI.

Në Shtojcën-6 po japim programin në MatLab R2013b për llogaritjen e distancës së sistemit të dy kamerave prej një objekti të caktuar.

3.6. Metoda e vlerësimit të distancës së objekteve prej një kamere referuar vlerësimit të thellësisë (Depth Estimation).

Në këtë paragraf do të trajtojmë një teknikë për vlerësimin e thellësisë duke përdorur kameran dhe më tej përpunimin e imazhit. Sistemi është i ndërtuar duke shfytëzuar parametrat specifikë të kamerave. Sisteme ndihmëse në fushën e drejtimit të automjeteve si Lane Departure warning (LDW), Dynamic High Beam (DHB), Forward Collision Warning (FCW) etj, përdorin kamera monoculare dhe aplikacione të ndërtuara për ato. Përdorimi i kamerave monoculare dhe aplikacioneve në sistemet e drejtimit te automjeteve, sot ka marrë një përdorim të gjerë. Shumë sisteme ekzistuese automjetesh (Mercedes E Class, BMW 7 series etj) kanë implementuar në një platformë të vetme të gjitha sistemet ndihmese të përshkruara më lart, duke reduktuar në maksimum kostot e tyre. Vlerësimi i thellësisë (pra i distancës së objektit nga plani i kamerës) është një detyrë e rëndësishme në aplikime të tilla si shmangia nga përplasjet, autoparkimet etj. Përdorimi i një kamere të vetme në këto raste është sfiduese sepse imazhi i kamerës është subjekt i shtrembërimeve të prespektivës. Në këto kushte vlerësimi i thellësisë mund të arrihet duke shfytëzuar vetitë fizike të kamerave si dhe duke përdorur analizat gjeometrike, të cilat bëjnë të mundur lidhjen e imazheve 2D të marra nga kamerat me ato 3D të mjedisit real [35]. Përpunimi i imazhit duke përdorur Matlab-in është përdorur tashmë në disa aplikime industriale. Me kamera të reja portative të përdorura gjerësisht sot është e mundur për të zhvilluar teknologji të kontrollit automatik të bazuara në përpunimin e imazhit në Matlab me besueshmëri të lartë të matjes [34]. Qëllimi kryesor i këtij paragrafi është të vlerësojë performancën e përpunimit të imazhit në Matlab duke punuar me një kamera, me qëllimin e matjes së distancës. Realizimi i kësaj detyre kërkon plotësimin e disa kushteve ku më kryesorët janë fakti që objekti që vëzhgohet duhet të jetë më i ndriçuar se pjesa tjetër në mjedisin rreth tij dhe mundësisht qëndra gjeometrike e objektit të jetë sa më afër aksit të kameres. Hapat që do të ndiqen deri në vlerësimin e distancës jepen në vijim :

- Merret imazhi prej kameres, ku brenda këtij imazhi ndodhet objekti që vëzhgohet (i ndriçuar).
- Objekti sillet në një distancë 20 cm larg prej planit të kamerës dhe pas përcaktimit të kontureve në imazhin e marrë nga kamera, llogaritet gjerësia w_k dhe lartësia h_k e tij. Kjo përbën dhe atë që njhet si procesi i kalibrimit për sistemin tonë.
- Gjatë ndryshimit të vendodhjes së objektit, llogaritja e distancës bëhet duke shfrytëzuar principet e gjeometrisë së prepektivës. Nisur nga llogaritjet e bëra në procesin e kalibrimit si dhe llogaritjet e gjerësisë w dhe lartësisë h së objektit në kushtet e një pozicionimi të ri, bëhet e mundur gjetja e distancës së re.
  Llogaritja e distancës kryhet nëpërmjet një formule empirike :

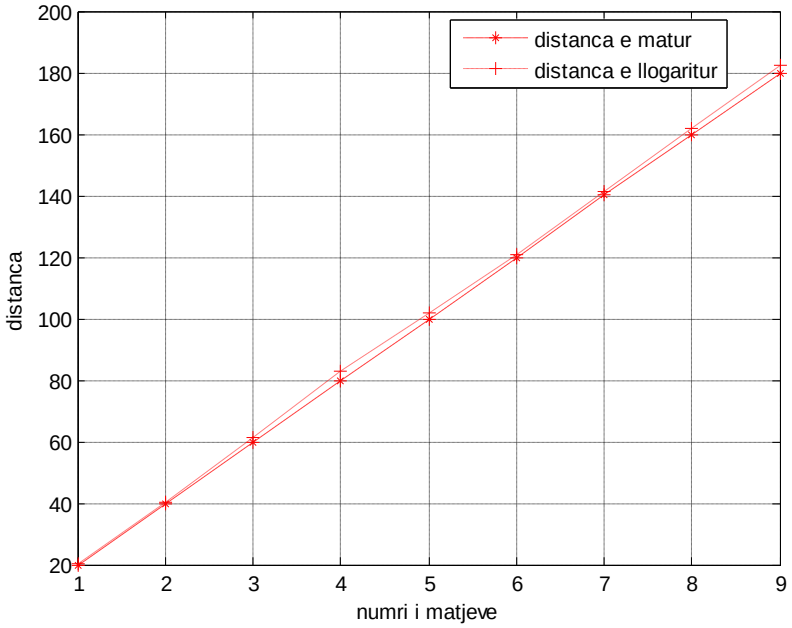
$$\text{Distanca} = -10.73 \cdot \log\left[\frac{w \cdot h}{w_k \cdot h_k} \cdot 21720\right] + 127.15$$

Rezultatet e matjeve të kryera jepen në tabelën e mëposhtme. Realizimi i tyre është kryer për distanca të njohura të planit të kamerave nga plani i objektit.

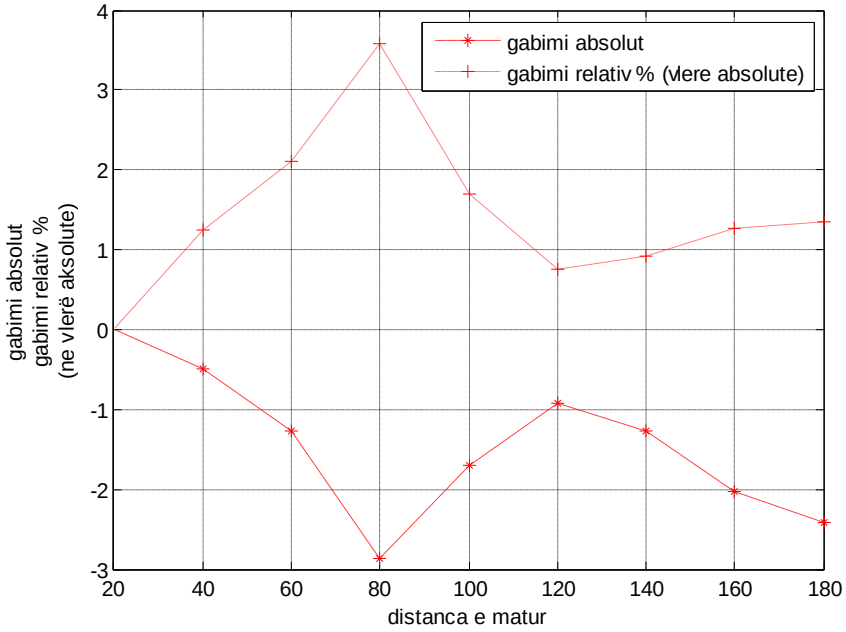
Kamerat e përdorura janë PcCamera HAMY c-200 Wireless Camera (të njëjta që janë përdorur si në metodat e përshkruara në paragrafet e mëparshëm), me këto të dhëna :

$$D = 1cm,\ f = 3cm, \Delta X = 11cm$$

Nr	Distanca e matur (në cm)	Distanca e llogaritur (në cm)	Shmangia (në cm)	Gabimi relativ në %
1	20	20.0035	-0.0035	0.0175
2	40	40.5006	-0.5006	1.2515
3	60	61.2640	-1.2640	2.1067
4	80	82.8723	-2.8723	3.5904
5	100	101.6982	-1.6982	1.6982
6	120	120.9165	-0.9165	0.7637
7	140	141.2765	-1.2765	0.9118
8	160	162.0316	-2.0316	1.2697
9	180	182.4126	-2.4126	1.3403



Grafiku 3.13. *Distancat e matura dhe të llogaritura*



Grafiku 3.14. *Gabimet absolute dhe relative në varësi të distancës së matur*

3.7. Algoritmika dhe programi i realizuar në Matlab R2013

Për të realizuar gjetjen e distancës së një objekti të shënjestruar nëpërmjet një kamere në Matlab R2013b, do të shfrytëzoj programin që po e shpjegojmë në vijim.
Programi përbëhet nga dy module që emërtohen :

- MatDistancenKamera.m që përbman funksione dhe subrutina
- MatDistancenKamera.fig që përbën ndërfaqësin grafik për përdoruesin ose e njohur si GUI.

Funksionet e ndryshme që janë përdorur në këtë program po i përshkruajmë më poshtë :

```
Function MatDistancenKamera_OpeningFcn(hObject, eventdata,
handles, varargin)
```

Ky funksion ekzekutohet para se dritarja e `MatDistancenKamera` të jetë bërë e dukshme. Funksioni nuk rikthen ndonjë madhësi pas ekzekutimit. Gjatë ekzekutimit të këtij funksioni krijohet dhe inicializohet objekti për imazhin video.

```
Function HapeKameran_Callback(hObject, eventdata, handles)
```

Ky funksion ekzekutohet pas shtypjes së butonin në `HapeKameran`. Funksioni siguron aktivizimin e objektit video dhe në GUI shfaqen imazhet e përcjella nga kamera.

```
Function Kalibrim_Callback(hObject, eventdata, handles)
```

Ky funksion ekzekutohet pas shtypjes së butonin `Kalibrim`. Funksioni siguron përcaktimin e madhësisë `factor` që do të shërbejë më tej për vlerësimin e distacave të objektit kundrejt planit të kameres.

```
Function [w, h, minc, minf] =
LlogaritGjeresiGjatesiObjektiPlan(video)
```
Ky funksion ekzekutohet gjthmonë kur kërkohet të vlerësohen koordinatat `minc, minf` si dhe gjerësia `w` dhe lartësia `h,` e objektit në vëzhgim.

```
Function [ ] = LlogaritDistancen(video,dist)
```
Ky funksion llogarit distancën e kërkuar për objektin në vëzhgim. Llogaritjet bëhen pas përcaktimit të koordinatave dhe përmasave të objektit për gjendjen e re, duke shfytëzuar shprehjen e mëposhtëme :

```
area = w*h;
distancia =( -10.73*log((area/factor)*21720)) + 127.15;
```

```
Function [ ] = DrawRect(video,dist)
```
Ky funksion shërben për të konturuar vizualisht objektin në GUI.

```
Function cameraGUI_CloseRequestFcn(hObject, eventdata, handles)
```
Ky funksion ekzekutohet kur aktivizohet mbyllja e GUI.

Në Shtojcën-7 po japim programin në MatLab R2013b, për llogaritjen e distancës së sistemit të kamerës prej objektit të caktuar.

Si përfundim mund të themi se :

- Nëpërmjet kësaj metode tregohet se si mund të vlerësohet distanca e një objekti prej një kamere të vetme duke shfrytëzuar principet e gjeometrisë së prespektivës.
- Metoda siguron rezultate të sakta edhe pa një njohje të parametrave të kamerës si largësia vatrore f dhe diametri i lentes D.
- Kjo metodë përdoret në rastin kur sistemi shihet në 3D sepse shfytëzon principet e dhëna më sipër
- Objekti duhet të jetë sa më i dukshëm ndaj kamerës që do të thotë se pozicioni më i mirë i tij do të qe në aksin që kalon nga qëndra e lentes së kamerës, pingul me planin e kamerës .
- Nga tabelat e rezultateve shihet se me rritjen e distancës midis kamerave dhe objektit rritet dhe masa e gabimit në vlerësimin e distancës (vlerat per distancën 120, 140, 160 dhe 180 cm në tabelë). Në matjet e dhëna gabimi relativ më i madh i marrë është në masën 3.6 %.

KAPITULLI IV

Hyrje

Në këtë kapitull do të përshkruhet një sistem i thjeshtë ndjekës objektesh nëpërmjet përpunimit të imazheve të sjella nga një kamer. Sistemi ndjekës i përshkruar dhe realizuar në këtë kapitull mund të shërbejë për ndjekjen e çfarëdo lloj objekti të cilësuar në një imazh të kameras, pra ai përbën një sistem ndjekës pothuajse universal. Sistemi ndjekës skematikisht i paraqitur në Fig.4.1, shpreh një mënyrë se si duke ndjekur lëvizjen e diellit, paneli diellor me fotoelementë pozicionohet sipas drejtimit ku sipërfaqja aktive për fluksin e dritës merr vlerën maksimale.

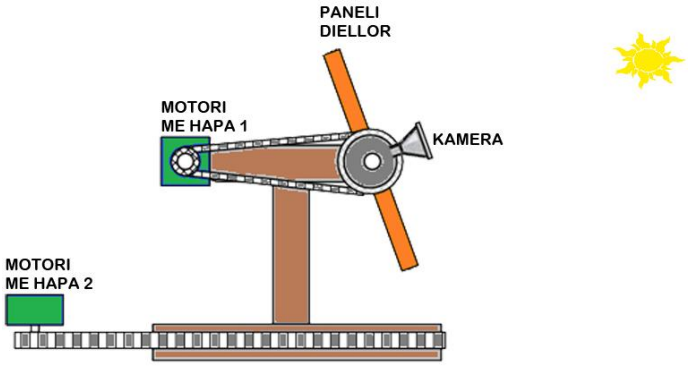


Fig. 4.1. *Skema pricipiale e sistemit ndjekës së objekteve me kamera*

Rrjedhojë e këtij pozicionimi të panelit dhe rryma e gjeneruar ka vlerën maksimale. Parimisht sistemi është i përbërë nga dy motorë me hapa të cilët sigurojnë lëvizjen në dy plane të panelit diellor duke siguruar mundësinë e ndjekjes së diellit. Në vazhdim po japim disa mënyra të realizimit të skemës së komandimit të motorëve me hapa, komanda të cilat janë rezultat i përpunimit të imazheve të mara nga kamera.

4.1. Sisteme të thjeshta robotike të komanduara nga kompjuteri, në ndjekjen sipas një, dy dhe tre drejtimeve të një objekti të paracaktuar të ndritshëm.

Në mënyrë të detajuar sistemi i plotë i ndjekjes së një objekti të ndritshëm jepet në Fig 4.1.1.

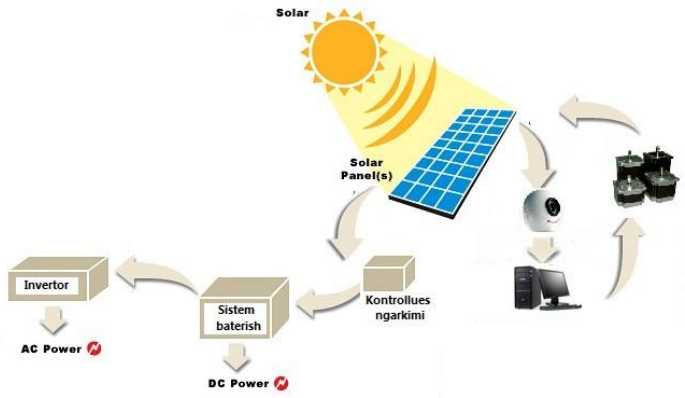
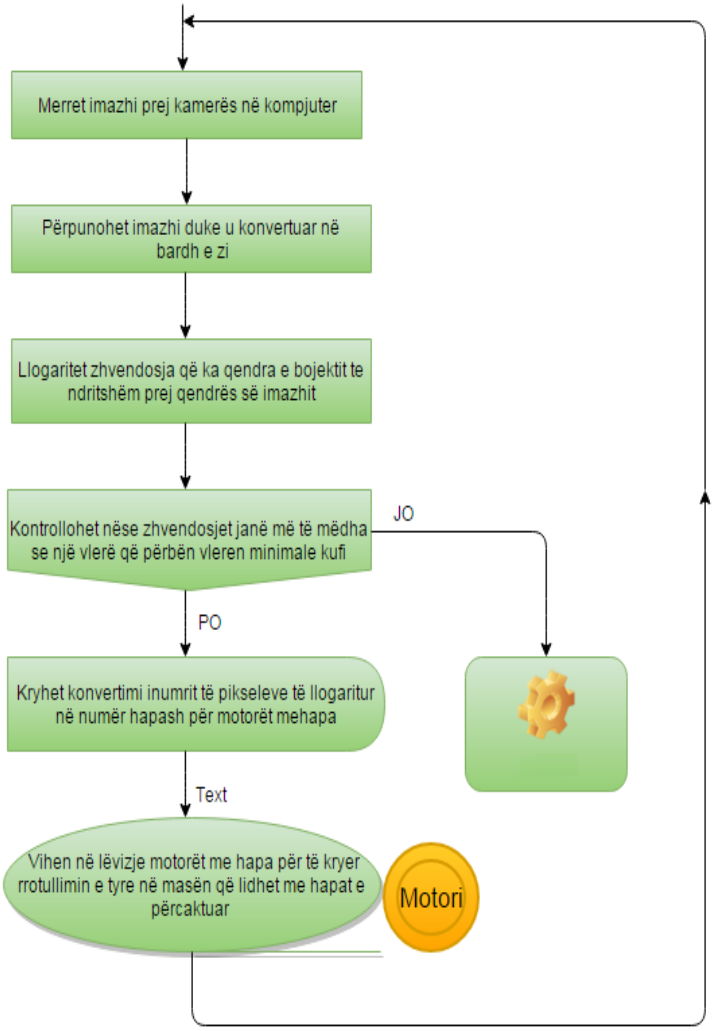


Fig. 4.1.1. *Sistemi i ndjekjes së objektit të paracaktuar të ndritshëm*

Elementët përbërës të sistemit që ndjek lëvizjen e diellit janë :

- Paisja kompjuterike.
- Moduli i kontrollit të motorit/motorëve me hapa.
- Kamera.
- Paisja e panelit diellor me sistemet mekeanike për lëvizjen rrotulluese të planit të panelit sipas një, dy dhe tre drejtimeve.
- Paisjet e kontrolluesit të nivelit të ngarkimit të baterive, sistemi i baterive, invertori etj.

Hapat kryesorë në realizimin e ndjekjes së objektit të ndritshëm do të paraqiteshin më poshtë nëpërmjet bllok diagramës :



Realizimi i ndjekjes së objektit të ndritshëm, i përshkruar në diagramën e mësipërme, kryhet sipas hapave të mëposhtëm :

- Prej kamerës merret imazhi në kompjuter.
- Përpunohet duke u konvertuar në bardh e zi (green scale).

- Kryhet identifikimi i pozicionimit që ka objekti i ndritshëm në imazh.
- Kryhet llogaritja e zhvendosjes që ka qëndra e objektit të ndritshëm prej qendrës së imazhit.
- Kontrollohet nëse zhvendosjet nga qëndra janë më të mëdha se një vlerë që përbën vlerën minimale për aksion lëvizjeje.
- Kryhet konvertimi i numrit të pikseleve të llogaritur nga pika 4 në numër hapash për motorët me hapa.
- Ekzekutohet komanda që ve në lëvizje motorët me hapa për të kryer rrotullimin e tyre në masën që lidhet me hapat e përcaktuar në pikën 6.
- Procesi ripërsëritet prej pikës 1, pas një intervali kohor.

Do të analizohen dy sisteme të thjeshta të komanduara nga kompjuteri, në ndjekje të një objekti të ndritshëm:

- Komandimi i panelit diellor prej kompjuterit nëpërmjet PIC18F4550-I/P Microchip Microcontroller & ULN2803.
- Komandimi i panelit diellor prej kompjuterit nëpërmjet modulit Stepper-Bee.

Në kapitullin III treguam funksionet që duhen përdorur për komandimin e motorit me hapa prej portës USB nëpërmjet PIC18F4550-I/P & ULN2803 si dhe modulit Stepper-Bee.
Programi që do të kontrollojë ndjekjen e objektit të lëvizshëm (diellit) duke orientuar panelin diellor të pozicionohet kundrejt diellit në mënyrë që energjia e gjeneruar prej fotoelementeve të jetë maksimale. Eshtë i qartë se pozicioni i panelit duhet të jetë i tillë që ai ta shohë 'ballë për ballë' diellin ose thënë ndryshe sipërfaqja e panelit dhe rrezet e diellit të formojnë këndin 90 gradë. Në këtë rast kamera e lidhur ngurtësisht me panelin diellor, me aks të saj perpendikular me planin e panelit do të ketë në qëndër të fokusit diellin. Në mënyrë që kamera të mos dëmtohet nga shkëlqimi diellor, lenta e saj lyhet me bojë të zezë ose mbi lente vendoset një xham i zi.
Gjatë lëvizjes së diellit, imazhi i ndritshëm i marrë nga kamera në një sfont të zi (realizuar nga lyerja me të zezë e lentes ose nga xhami i zi) do të zhvendoset prej pozicionit të tij fillestar. Si rezultat rrezet e diellit nuk do të bien pingul me planin e panelit diellor. Që ato të jenë pingul me këtë panel duhet që imazhi i ndritshëm i diellit të zhvendoset në qender të imazhit të marrë nga kamera. Kjo mund të bëhet vetëm duke realizuar rrotullimin e panelit diellor derisa imazhi i ndritshëm i diellit të zhvendoset në qendër të imazhit. Rrotullimi i panelit mund të bëhet nëpërmjet një motori me hapa, vetëm sipas një drejtimi duke ndjekur lëvizjen e diellit nga lindja drejt perëndimit. Në këtë rast komandimi i panelit djellor realizohet prej kompjuterit dhënë në Fig 4.1.3, nëpërmjet PIC18F4550-I/P Microchip Microcontroller & ULN2803.

Fig. 4.1.3 *Komandimi i panelit diellor prej kompjuterit nëpërmjet PIC18F4550-I/P Microchip Microcontroller & ULN2803*

Rrotullimi i panelit mund të bëhet edhe nëpërmjet dy motorëve me hapa. Në këtë rast lëvizja e diellit do të ndiqej në dy plane, sipas drejtimit nga lindja drejt perëndimit si dhe sipas drejtimit veri-jug.

Fig. 4.1.4 *Komandimi i panelit diellor prej kompjuterit nëpërmjet modulit Stepper-Bee*

Sistemi i ndjekjes së diellit gjatë lëvizjes së tij do të përshkruhet nëpërmjet skemës së dhënë në Fig. 4.1.5. Programi i ndërtuar në Visual Basic 6.0 realizon komandimin e sistemeve të paraqitura në Fig 4.1.3 dhe Fig 4.1.4. Pas identifikimit të pozicionit të imazhit të ndritshëm të diellit në imazhin e përgjithëshëm të kameras, llogaritet numri i pixeleve prej qendës së imazhit të diellit deri në qendrën e imazhit të kamerës. Në figurë këto madhësi përfaqësohen nga NrRreshtaPixelNgaQendraObjektit dhe NrKolonaPixelNgaQendraObjektit. Këto dy madhësi do të konvertohen në numër hapash për motorët me hapa dhe më tej do të jepen komandat që secili prej tyre të kryejë rrotullimin sipas hapave të llogaritur . Kjo do të sigurojë që objekti në ndjekje (dielli) do të zhvendoset drejt qendrës së imazhit të kamerës.

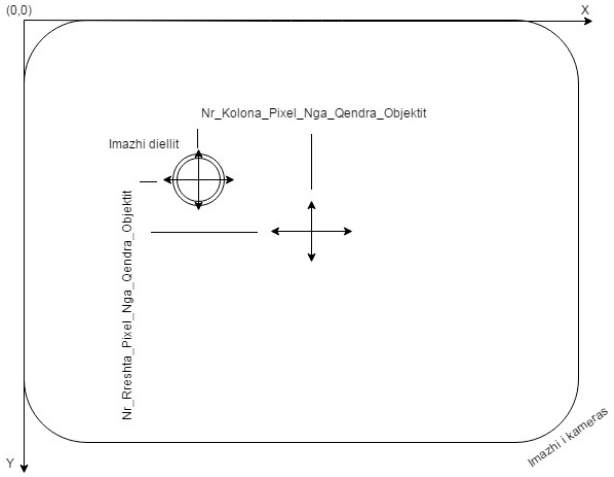


Fig. 4.1.5 *Skema e ndjekjes së djellit prej sistemit pas analizimit të imazhit të marrë nga kamera*

Programi GUI i ndërtuar për komandimin e lëvizjes së panelit sipas një aksi dhe dy akseve jepet në Fig 4.1.6. Aktivizimi fillestar i programit nis me shtypjen e butonit 'INICIALIZIME PER MOTORET' i cili krijon lidhjen me modulin PIC18F4550-I/P Microchip Microcontroller & ULN2803 apo me modulin Stepper-Bee. Programi ka dy mënyra funksionimi, komandimin manual dhe atë automatik të moduleve. Komandimi manual i modulit lejon sistemin të funksionojë në mënyrë manuale duke lejuar përdoruesin të veprojë për komandimin e motorëve A dhe B. Motorët marrin lëvizjet duke vepruar me shigjetat e treguara në GUI dhe ndalojnë pas shtypjes së butonave 'Stop'.
Në program komandat e vënies në lëvizje të motorëve jepen si më poshtë :

```
FilloLevizjenMotori 1, CInt(Abs(.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi) / txtInterval.Text), txtInterval.Text, 0, 1

FilloLevizjenMotori 2, CInt(Abs(.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) / txtInterval.Text), txtInterval.Text, 1, 1
```

Në GUI jepen të llogaritura koordinatat e qendrës së objektit në ndjekje, koordinatat e qendrës së imazhit si dhe distanca midis tyre në pixel.

Në program koordinatat jepen nëpërmjet kodeve si më poshtë :

```
LlogaritKordinataXY = GjejPozicioninObjektit(Picture1, txtStep.Text)

valRreshtiQenderObjekti.Caption = LlogaritKordinataXY.NrRreshtitQenderObjekti
valKolonaQenderObjekti.Caption = LlogaritKordinataXY.NrKolonesQenderObjekti

valRreshtiQenderImazhi.Caption = LlogaritKordinataXY.NrRreshtitQenderImazhi
valKolonaQenderImazhi.Caption = LlogaritKordinataXY.NrKolonesQenderImazhi

lDistanca_QenderObjekti_QenderImazhi.Caption = LlogaritKordinataXY.Distanca_QenderObjekti_QenderImazhi
```

Kur programi funksionon në komandim automatik të modulit, ai kryhen me rigorozitet të gjitha hapat e dhëna në fillim të këtij paragrafi. Një parametër i rëndësishëm në këtë moment do të jetë 'Gabimi i lejuar në pixel' i cili jep vlerën më të vogël të lejuar të differencës së koordinatave midis

qëndrave të objektit në ndjekje dhe imazhit, për të cilën motorët nuk vihen në lëvizje. Intervalet për çdo rillogaritje të koordinatave të objektit në ndjekje, përcaktohen nga një parametër tjetër i cili shprehet nëpërmejt një komponenti Timer në Visual Basic. Ai do të aktivizohet vetëm atëhere kur të ketë përfunduar lëvizja me hapa e motorëve duke rifilluar ciklin nga e para që me marrjen e imazhit të ri prej kamerës.

Në program komandimi i motorit jepet me kodet :

```
' Fillim i levizjes - Motor 1
Public Function FilloLevizjenMotori(ByVal NumberMotor As Integer, _
                    ByVal Steps As Integer, _
                    ByVal Interval As Integer, _
                    ByVal Direction As Integer, _
                    ByVal Outputs As Integer) As Boolean
  Select Case NumberMotor
  Case 1
    ' Startim i motorit 1
    FilloLevizjenMotori = RunMotor1(Steps, Interval, Direction, Outputs)
  Case 2
    ' Startim i motorit 2
    FilloLevizjenMotori = RunMotor2(Steps, Interval, Direction, Outputs)
  Case Else
  End Select
End Function
```

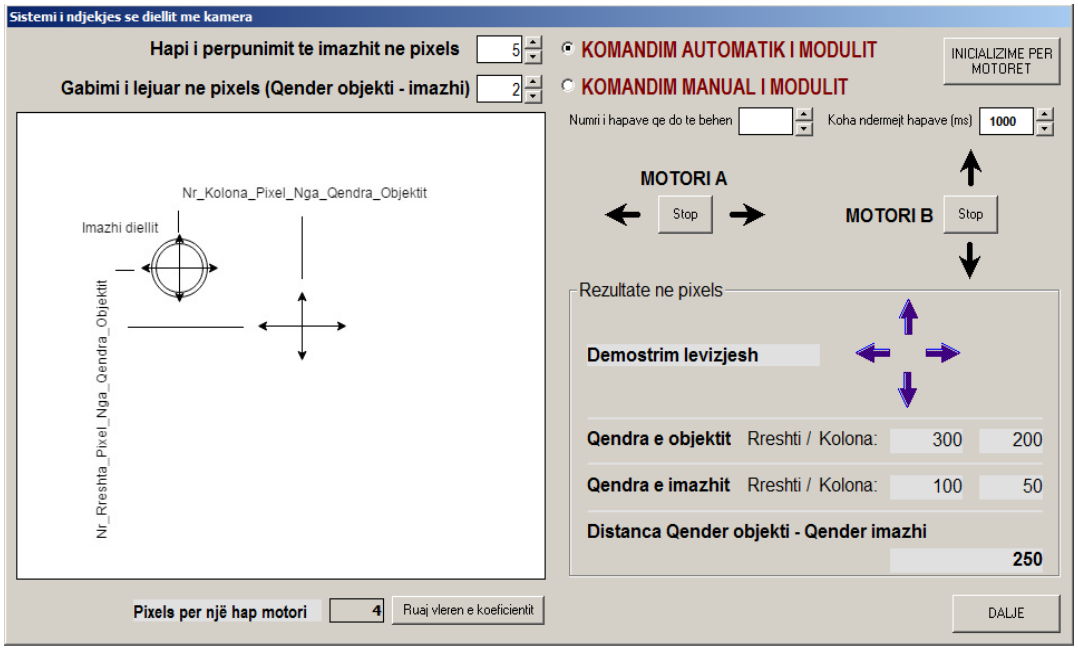


Fig. 4.1.6. *GUI i ndërtuar për komandimin e lëvizjes së panelit sipas një aksi dhe dy akseve*

Në Shtojcën-8 po japim programin e plotë të ndërtuar në Visual Basic 6.0 për komandimin e modulit PIC18F4550-I/P Microchip Microcontroller & ULN2803 dhe modulit Stepper-Bee në ndjekje të objektit në lëvizje që për rastin në trajtim lidhet me panelin diellor.

Si përfundim mund të themi se :

- Përshkruam një sistem të thjeshtë ndjekës objektesh nëpërmjet përpunimit të imazheve të sjella nga një kamer. Sistemi ndjekës i përshkruar dhe realizuar në këtë kapitull mund të shërbejë për ndjekjen e çfarëdolloj objekti të ndritshëm, pra ai përbën një sistem ndjekës universal. Sistemi ndjekës shpreh një mënyrë se si duke ndjekur lëvizjen e diellit, paneli diellor me fotoelementë pozicionohet sipas drejtimit ku sipërfaqja aktive për fluksin e dritës merr vlerën maksimale.
- Treguam funksionet që duhen përdorur për komandimin e motorit me hapa prej portës USB nëpërmjet :
    - PIC18F4550-I/P & ULN2803
    - Modulit Stepper-Bee.

KAPITULLI V

Hyrje

Pasi kemi përshkruar në kapitullin e parë mënyrat e komandimit të një motori me hapa prej kompjuterit nëpërmjet portave hyrëse dhe dalëse (I/O) e cila përbën bazën kryesore në komandimin e paisjeve robotike, në vazhdim do të shohim se si një sistem robotik i përparuar mund të programohet në realizimin e dy detyrave të shtruara për zgjidhje në këtë material; komandimit dhe orientimit të sistemit robotik në ndjekje të objekteve në lëvizje. Detyrat e shtruara do të zgjidhen bazuar në skemen pricipiale të dhënë në Fig. 5.1, ku paraqitet thelbi i rrugës që do të ndiqet.

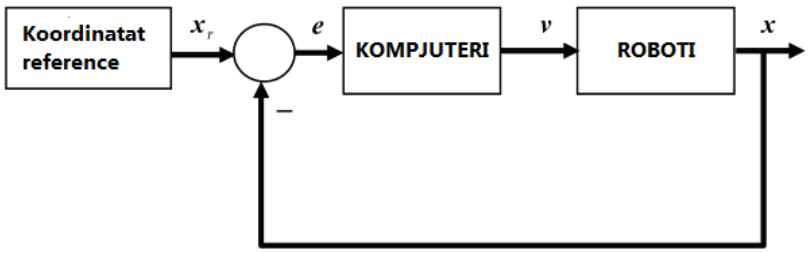


Fig. 5.1. *Skema pricipiale e komandimit per robotin*

Në realizimin e detyrave të shtruara më sipër është e nevojshme që përveç paisjeve fizike (pjesës hardware) të sigurohen dhe programet (pjesa software). Programet ose pjesa software përbëhet nga dy pjesë :

- Programi i komandave, i cili shpreh detyrën që duhet të kryhet, nëpërmjet një algoritmi të caktuar.
- Programet e komunikimit me robotin që përfaqësojnë një librari funksionesh dhe komandash, të ndërtuara për mjedise të ndryshme programimi.

Në këtë kapitull do të trajtohen pjesa hardware dhe konkretisht roboti iRobot Create Roomba 4400, i njohur si Roomba 4400 së bashku me aksesorët e tij si dhe paisjet BAM (Bluetooth Adaptor Module), DEBR (Direct Element Bluetooth Radio) dhe Camera Wireless 2.432 GHz.
Gjithashtu, do të tregohet si janë ndërtuar programet e komunikimit me robotin që emërtohen si libraritë e funksioneve dhe komandave për mjedisin Matlab, Visual Basic 6.0, Visual Studio 2005, 2008, 2010, 2012 .Net dhe Basic4Androide (Java).

5.1. Sisteme moderne robotikë të programueshëm.

Sistemet e sotëm robotikë shprehin në mënyrë direkte nivelin e lartë të teknollogjisë në botë. Përdorimet e këtyre sistemeve janë në të gjitha fushat e shoqërisë. Duke u kufizuar vetëm në modelet e iRobot Create, mund të themi se drejtimet kryesore të përdorimit të tyre janë :

- Përdorimet për qëllime të biznesit (a).
- Përdorimet në ambjentet familjare (b).
- Përdorimet në fushën e mbrojtjes dhe të sigurisë (c).

(a) (b) (c)

Fig. 5.2. *Përdorimet në fushat bisnes, familje, mbrojtje dhe siguri.*

Në mënyrë të përmbledhur përdorimet e tyre do të jepeshin si më poshtë.

Përdorimet më të njohura në fushën e biznesit janë kryesisht përdorimet e tyre në :

- Telepresence – video & audio konferenca duke lidhur me pamje dhe figurë njerëz të ndryshëm në lëvizje të lirë të tyre.
- Telemedicine - video & audio biseda midis pacientëve dhe mjekëve që ndodhen larg prej pacientit. Në këtë rast robotët janë të shoqëruar dhe me një numër aplikacionesh ndihmëse për pacientin në zonën e shërbimit për të.
- FitnessRobotike – e cila nënkupton përdorimin e këtyre robotëve si instruktorë të lëvizshëm për persona që zhvillojnë ushtrime fizike në palestra dhe më tej.

Përdorimet më të njohura në ambjentet familjare janë kryesisht përdorimet e tyre si :

- Pastruesa të sipërfaqeve të dhomave dhe sallave të ndryshme.
- Pastruesa të pjesëve të brendëshme të pishinave.
- Pastruesa të pjesëve specifike të çative (ulluqeve) kur ato janë të mbuluara nga gjethet apo plurat e baltosur.

Përdorimet më të njohura në fushën e mbrojtjes dhe të sigurisë janë kryesisht përdorimet e tyre në :

- Shërbime publike - këta robotë janë të aftë për të mbajtur njeriun larg nga zona e rezikut duke zgjidhur problemet në distancë, në mënyrë të përsosur. Përdorime të tilla mund të jenë rastet kur duhet të punohet në mjedise me agjentë të dëmshëm për njeriun si p.sh në zona me radioaktivitet apo në mjedise acide, helmuese etj.
- Fushën e sigurisë - përdorimi i tyre për hetim, mbulim, verifikime dhe çaktivizime të lëndëve plasëse etj. Këta robotë janë të paisur me sisteme të sofistikuara sensorësh bazuar në ultratingujt, kamerat, infrared, blutouth, Wi-Fi dhe GPS për kontroll në distancë.
- Fushën e mbrojtjes - robotë të fuqishëm me një lëvizmëri të madhe në terrene të vështira dhe shkallë me aftësi ngritje peshash deri në 150 kg, të pajisur me sisteme vëzhgimi, infrared, blutouth, sensore nxehtësie dhe Wi-Fi dhe GPS për kontroll në distancë.

iRobot Create Roomba 4400 është pjesë e grupit të robotëve që përdoren në ambjentet familjare si dhe për qëllime studimi si një robot i programueshëm.

5.2. Roboti model Roomba 4400.

iRobot Create Roomba 4400 është një robot i programueshëm i cili të lejon të ndërtosh programe që do të kontrollonin dhe drejtonin sjelljen e tij, pa u shqetësuar për gjuhën e programimit në nivel të ulët. Gjithashtu Roomba 4400 ofron një grup komandash dhe funksionesh të ngjashme me “driver” komandat. Me robotin Roomba 4400 të paraqitur në Fig 5.3, mund të ndërtohen sisteme më të zhvilluara duke shtuar paisje të tjera elektronike si krah robotik, ekrane të vegjël dhe sensorë të

ndryshëm. Edhe në këto raste përsëri përdoruesit kanë hapësirë për të komanduar këto sisteme duke punuar jo në gjuhë të nivelit të ulët por në gjuhë të nivelit të mesëm dhe të lartë.

Fig. 5.3. *Roomba 4400 dhe aksesorë të tjerë.*

Aksesorët që lidhen me Roomba 4400 po i përshkruajmë shumë shkurtimisht më poshtë :

- Remonte Control - shërben për të dërguar komanda direkt në robot prej përdoruesit.

- Virtual Halo for iRobot Create - Kufizon vajtjen e robotit në zona të caktuara, duke shfrytëzuar rezet infrared, ku mund të këtë kafshë apo zona me ushqime, ujë etj.

- Compact Home Base® for Create - Baza e ngarkimit të baterive të robotit paisur me komunikim infrared.

- Advanced Power System (APS) 3000 mAh NiMH Battery.

- Ushqyesi i tensionit me nivel 16 V - Karikon bateritë e robotit në 3 orë.

- BAM - Paisja e komunikimit me valë bluetooth e cila montohet në porta me 25 pine e robotit.

- DEBR (Direct Element Bluetooth Radio) - Paisja e komunikimit me valë bluetooth e cila montohet në kompjuter në një nga portat USB dhe komunikon me BAM pas konfigurimeve të nevojshme.

- Kablli serial për komunikimin midis kompjuterit dhe robotit.

- RCM - Robot Command Module nëpërmjet këtij moduli me ATMega168 microcontroller bëhet e mundur lidhja e robotit me kompjuterin. Ky modul përmban katër porta seriale dhe një USB.

Komandimi i robotit mund të bëhet në dy mënyra nga kompjuteri duke shfrytëzuar :

- Lidhjen seriale nëpërmjet UART (Universal Asynchronous Receiver/Transmitter) të paraqitur në Fig. 5.4 dhe në Fig. 5.5, pjesë e komunikimit tek paisjet kompjuterike kur komunikimet kryhen në portat paralele dhe seriale. Komunikimi nëpërmjet UART përmbledh lidhjet e njohura VNM, RS-232, RS-422 dhe RS-485. Përcaktimi 'Universal' në emërtimin e UART, nënkupton se transmetimi i të dhënave kryhet me shpejtësi të konfigurueshme.

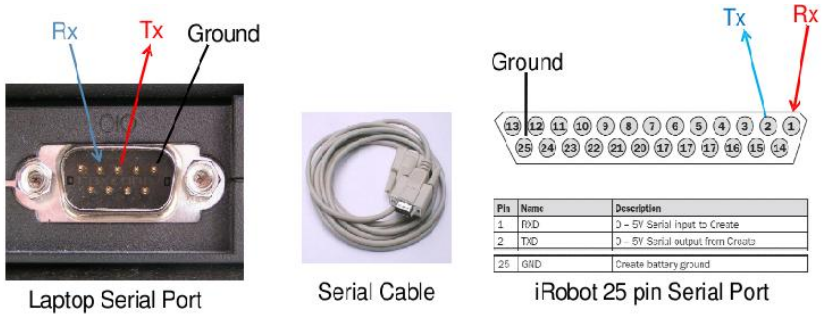


Fig. 5.4. *Portat seriale, kablli serial, porta seriale 25 pin.*

Konfigurimet për komunikimet me robotin janë në vlerat e mëposhtëme :
Baud: 57600 or 19200 ; Data bits: 8; Parity: None; Stop bits: 1; Flow control: None
iRobot Create komunikon në band 57600 bit/sec. N.q.s se jeni duke përdorur mikrocontroller që nuk suporton baud 57600 bit/sec, atëhere ka dy rrugë të konfigurohet iRobot Create në baud 19200 bit/sec.

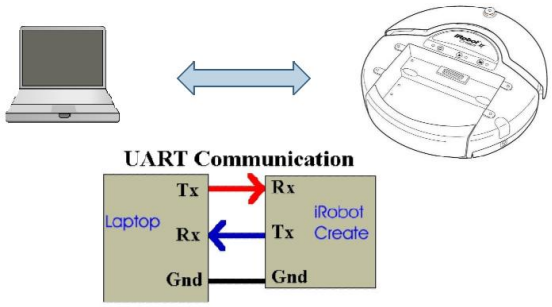


Fig. 5.5. *Komunikimi i PC me Roomba 4400 sipas nëpërmjet pasijes UART (seriale).*

- RCM (Robot Command Module) një modul me mikrokontroller ATMega168. Për këtë rast komunikimi fizik bëhet nëpërmjet një kablli USB që lidh kompjuterin me RCM. Në këtë material nuk do të trajtojmë zgjidhjen e detyrave të shtruara me ndihmën e këtij moduli.

Komunikimi me robotin nga kompjuteri nëpërmjet objekteve dhe portave seriale mund të realizohet fizikisht në dy mënyra të cilat po i rendisim më poshtë :

- Lidhja kabllore nga portat seriale paraqitur në Fig. 5.6. Në gjuhët e programimit komunikimi i PC me Roomba 4400 realizohet nëpërmjet objekteve seriale.

Fig. 5.6. *Komunikimi i PC me Roomba 4400 me kabël.*

- Lidhja valore realizuar nga dy paisje ndihmëse BAM dhe Direct Element Bluetooth Radio nëpërmjet objekteve seriale, paraqitur skematikisht në Fig. 5.7

Fig. 5.7. *Komunikimi i PC me Roomba 4400 me Bluetooth*

.

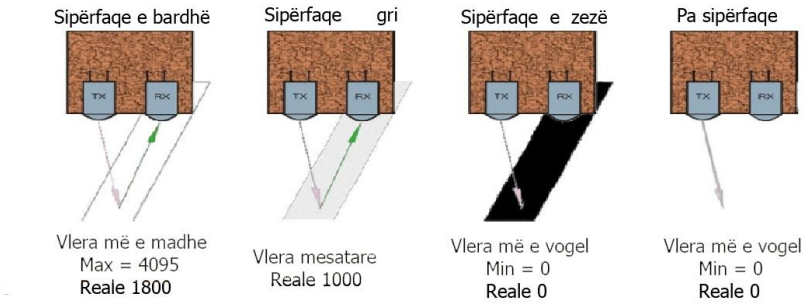
Në gjuhët e programimit komunikimi i PC me Roomba 4400 për të dyja rastet realizohet nëpërmjet objekteve seriale te konfiguruar në vlerat :

Baud: 57600 or 19200 ; Data bits: 8; Parity: None; Stop bits: 1; Flow control: None

Në vazhdim të ndërtimit të robotit mund të themi se ai përbëhet nga dy pjesë kryesore :

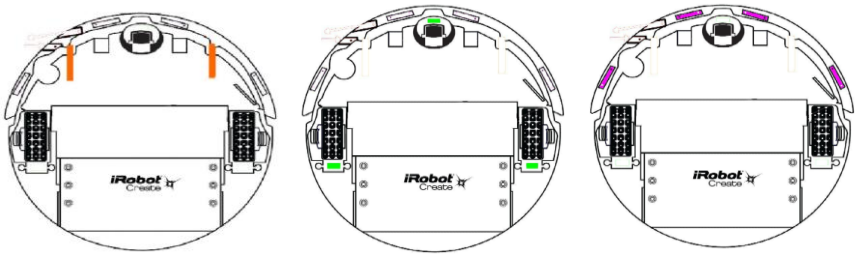
- Pjesët e lëvizëshme (actuators). Këtu hyjnë pesë motorët e robotit Roomba.

- Sensorët të cilët klasifikohen në tre grupe sipas funksionit që ata kryejnë :
  - Të jashtëm që sjellin në robot informacione nga mjedisi përreth.
    - *Bumper* : Parakolpi është pjesë e luajtshme në pjesën e përparme të Roombas siç tregohen në Fig 5.8 (a). Roboti ka një sensor presioni në çdo anë të parakolpit (gjithsej janë dy), kështu që mund të themi se nëse Roomba përplaset në diçka në anën e majtë apo të djathtë, sensorët japin informacion mbi përplasjen.
    - *Wheeldrops* : Në secilën nga rrotat e robotit ka nga një sensor që sinjalizon nëse rrota është e shkëputur nga toka ose jo, siç tregohen në Fig 5.8 (b). Gjithsej ka tre të tillë .
    - *Wall* : Sensori i murit është një sensor infrared që përcakton nëse roboti është duke ecur ngjitur me murin. Ky sensor është në pjesën e djathtë të përparme të parakolpit të robotit.
    - *Cliff* : Janë katër sensorë të tillë përgjatë parakolpit në pjesën e brendëshme të poshtëme të parakolpit siç tregohen në Fig 5.8 (c). Këta sensorë mund të shikohen pas kthimit përmbys të robotit ku mund të shikohen diodat infrared të plastifikuara. Funksioni i tyre jepet në figurën e mëposhtëme.



- *Virtual wall* : Sensori i murit virtual është një sensor infrared i kapjes së valës në të gjitha drejtimet (omnidirectional), vendosur në pjesën e sipërme të parakolpit. Paisja e murit virtual nxjerr një sinjal të moduluar të valëve infra të kuqe, dhe në qoftë se Roomba merr këtë sinjal ai zbulon murin virtual. Në këtë mënyrë roboti reagon duke ju shmangur kufizimit të murit virtual. Vlera 0 (zero) nënkupton mungesën e murit virtual.
- *Motor Overcurrents* : Sensorët e mbirrymës janë sensorë që lidhen me çdonjërin nga pesë motorët e robotit të cilët janë :
  - Motori i rrotës së majtë
  - Motori i rrotës së djathtë
  - Motori i furçës kryesore
  - Motori i vakumit
  - Motori i furçës anësore

  Kur ndonjë prej motorëve nuk mund të rrotullohet prej pengesave të jashme që shfaqen, në këta motorë mbirrymat që lindin janë shenja që venë në veprim sensorët e mbirrymës.
- *Dirt Dedect* : DD sensor është një disk i vogël bronxi në pjesën e furçës së vakumit që jep informacion mbi madhësinë e ndotjes.



(a) (b) (c)

Fig. 5.8. *Sensorët e jashtëm në Roomba 4400*

- Të brendshëm që sjellin në robot informacione për gjendjen e brendshme të robotit.
  - *Charging state* : Tregon gjendjen e Roombas në ngarkim (ushqim ose në gjuhën popullore karikim): Jo ngarkuar, duke u ngarkuar, gabim ngarkimi.
  - *Voltage* : Tregon nivelin e tensionit në mV. Kjo është normalisht midis 15000 - 17000 mV.

- *Current* : Sensori që jep vlerësim për rrymën që rrjedh në baterinë e robotit. Ajo ka shenjë pozitive kur roboti është duke u ngarkuar dhe negative kur është duke u shkarkuar.
- *Temperature* : Sensori që shpreh temperaturën e baterisë në ^0C .

- Të pozicionit dhe ndërlidhjes të cilët marrin informacione mbi lëvizjen dhe ndërlidhjen (remote control).
  - *Buttons* : Sensorë binarë që kanë dy gjendje 1 (on - shtypur) ose 0 (off - jo shtypur).
  - *Distance* : Jep distancën në *mm* për rrugën që ka kryer roboti që prej kërkesës së fundit për *distancën.*Vlera e dhënë nuk është plotësisht e saktë për të gjitha rastet sepse rrotat e robotit në raste rrëshqitjeje nuk reflektojnë lëvizjen. Distanca është rruga mesatare e kryer e dy rrotave të robotit dhënë në Fig. 5.9 (a,b,c). Pas zerimit, vlera fillon nga zero.
  - *Angle* : Këndi për këtë rast llogaritet nëpërmjet disa formulave specifike të nxjerra nga projektuesit e robotit që marrin parasysh dy leximet e njëpasnjëshme të lëvizjes së robotit. Edhe vlera e këtij sensori si edhe ajo e sensorit *distance* është jo plotësisht e saktë, për të njëjtat arsye që u përshkruan më sipër, që lidhej me rrëshqitjet e rrotave të robotit dhënë në Fig. 5.9 (a,b,c).
  - *Remote Opcode* : Sensori që merr komandat e dërguara nga Remote Control. Gjendja e tij është 255 kur mungojnë komandat.

(a) (b) (c)

Fig. 5.9. *Sensorët e pozicionit në Roomba 4400*

Në Fig. 5.10 jepen katër raste të përdorimit të Roombas dhe paisjeve të tjera së bashku, në zgjidhjen e detyrave që kemi shtruar në këtë material dhe të cilat do të trajtohen më hollësisht në vijim.

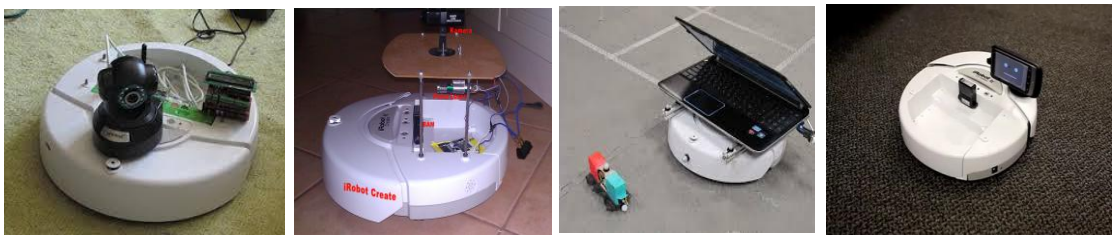


Fig. 5.10. *Roomba 4400 lidhur me paisje të tjera laptop, android mobile dhe kamera.*

5.3. Specifikime dhe libraria e komandave në mjedisin Matlab.

iRoboti Create është një model më së miri i programueshëm në Matlab. Ndërtimi i programeve në Matlab bëhet edhe më i thjeshtë pasi për këtë robot është ndërtuar një librari që siguron komunikimet me të. Në këtë librari të ndërtuar nga Joel Esposito, Owen Barton, Joshua Koehler, David Lim, SEAP Interns, jepen të gjithë funksionet e komadimit dhe ato të gjendjes së robotit Roomba [42]. Njohja e kësaj librarie është shumë e rëndësishme sepse ajo do të përbëjë një bazë të rëndësishme të mënyrës se si do të organizohen programet për kryerjen e detyrave të ndryshme si dhe kjo librari do të jetë baza për ndërtimin e librarive në mjedise të tjera programimi që do të përshkruhen në paragrafet në vijim.

Paisjet fizike në realizimin e lidhjes midis PC dhe robotit.

Pavarësisht lidhjes fizike që realizon komunikimin e kompjuterit me robotin si mini-DIN, USB-serial adapter ose Bluetooth, Matlab-i i trajton të gjitha lidhjet si një lidhje në portë seriale. Matlab-i menaxhon portën seriale të komunikimit duke krijuar atë që njihet si 'serial port object'. Ato janë variabla objekt, që përmbajnë të gjithë numrat e portave seriale dhe parametrat që do të duhen për të komunikuar me Roomba 4400 [39].

Funksione te librarise per realizimin e komunikimit te kompjutertit me robotin.

Krijimi i variablit objekt të portës seriale në Matlab bëhet si më poshtë :

```
>> [serialObject] = RoombaInit(1) ;
```

Variabli hyrës në funksion, është një numër i plotë që specifikon numrin e portës, p.sh `RoombaInit(1)` përdoret kur krijohet variabël objekt për portën seriale COM1.

```
function [serPort] = RoombaInit(my_COM);
%[serPort] = RoombaInit(my_COM)
% By; Joel Esposito, US Naval Academy, 2011
global td
td = 0.015;
% This code puts the robot in CONTROL(132) mode, which means does NOT stop
% when cliff sensors or wheel drops are true; can also run while plugged into
charger
Contrl = 132;
% Esposito 9/2008
warning off
%% set up serial comms,
% output buffer must be big enough to take largest message size
comm = strcat('COM', num2str(my_COM));

a = instrfind('port',comm);
if ~isempty(a)
            disp('That com port is in use.   Closing it.')
            fclose(a);
            pause(1)
            delete(a);
            pause(1)
end

disp('Establishing connection to Roomba...');
```

```
% Band rate - defaults at 57600, can change
% Kompjuteri dhe Matlab komunikojnë në 576000 baud. Kjo nënkupton që ka një limit
% se sa shpejt roboti dërgon, merr dhe i përgjigjet komandave.

serPort = serial(comm,'BaudRate', 57600);
set(serPort,'Terminator','LF')
set(serPort,'InputBufferSize',100)
set(serPort, 'Timeout', 1)
set(serPort, 'ByteOrder','bigEndian');
set(serPort, 'Tag', 'Roomba')
disp('Opening connection to Roomba...');

fopen(serPort);                  %% Hapja e portes seriale te gjedhur

%% Confirm two way connumication
disp('Setting Roomba to Control Mode...');

% Start! and see if its alive
Start=[128];
fwrite(serPort,Start);
pause(.1)
fwrite(serPort,Contrl);
pause(.1)
% light LEDS
fwrite(serPort,[139 25 0 128]);
disp('I am alive if my two outboard lights came on')
confirmation = (fread(serPort,4))
pause(.1)
```

Nëse funksioni inicializon drejt robotin, do të shohim dy drita të ndezura në pjesën e sipërme të tij. Nëse dështon inicializimi, fillimisht kontrollohet porta COM, që të mos jetë e zënë prej ndonjë programi tjetër. Pas këtij veprimi, veprimi i radhës është rindezja përsëri e robotit. Në të gjitha rastet duhet të sigurohemi që kablli i ushqimit të mos jetë i lidhur dhe me tension. Funksioni i mësipërm starton robotin në 'full mode'. Output-i i funksionit (`RoombaInit`) është një objekt i portës seriale që përmban të gjitha karakteristikat për të komunikuar me robotin. Nëse kemi disa robotë Roomba që do të dëshironim të operonin në të njëtën kohë, secili prej tyre mund të lidhej me një portë të veçantë. Një mënyrë tjetër komunikimi bëhet nëpërmjet Bluetooth-it, i cili e bën më të thjeshtë, por mund të shfrytëzohen edhe paisjet USB hub ose Serial hub. Në rastin e operimit me dy robotë, do të nevojitej të krijohen objekte për portat seriale për çdo robot [39]. Në rastin e tre robotëve inicializimet do të qenë :

```
>> [ObjectSerialRoomba1] = RoombaInit(1) ;
>> [ObjectSerialRoomba2] = RoombaInit(2) ;
>> [ObjectSerialRoomba3] = RoombaInit(3) ;
```

Janë dy metoda për të lexuar gjendjet e sensorëve :

- Përdorimi i funksionit `AllSensorsReadRoomba`. Ky funksion rikthen shumë shpejt vlerën e të gjithë sensorëve më një kërkesë të vetme. Përgjigja përbën një varg të gjatë string, i cili përmban gjendjen e sensorëve të robotit. Pas thirrjes së këtij funksioni përveç leximit të gjendjeve të sensorëve, bëhet dhe zerimi i vlerave për distancën dhe këndet e lëvizjes. Kjo do të thotë se përdorimi i kësaj komande jo në çdo rast është i përshtatshëm. Në të tilla kushte, përcaktimi i vlerave për distancën dhe këndet e lëvizjes do të bëhet nëpërmjet programeve specifike të ndërtuara nga programues.

- Përdorimi individual i komandave për sensoret si p.sh : DistanceSensorRoomba, AngleSensorRoomba, BatteryVoltgeRoomba, etj. Kjo metodë ka avantazhin se duke përdorur komanda të veçanta bëhet e mundur të informohemi për parametra dhe madhësi të specifikuara, në një kohë të shkurtër. E meta e kësaj metode është se informacionet për parametrat dhe madhësite e kërkuara nuk janë koherente, që do të thotë se ato të grupuara nuk përfaqësojnë gjendjen e robotit në një çast të kohës. Kështu n.q.s do të duhej të informoheshim për këndin e lëvizjes prej funksionit *AngleSensorRoomba* dhe distancën e kryer prej funksionit *DistanceSensorRoomba* , rezultatet e marra do të përfaqësonin vlerat në kohën t_1 dhe t_2 , kohë kur janë ekzekutuar funksionet përkatëse. I njëjti arsyetim nuk do të bëhej për rastet kur do të kërkohej informacion mbi gjendjen e robotit (sensorëve të ndryshëm), për sa kohë nuk ndryshon gjëndja e tyre.

Më poshtë po japim listën e funksioneve inicializuese dhe atyre që përcaktojnë gjëndjen e sensorëve, parametrave kryesorë dhe madhësive që përcaktojnë lëvizjen [39].

- *RoombaInit* Inicializon portën seriale për t'u përdorur me Roomben

Leximi i gjëndjeve të sensorëve :

- *[BumpRight, BumpLeft, BumpFront, Wall, virtWall, CliffLft, ...*
  CliffRgt, CliffFrntLft, CliffFrntRgt, LeftCurrOver, RightCurrOver, ...
  DirtL, DirtR, ButtonPlay, ButtonAdv, Dist, Angle, ...
  Volts, Current, Temp, Charge, Capacity, pCharge] = AllSensorsReadRoomba(serPort);

  Ky funksion lexon të gjithë sensorët e Roombës në të njëjtën kohe. Vlerat e kthyera janë 0 ose 1, për variablat llogjikë të mëposhtëm :

  BumpRight (0/1),
  BumpLeft(0/1),
  BumpFront(0/1),
  Wall(0/1),
  virtWall(0/1),
  CliffLft(0/1),
  CliffRgt(0/1),
  CliffFrntLft(0/1),
  CliffFrntRgt(0/1),
  LeftCurrOver (0/1),
  RightCurrOver(0/1), ...
  DirtL(0/1),
  DirtR(0/1),
  ButtonPlay(0/1),
  ButtonAdv(0/1),
  Dist (metrat që nga thirrja e fundit),
  Angle (këndi në rad që nga thirrja e fundit),
  Volts (V),
  Current (Amps, positive is charging),
  Temp (celcius),

Charge (milliamphours),
Capacity (milliamphours),
pCharge (ngarkesa në %)

Nëse do të duhet të lexojmë disa sensorë në të njëjtën kohë, kjo do të ishte mënyra më e shpejtë, krahasuar me rastin e thirrjeve individuale të funksioneve të sensorëve. Por duhet patur kujdes sepse sa herë që thirret funksioni *AllSensorsReadRoomba*, ai zeron (reset-on) vlerat për sensorët e distancës dhe të këndit. Këta sensorë do të kthejnë distancën e kryer që nga thirrja e fundit.

- *[AngleR]=AngleSensorRoomba(serPort)*
  Këndi që është kthyer Roomba gjatë lëvizjes që prej leximit të fundit. Variabli *serPort* tregon objektin serial nëpërmjet të cilit realizohet lidhja e Roombës me kompjuterin. Rezultati i marrë nga thirrja e këtij funksioni është një numër real (dooble) i cili mund të marrë vlera positive ose negative. Këndet në drejtim të kundërt me akrepat e orës janë positive dhe këndet në drejtimin e akrepave të orës janë negativë.

- *[Charge, Capacity, Percent] =BatteryChargeReaderRoomba(serPort)*
  Funksion për gjendjen e baterisë. Funksion kthen tre vlera të cilat janë :
  - Karikimin e mbetur (milliAmpHours)
  - Kapacitetin maksimal të baterisë (mAH)
  - Përqindjen e mbetur të baterisë

- *[Voltage]=BatteryVoltageRoomba(serPort)*
  Tregon nivelin e tensionit të baterisë në volt (V).

- *[BumpRight,BumpLeft,WheDropRight,WheDropLeft,WheDropCaster,BumpFront] = BumpsWheelDropsSensorsRoomba(serPort)*
  Funksioni kthen gjendjen e sensorëve *të përplasjes (bump)* dhe *rënies së rrotës* (*wheel drop)*. Parakolpi në roboti është pjesë e luajtshme, në pjesën e përparme. Ai ka një sensor presioni në çdo anë të tij (gjithsej janë dy), kështu që mund të themi se nëse Roomba përplaset në diçka në anën e majtë apo të djathtë, sensorët japin informacion mbi përplasjen.
  BumpFront merr vlere kur te dy sensoret *BumpRight* dhe *BumpLeft* kanë marrë vlera. Nëse ndodh kështu, *Right* dhe *Left* marrin vlerat 0 dhe *Front* merr vlerën 1.

- *[ButtonAdv,ButtonPlay] = ButtonsSensorRoomba(serPort)*
  Shfaq gjendjen e butonave Play dhe Advance te Roombes per gjendjet shtypur (1), ose jo shtypur (0).

- *[state] = CliffFrontLeftSensorRoomba(serPort)*
  Gjendja e sensorit *cliff front left*. Variabli *state* merr vlerat 1 (ngacmuar) ose 0 (jo i ngacmuar)

- *[strg] = CliffFrontLeftSignalStrengthRoomba(serPort)*
  Fuqia e sinjalit të sensorit *front left cliff* në % . Vlerat e *strg* janë në intervalin $(0-100)\%$

- *[state] = CliffFrontRightSensorRoomba(serPort)*

Gjendja e sensorit *cliff front right*. Variabli $state$ merr vlerat 1 (ngacmuar) ose 0 (jo i ngacmuar)

- *[strg] = CliffFrontRightSignalStrengthRoomba(serPort)*
  Fuqia e sinjalit të sensorit *front right cliff* në % . Vlerat e $strg$ janë në intervalin $(0-100)\%$

- *[state] = CliffLeftSensorRoomba(serPort)*
  Gjendja e sensorit të *cliff left*. Variabli $state$ merr vlerat 1 (ngacmuar) ose 0 (jo i ngacmuar)

- *[strg] = CliffLeftSignalStrengthRoomba(serPort)*
  Fuqia e sinjalit të sensorit *left cliff* në % . Vlerat e $strg$ janë në intervalin $(0-100)\%$

- *[state] = CliffRightSensorRoomba(serPort)*
  Gjendja e sensorit të *cliff right*. Variabli $state$ merr vlerat 1 (ngacmuar) ose 0 (jo i ngacmuar)

- *[strg] = CliffRightSignalStrengthRoomba(serPort)*
  Fuqia e sinjalit të sensorit *right cliff* në % . Vlerat e $strg$ janë në intervalin $(0-100)\%$

- *[Current] = CurrentTesterRoomba(serPort)*
  Paraqet energjinë (në amper) që rrjedh brenda ose jashtë baterisë. Rryma negative tregon se energjia është duke u konsumuar pra duke "ikur jashtë" baterisë. Rryma positive tregon se është duke u karikuar pra energjia po "shkon drejt" baterisë.

- *[Distance] = DistanceSensorRoomba (serPort)*
  Funksioni kthen madhësinë $Distance$ të kryer gjatë lëvizjes prej robotit të lidhur në portën $serPort$. Kjo distancë llogaritet prej momentit të zerimit të fundit, që lidhet me përdorimin e herës së fundit të këtij funksioni. Kur vlera e marrë prej funksionit është pozitive, kjo nënkupton se drejtimi i lëvizjes është në të njëjtën kahje me lëvizjen paraardhëse. Kur vlera e marrë është negative, kjo nënkupton se drejtimi i lëvizjes është në të kundërt me kahjen e lëvizjes paraardhëse. Vlerat e kthyera janë në diapazonin $(32.768 \div -32.768)\ \mathrm{m}$.

Komandat që mund të dërgohen drejt robotit :

- *DemoCmdsCreate*
  Plays built-in demos

- *SetLEDsRoomba(serPort, LED, Color, Intensity)*
  Diodat LED në pjesën e sipërme të robotit tek butonat *Play* dhe *Advance* janë në gjendjet *on* ose *off* dhe kontrollohen nga variabli $\mathrm{LED} = 0(off) / 1(on), 2(on), 3(on)$.
  1 për të kthyer *Play* jeshile.
  2 për të kthyer *Advance* jeshile.
  3 për të kthyer *Play* dhe *Advance* jeshile.
  $Color$ përcakton se ç'farë ngjyre do të marrë LED . Vlerat e këtij variabli janë :
  $$Color = 0\%(jeshile) \div 100\%(kuqe).$$
  $Intensity$ tregon se sa shumë do të shkëlqejë.

- *SetFwdVelRadiusRoomba(serPort, FwdVel, Radius)*
  Funksioni lëviz robotin duke specifikuar shpejtësinë dhe këndin. Variabli $serPort$ është një objekt serial port e krijuar nga *Roombainit*. $FwdVel$ është shpejtësia përpara në $\frac{m}{s}$ në kufijtë $[-0.5,\ 0.5]$. $Radius$ është në metra, positive në anë të kundërt me akrepat e orës, negative në drejtimin e akrepave të orës. Vlerat merren nga $[-2,\ 2]$.
  Raste speciale :
  - Lëvizje drejtvizore $Radius = Inf$ (infinit ne Matlab)
  - Rrotullohet në vend në kahun e akrepave të orës kur $Radius = -eps$ (eps është një vlerë reale shumë e vogël në Matlab)
  - Rrotullohet në vend në kahun e kundërt të akrepave të orës kur $Radius = -eps$ (eps është një vlerë reale shumë e vogël në Matlab)

- *SetDriveWheelsCreate(serPort, rightWheel, leftWheel)*
  Specifikon shpejtësine lineare të rrotës së majtë $leftWheel$ dhe të djathtë $rightWheel$ në $\frac{m}{s}$ $[-0.5,\ 0.5]$. Shpejtësia negative është lëvizje mbrapsht.

- *SetFwdVelAngVelCreate(serPort, FwdVel, AngVel)*
  Komanda siguron një lëvizjë të robotit me një shpejtësi përpara $FwdVel = (-0.5 \div 0.5)\frac{m}{s}$ dhe në një lëvizje rrethore me shpejtësi këndore $AngVel$.

- *BeepRoomba(serPort)*
  Komanda siguron një sinjal akustik tek roboti. Beep-i është i gjatë 20/64 pjesë të një sekonde. Kur komanda ekzekutohet në mënyrë të përsëritur, sekuencat nuk dallohen nga njëra tjetra duke krijuar idenë e një sinjali akustik të vijushëm. Për të ekzekutuar sinjale akustike të ndarë në kohë nga njeri tjetri përdoret komanda *pause(time)* si në rastin e mëposhtëm :
  >> BeepRoomba(serPort); pause(0.5); BeepRoomba(serPort);

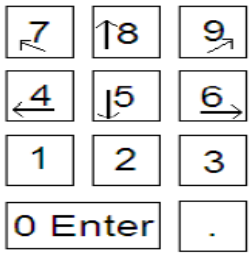
- *travelDist(serPort, speed, distance)*
  Komandon robotin e lidhur në portën $serPort$ të kryejë zhvendosjen *distance* në metra, me një shpejtësi $speed = (0.025 \div 0.5)\frac{m}{s}$.
- *turnAngleturnAngle(serPort, speed, turnAngle)*
  Komandon robotin e lidhur në portën $serPort$ të kryejë rrotullimin në këndin $turnAngle = (-360 \div 360)^0$ me një shpejtësi këndore $speed = (0 \div 0.2)\frac{rad}{s}$.

Për të komanduar robotin prej tastjerës së kompjuterit thirren funksionet e mëposhtëme :

- *[keyPresses , elapsedTime] = RobotHardKeyBoard(CommPort)*

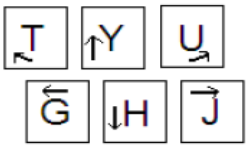
Roomba mund të kontrollohet nga tastierë numerike ose alfabetike. Për të dalë nga kontrolli i robotit nga tastiera, shtypet ne tastjerë butoni '*Q*'.

- Komandimi nga tastiera numerike.



8 – > lëviz 10 cm përpara
5 – > lëviz 5 cm prapa
4 – > kthen majtas 15 gradë
7 – > kthen majtas 5 gradë
6 – > kthen djathtas 15 gradë
9 – > kthen djathtas 5 gradë

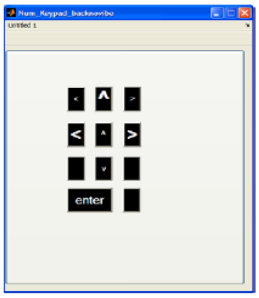
- Komandimi nga tastiera alfabetike.



Y – > leviz para 10cm
H – > leviz prapa 5cm
G – > leviz 15 grade majats
T – > leviz 5 grade majtas
J – > leviz 15 grade djathtas
U – > leviz 5 grade djathtas

- *varargout = RobotGuiControl(varargin)*
  Roomba mund të kontrollohet nga shigjetat në ekranin e kompjuterit. Njëlloj si tek funksioni *RobotHardKeyBoard*
  .
  Shigjetat kontrolluese :

Shigjeta e madhe përpara	– >	10 cm
Shigjeta e vogël përpara	– >	5 cm
Shigjeta prapa	– >	5 cm
Shigjeta e madhe majtas	– >	15 grade
Shigjeta e vogël majatas	– >	5 grade
Shigjeta e madhe djathtas	– >	15 grade
Shigjeta e vogel djathtas	– >	5 grade



Duke u mbështetur në listën e gjatë të funksioneve dhe komandave të dhëna në këtë paragraf, bëhet shumë i thjeshtë komandimi i robotit për kryerjen e detyrave të caktuara, ndërmjet të cilave dhe të atyre që janë shtruar për zgjidhje në këtë material. Një pjesë e këtyre funksioneve do të përdoren në realizimin e komandimeve sipas algoritmave të ndërtuar për kontrollin e pozicionit të robotit kundrejt objekteve të tjera në mjedis, objekte në lëvizje ose në prehje.

Në file-in MatlabToolboxiRobotCreate.zip, i cili mund të zbritet nga faqja : *www.usna.edu/Users/weapsys/esposito/roomba.matlab/*, jepen funksionet për komandimin e lëvizjeve të robotit në Matlab zhvilluar nga Joel Esposito, Owen Barton, Joshua Koehler, David Lim, SEAP Interns [39].

5.4. Specifikime dhe libraria e komandave në Visual Studio .Net (VB). Komunikimet me kabëll (wire) dhe me valë (wireless and bluetooth).

iRoboti Create është një model i programueshëm i versionit Roomba Model 4400 [41] që përdoret për qëllime mësimi dhe studimi. Për komandimin e këtij roboti prej PC është e nevojshme shfrytëzimi i portës seriale (7 pin Mini - DIN connector) ose i portës DB 25 pin ku montohet paketa Bluetooth (Fig 5.13). Programet që përbëjnë librarinë e ndërtuar paraqiten nga një grup komandash dhe funksionesh të cilat realizojnë detyra specifike. Kjo librari do të shërbejë si një urë lidhëse ndërmjet robotit dhe përdoruesit kur ai programon detyra specifike në Visual Basic.NET [5], në dërgimin e komandave drej robotit. Komandat përgjithësisht venë robotin në veprim lëvizjeje ndërsa funksionet sjellin tek përdoruesi vlerat e madhësive që karakterizohen si parametra të robotit si psh madhësia e tensionit të baterive, gjendjet e sensorëve të robotit, rrugën e kryer prej tij, etj.

Paisjet fizike në realizimin e lidhjes midis PC dhe robotit.

Dërgimi i instruksioneve dhe komandave drej robotit do të realizohet fizikisht nëpërmjet kabllit serial i cili në PC lidhet në portën COM ndërsa në iRobot Create lidhet në portën seriale 7 pin Mini-DIN connector (Fig 5.13). Pasi është bërë lidhja fizike është e nevojshme konfigurimi i portës COM në PC. Kjo në PC realizohet në Control Panel / System / Hardware / Device Manager / Ports (LPT & COM). Për konfigurin e portës seriale duhet në fillim të përcaktohet cila portë COM është lidhur (COM1, COM2 ...).Në rastin e nje PC desktop parapëlqehet që lidhja ne PC të realizohet në COM1; në rastin e një laptopi i cili nuk përmban fizikisht portë COM, lidhja realizohet nëpërmjet një konvertuesi USB në COM. Lidhja valore me robotin mund të realizohet nëpërmjet paketes Bluetooth të njohur si Element Direct - BAM (Bluetooth Adaptor Module) .

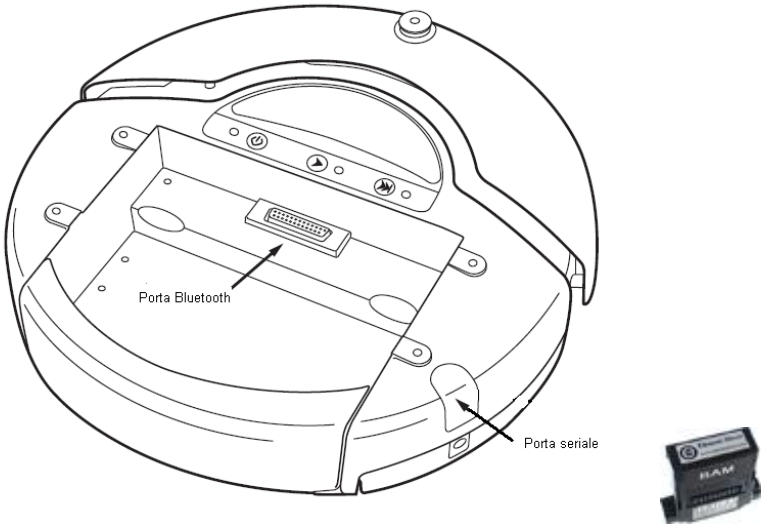


Fig 5.13. *iRobot Create dhe elementi BAM*

Programet e komunikimit te kompjutertit me robotin.

Komandimi i iRobot Create kryhet nëpërmjet një programi të thjeshtë i cili përbëhet nga një dritare ku janë vendosur disa butona me simbolikat e lëvizjeve më të mundëshme të robotit.
Pas cdo butoni që klikohet ekzekutohen komanda dhe funksione të cilat janë në përbërje të librarisë së ndërtuar në Visual Basic .NET. Në një tjetër punim, për ndërtimin e librarisë së komandimit të robotit është shfrytëzuar ActiveComport Serial Port [36]. Për realizimin e komunikimeve me Roomba 4400 në këtë paragraf nëpërmjet portës seriale do të shfrytëzohet libraria e .NET System.IO.Ports [41].

Programimi me librarinë .NET SYSTEM.IO.PORTS.

Problemi i komunikimit nëpërmejt portës seriale edhe pse është një problem i trajtuar në shumë materiale, përsëri ka specifikat e veta. Realizimi i komunikimit duke qenë kompleks ka bërë që kompani të specilizuara të ndërtojnë programe ndihmëse që shërbejnë si ura lidhëse midis programuesve dhe portës seriale. Përdorimi i librarisë së .NET System.IO.Ports është me qëllime të ndryshme si psh komandimi i paisjeve industriale të ndryshme që mund të komunikojnë me kompjuterin nëpërmjet portës seriale, komandimi dhe kontrolli i ruterave të ndryshëm në rrjet, kontrolli i modemave serialë, Bluetooth etj [36]. Kontrolli i paisjeve që lidhen me kompjuterin nëpërmjet portave USB trajtohen si lidhje me porta virtuale seriale. Kjo siguron që përdorimi i teknikave të menxhimit të portave seriale të jetë i rëndësishëm.

Për të krijuar një serial objekt duke përdorur librarinë e .NET System.IO.Ports janë të domosdoshme linjat e mëposhtme :

```
Imports System.IO.Ports
Private m_CommPort As New SerialPort
```

Objekti serial i krijuar duhet të konfigurohet dhe kryhet hapja e portës së komunikimit. Për rastin konkret për komunikimin me iRobot Create, konfigurimi i tij përcaktohet si më poshtë :

```
    Public Function OpenSerialPort(ByVal ComPort As Integer) As Boolean
        Try
            With m_CommPort
                .PortName = "COM" & CStr(ComPort)
                .BaudRate = 57600
                .DataBits = 8
                .Parity = Parity.None
                .StopBits = StopBits.One
                .Handshake = Handshake.None
                .ReadTimeout = 1000
            End With
            ' Hapja e portes seriale e suksesshme
            Return True
        Catch
            ' Problem ne hapjen e portes seriale
            Return False
```

```
        End Try
    End Function
```

Për të kryer procesin e dërgimit të datave në portë komanda e përdorur është si më poshtë :
`m_CommPort.WriteString( "any text")` ose `m_CommPort.WriteByte wByte`
Instruksioni i parë (`m_CommPort.WriteString( "any text")`) dërgon një informacion në portë të organizuar në një string, ndërsa i dyti (`m_CommPort.WriteByte wByte`) dërgon në portë 1 byte informacion (wByte). Në rastin e komunikimit me iRobot Create instruksioni që do të përdoret për dërgimin e datave në portë do të jetë ai i dërgimit byte pas byte.

Për të kryer procesin e marrjes së të dhënave në portë, të dërguara nga paisja e lidhur në anën tjetër të kabllit serial si psh roboti iRobot Create, përdoren instruksionet e mëposhtme:

```
m_CommPort.ReadByte()

m_CommPort.ReadChar()

m_CommPort.ReadLine()

m_CommPort.ReadTo()
```

Blloku i leximit të datave në portën seriale jepet më poshtë :

```
Dim n As Integer = m_CommPort.BytesToRead
Dim buffer As Byte() = New Byte(n - 1) {}
Dim offset As Integer = 0
Dim count As Integer = 4
Dim returnvalue As Integer
If n > 0 Then
   returnvalue = m_CommPort.Read(buffer, offset, count)
End If
```

Mbyllja e portës seriale të komunikimit kryhet nga komanda :

```
m_CommPort.Close()                    ‘ Komanda per mbylljen e portes
```

Mbështetur në sa u përshkruan më sipër po jepen në vijim disa komanda dhe funksione që do të përdoren për të vënë në lëvije iRobot Create.

Komandat e lëvizjes dhe funksionet e gjëndjes së robotit.

Disa nga komandat dhe funksionet më të përdorëshme do të përshkruhen më poshtë.
Hapja e portës seriale për komunikimin me robotin kryhet nga funksioni :
`RoombaInit “COM3”`
Ndërtimi i këtij funksioni jepen në vijim :

```
Public Function RoombaInit(my_COM As String) As Boolean
   Public Function IsPortAIRobotCreate(ByVal ComPort As Integer) As Boolean
        Try
            With m_CommPort
                .PortName = "COM" & CStr(ComPort)
                .BaudRate = CInt(frmMain.cmbBaud.SelectedItem)
```

```
                .DataBits = 8
                .Parity = IO.Ports.Parity.None
                .StopBits = IO.Ports.StopBits.One
                .Handshake = IO.Ports.Handshake.None
                .ReceivedBytesThreshold = 1
                .RtsEnable = True
                .ReadTimeout = 1000
                .ReadBufferSize = 1000
            End With
            Try
                m_CommPort.Open()
            Catch ex As Exception
                MessageBox.Show(ex.Message)
            End Try
            ' Write an 128 Command to the Port.
            m_CommPort.Write({128}, 0, 1)
            System.Threading.Thread.Sleep(100)
            ' Write an 132 Command to the Port.
            m_CommPort.Write({132}, 0, 1)
            System.Threading.Thread.Sleep(100)
            ' light LEDS
            Dim LedsBytes() As Byte = {139, 25, 0, 128}
            m_CommPort.Write(LedsBytes, 0, 4)
            System.Threading.Thread.Sleep(100)
            Erase LedsBytes
            ' set song
            Dim SongBytes() As Byte = {140, 1, 1, 48, 20}
            m_CommPort.Write(SongBytes, 0, 5)
            System.Threading.Thread.Sleep(50)
            ' sing it
            Dim SingBytes() As Byte = {141, 1}
            m_CommPort.Write(SingBytes, 0, 2)

            ' Lidhja me iRobot Create OK
            Return true
        Catch exc As Exception
            MsgBox("Problem lidhjeje me iRobot Create.", MsgBoxStyle.OkOnly)
            Return False
        End Try
    End Function
```

Funksioni kthen mbrapsh vlerat True ose False për rastet kur lidhja realizohet me robotin ose jo. Lëvijet para, prapa si dhe majtas, djathtas me shpejtësi lineare të përcaktuara për secilën rrotë të robotit jepet nga komanda :

```
SetDriveWheelsCreate ShpejtesiaRrotesMajtas, ShpejtesiaRrotesDjathtas
```

Ku parametrat `ShpejtesiaRrotesMajtas` dhe `ShpejtesiaRrotesDjathtas` përcaktojnë shpejtësitë e rrotës së majtë dhe të djathtë të robotit (Fig 5.14). Shpejtësia lineare e rotave të robotit përcaktohet në vlerat (-500 – 500 mm/s) [37].

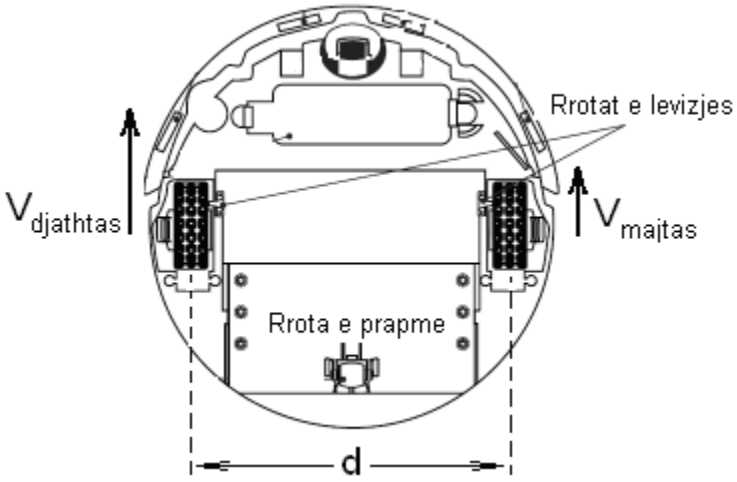


Fig 5.14. *Rrotat e iRobot Create dhe shpejtesitë lineare të tyre*

Kur kërkohet lëvizja perpara e robotit pas përcaktimit të shpejtësisë me të cilën ai do të lëvize, mjafton të ekzekutojmë komandën :

```
SetDriveWheelsCreate vLevisjeje, vLevisjeje ‘Shpejtesia  vLevisjeje =[0, 500] mm/s
```

Për lëvizjen prapa do të ekzekutonim komandën :

```
SetDriveWheelsCreate vLevisjeje, vLevisjeje ‘Shpejtesia  vLevisjeje =[-500 , 0] mm/s
```

Rrotullimet me 90 gradë majtas dhe djathtas do të kryheshin nga ekzekutimi i komandave :

```
SetDriveWheelsCreate 0, vLevisjeje   ‘Shpejtesia  vLevisjeje =[0 , 500] mm/s
SetDriveWheelsCreate vLevisjeje, 0   ‘Shpejtesia  vLevisjeje =[0 , 500] mm/s
```

Rrotullimi me 180 gradë majtas dhe djathtas do të kryheshin nga ekzekutimi i komandave :

```
SetDriveWheelsCreate - vLevisjeje, vLevisjeje      ‘Shpejtesia vLevisjeje =[0 , 500] mm/s
SetDriveWheelsCreate vLevisjeje, - vLevisjeje      ‘Shpejtesia vLevisjeje =[0 , 500] mm/s
```

Duke njohur distancën midis rrotave të robotit si dhe shpejtësitë lineare të rrotave jemi në gjendje të përcaktojmë rrotullime në kënde të dëshiruara të sistemit robotik. Për këtë do të shfrytëzojmë disa formula të thjeshta të fizikës. Problemi do të shtrohej në formën : sa duhet të jetë shpejtësia lineare e lëvizjes së rrotave në momentin kur kërkohet rrotullimi i robotit me këndin α ?
Duke u mbështetur në formula të njohura që lidhin shpejtësinë këndore me atë lineare dhe shpejtësine lineare me kohën mund të shkruajmë :

$$\omega = \frac{v}{r} = \frac{2 \cdot v}{d} \qquad (1)$$

$$\omega = \frac{\alpha}{t} \qquad (2)$$

Nisur nga 1 dhe 2 mund të llogaritim shpejtësinë lineare të rrotës për një rrotullim me këndin α të robotit gjatë kohës t.

$$\alpha = \frac{2 \cdot v \cdot t}{d} \qquad (3)$$

Ku d është distanca midis rrotave të robotit Fig 5.15.

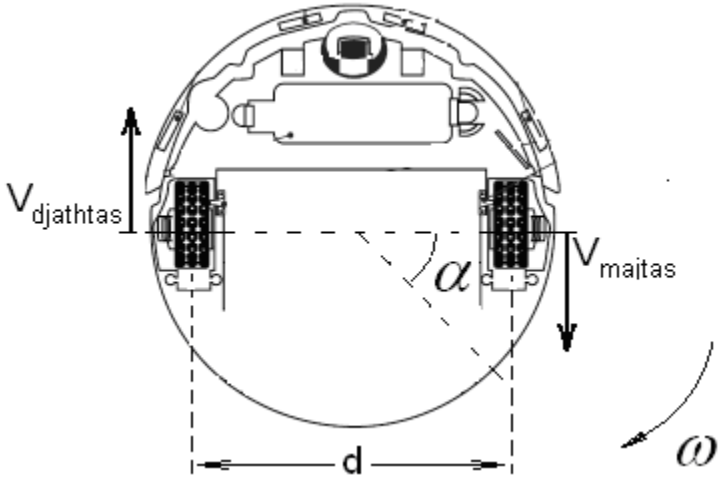


Fig 5.15. *Rrotullimi i iRobot Create*

Një komandë për lëvizje të përberë në forme harku rrethi për iRobotCreate jepet si më poshtë :

```
SetFwdVelRadiusRoomba  vLevisjeje, rRezja
```

ku :

`vLevisjeje` është shpejtësia lineare e robotit
`rRezja` është rezja e trajektores që do të përshkruaje roboti (Fig 5.16).

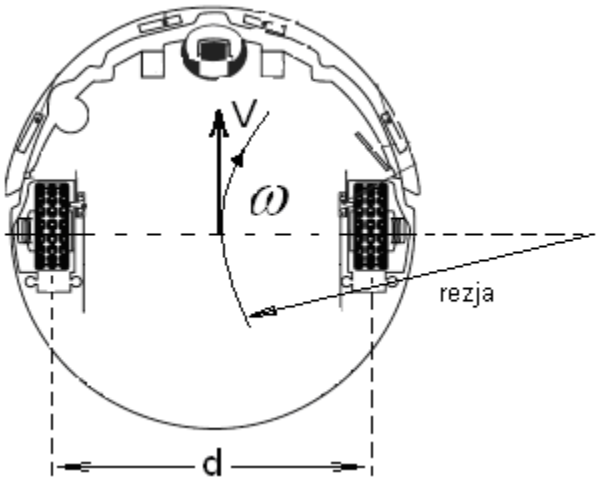


Fig 5.16. *Lëvizja e përbërë e iRobot Create*

Për të ndaluar robotin mund të përdoren komandat :
`SetDriveWheelsCreate 0, 0` ose `SetFwdVelRadiusRoomba 0,1`

Për të dy komandat e përshkruara më sipër madhësitë `vLevisjeje, rRezja` janë numra nga 16 bit pra 4 byte në total. Për rastin kur shpejtësia do të jetë -200 mm/s dhe rezja 500 mm, informacioni që do të dërgohet në portë do të jetë i renditur në byte si në vijim :
`[137] [255] [56] [1] [244]`
Komanda [137] i vendos rrotat e robotin në kontroll për një lëvizje në formë harku.
Vargu i mësipërm përfitohet nga zbërthimi i madhësive sipas formës së mëposhtme [38] :

```
vLevisjeje = -200 = hex FF38 = [hex FF] [hex 38] = [255] [56]
rRezja = 500 = hex 01F4 = [hex 01] [hex F4] = [1] [244]
```

Për të përfituar zbërthimin e mësipërm në Visual Basic .NET do të shfrytëzojmë funksionet `HIBYTE (intParameter)` dhe `LOBYTE (intParameter)` të përshkruara më poshtë :

```
Public Function HIBYTE(ByVal Intg As Integer) As Byte
    HIBYTE = CByte((Intg And &HFF00&) / 256)
End Function

Public Function LOBYTE(ByVal Intg As Integer) As Byte
    LOBYTE = CByte(Intg And &HFF&)
End Function
```

Për të përcaktuar distancën që përshkon roboti thirret funksioni `DistanceSensorRoomba()` i cili kthen mbrapsh metrat e përshkruara.

Për të vënë robotin Roomba në gjendjen beep (tingull zanor) ekzekutohet komanda : `BeepRoombaVBNET().`

Për të marë informacion mbi nivelin e baterive të robotit ekzekutohen komandat : `BatteryChargeReaderRoombaVBNET()` ose `BatteryVoltageRoombaVBNET()` të cilat kthejnë vlerën e tensionit të paketës burim.

Realizimi eksperimental i komunikimit të PC me iRobot Create.

Për të realizuar komandimin e robotit programi i ndërtuar në Visual Basic .NET përbëhet nga një formë (dritare) e paraqitur në Fig 5.17 dhe prej 12 modulesh ku secili prej tyre përmban komanda , funksione që lidhen me komunikimin e robotit si dhe funksione ndihmëse të tjera.

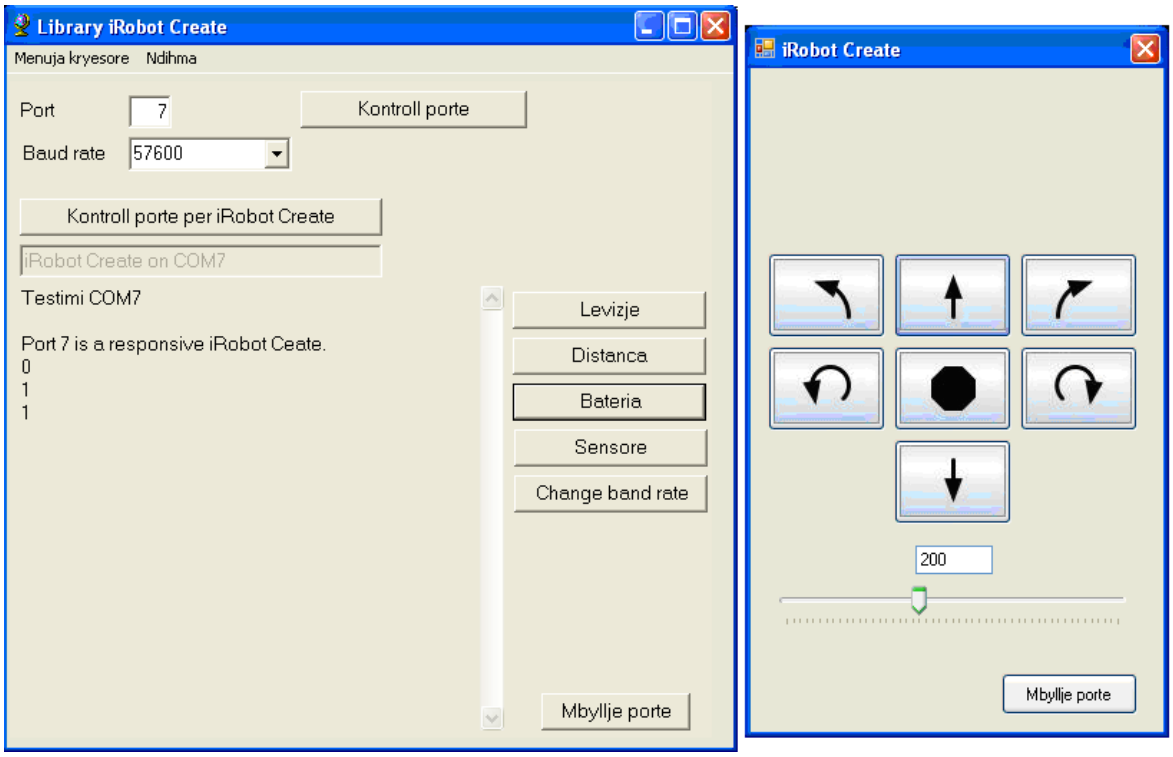


Fig 5.17. *Programi i komandimit të lëvizjes së robotit*

Funksionimi i programit është i thjeshtë. Në dritaren e portës së komunikimit (Fig 5.17) zgjidhet numri i portes seriale. Në rastin kur PC lidhet me robotin me ndihmën e kabllit serial, në portën e komunikimit zgjidhet COM1 ose COM2. Kur lidhjen midis PC dhe robotit e kryejmë në mënyrë valore atëhere në portën e komunikimit zgjidhet COM3 – COM8 (kjo zgjedhje përcaktohet nga bluetooth pas instalimit të saj).

Si përfundim :

- Nëpërmjet këtij paragrafi përshkruam se si mund të realizohet komandimi i robotit prej portës seriale duke shfrytëzuar librarinë e komandave të ndërtuar për këtë qëllim.
- Programi i ndërtuar tregon mënyrën se si mund të shfrytëzohet librarina e .NET System.IO.Ports në Visual Basic .NET.
- Nëpërmjet programit të ndërtuar në Visual Basic .NET tregohet se si mund të thirren komandat për vënien e robotit në lëvizje drejtvizore dhe rrotulluese në mënyrë të kontrolluar.

Në Shtojcën-9 po japim librarinë e komandimit të lëvizjeve të robotit në Visual Basic .NET bazuar në librarinë .NET System.IO.Ports të Microsoft.

5.5. Specifikime dhe libraria e komandave në Visual Studio 6 (VB). Komunikimet me kabëll (wire) dhe me valë (wireless and bluetooth)

Për komandimin e robotit prej PC është e nevojshme shfrytëzimi i portës seriale (7 pin Mini-DIN connector) ose i portës 25 pin (DB 25 pin connector) ku montohet paketa Bluetooth (Fig 5.18), si dhe në paragrafin e mëparshëm. Libraria e ndërtuar përbëhet prej komandash dhe funksionesh të cilat realizojnë detyra specifike. Kjo librari do të shërbejë për dërgimin e komandave prej përdoruesit drej robotit. Komandat përgjithësisht venë robotin në veprim lëvizjeje ndërsa funksionet sjellin tek përdoruesi vlerat e madhësive që karakterizohen si parametra të robotit.

Paisjet fizike në realizimin e lidhjes midis PC dhe robotit.

Dërgimi i instruksioneve dhe komandave drejt robotit do të realizohet fizikisht nëpërmjet kabllit serial i cili në PC lidhet në portën COM ndërsa në iRobot Create lidhet në portën seriale 7 pin Mini-DIN connector (Fig 5.18). Pasi është bërë lidhja fizike, është i nevojshëm konfigurimi i portës COM në PC. Kjo në PC realizohet në Control Panel / System / Hardware / Device Manager / Ports (LPT & COM). Për konfigurin e portës seriale duhet në fillim të përcaktohet cila portë COM është lidhur (COM1, COM2 ...).Në rastin e një PC desktop parapëlqehet që lidhja ne PC të realizohet në COM1; në rastin e një laptopi i cili nuk përmban fizikisht portë COM, lidhja realizohet nëpërmjet një konvertuesi USB në COM. Lidhja valore me robotin mund të realizohet nëpërmjet paketës Bluetooth të njohur si Element Direct - BAM (Bluetooth Adaptor Module).

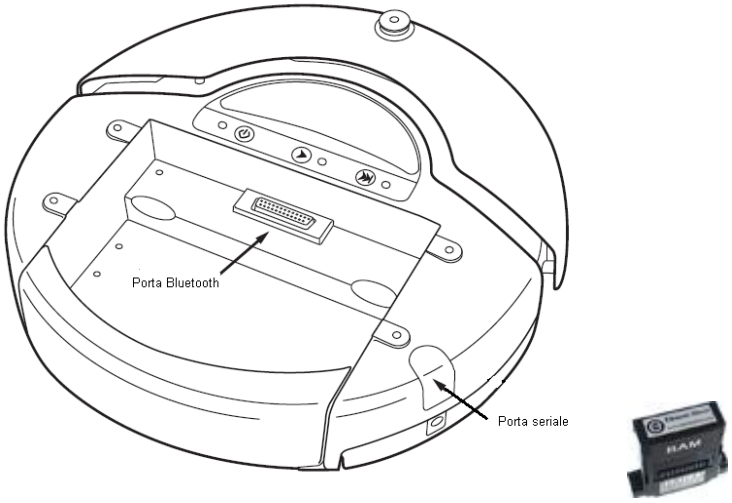


Fig 5.18. *iRobot Create dhe elementi BAM*

Programet e komunikimit të kompjutertit me robotin.

Komandimi i iRobot Create kryhet nëpërmjet një programi të thjeshtë i cili përbëhet nga një dritare ku janë vendosur disa butona me simbolikat e lëvizjeve më të mundëshme të robotit.
Pas çdo butoni që klikohet ekzekutohen komanda dhe funksione të cilat janë në përbërje të librarisë së ndërtuar në Visual Basic 6.0 [4] dhe Visual Basic .NET [5]. Libraria e komandimit te robotit

është ndërtuar duke shfrytëzuar ActiveComport Serial Port [36]. Për realizimin e komunikimeve me iRobot Create nëpërmjet portës seriale PC, mund të shfrytëzohej edhe Microsoft Comm Control 6.0 (ActiveX Control për Visual Basic 6.0). Në të dy rastet struktura bazë e komunikimeve do të qe e njëjtë me ndryshime të vogla.

Programimi me ACTIVECOMPORT SERIAL PORT

Problemi i komunikimit nëpërmjet portës seriale edhe pse është një problem i trajtuar në shumë materiale, përsëri ka specifikat e veta. Realizimi i komunikimit duke qenë kompleks ka bërë që kompani të specilizuara të ndërtojnë programe ndihmëse që shërbejnë si ura lidhëse midis programuesve dhe portës seriale [36]. Një i tillë program ndihmës që në gjuhën e programimit quhet kompoment COM është dhe ActiveComport Serial Port. Ky komponent COM është ndërtuar për të ndihmuar programuesit në gjuhët e programimit VBScript, Visual Basic 6.0 [4], Visual Basic .Net [5] dhe Visual C ++ [6]. Përdorimi i ActiveComport Serial Port është me qëllime të ndryshme si psh komandimi i paisjeve industriale të ndryshme që mund të komunikojnë me PC nëpërmjet portës seriale, komandimi dhe kontrolli i ruterave të ndryshëm në rrjet, kontrolli i modemave serialë, Usb, Bluetooth etj [36].

Për të krijuar një serial objekt COM janë të domosdoshme linjat e mëposhtme :
VB6 :

```
Dim objComport As ACOMPORTLib.ComPort
Set objComport = CreateObject("ActiveXperts.ComPort")       ‘Krijon nje Comport instance
```

.NET :

```
Public objComport As ACOMPORTLib.ComPort
Try
   objComport = CreateObject("ActiveXperts.ComPort")
   objComport.ComTimeout = 1000
Catch
   MsgBox("Problem ne procesin e krijimit te objectit ActiveXperts.ComPort", vbCritical,
            Application.ProductName)
End Try
```

Objekti serial i krijuar duhet të konfigurohet. Për rastin konkret për komunikimin me iRobot Create, konfigurimi i tij përcaktohet si më poshtë :

VB6:

```
 objComport.Device = “COM1”     ‘Perdor porten COM1 direkt pa ndermjetesine e driverave te Windows_it
 objComport.ComTimeout = 100                    ‘ Koha jashte komunikimit ( timeout )
 objComport.LogFile = App.Path & "\Errors.Log"  ‘ Log file ku rruhen mesazhe gabimesh
 objComport.BaudRate = 57600                    ‘ BoudRate ne bps (te mundeshme 19200 ose 57600)
 objComport.HardwareFlowControl = 0                  ‘
 objComport.SoftwareFlowControl = 0                  ‘
 objComport.DataBits = objComport.asDATABITS_8       ‘
 objComport.StopBits = objComport.asSTOPBITS_1 ‘
 objComport.Parity = objComport.asPARITY_NONE‘
objComport.Open                                      ‘ Komanda per hapjen e portes seriale
```

.NET

```
comm = "COM1"
```

```
objComport.Device = comm
objComport.ComTimeout = 100
objComport.LogFile = System.Windows.Forms.Application.ExecutablePath & "\Errors.Log"
objComport.BaudRate = 57600
objComport.HardwareFlowControl = 0
objComport.SoftwareFlowControl = 0
objComport.DataBits = objComport.asDATABITS_8
objComport.StopBits = objComport.asSTOPBITS_1
objComport.Parity = objComport.asPARITY_NONE
objComport.Open()
```

Për të kryer procesin e dërgimit të datave në portë komanda e përdorur është si më poshtë :
objComport.WriteString("any text") ose objComport.WriteByte wByte
Instruksioni i parë (objComport.WriteString("any text")) dërgon një informacion në portë të organizuar në një string, ndërsa i dyti (objComport.WriteByte wByte) dërgon në portë 1 byte informacion (wByte). Në rastin e komunikimit me iRobot Create instruksioni që do të përdoret për dërgimin e datave në portë do të jetë ai i dërgimit byte pas byte.

Për të kryer procesin e marrjes së të dhënave në portë, të dërguara nga paisja e lidhur në anën tjetër të kabllit serial si psh roboti iRobot Create, përdoren instruksionet e mëposhtme :

VB6 dhe vb .NET:

```
objComport.ComTimeout = 100          ‘ Koha pas se ciles ka TimeOut

str = objComport.ReadString          ‘ Lexon nje informacion text nga porta

wByte = objComport.ReadByte          ‘ Lexon nje informacion byte nga porta

wByte = objComport.ReadBytes         ‘ Lexon nje varg informacioni byte nga porta
```

Mbyllja e portës seriale të komunikimit kryhet nga komanda :

```
objComport.Close                     ‘ Komanda per mbylljen e portes
```

Mbështetur në sa u përshkruan më sipër po jepen në vijim disa komanda dhe funksione që do të përdoren për të vënë në lëvije iRobot Create.

Komandat e lëvizjes dhe funksionet e gjëndjes së robotit.

Disa nga komandat dhe funksionet më të përdorëshme do të përshkruhen më poshtë.
Hapja e portës seriale për komunikimin me robotin kryhet nga funksioni :

RoombaInit “3” ku “3” shpreh numrin e portës seriale ku do të bëhet komunikimi

Ndërtimi i këtij funksioni jepen në vijim :

```
Public Function RoombaInit(my_COM As String) As Boolean
```

```
    Dim Key                 As Integer
    Dim s                   As String
    Dim comm                As String

' Output buffer must be big enough to take largest message size
    If IsNumeric(my_COM) = False Then
       MsgBox "Porte komunikimi e pa percaktuar.", vbInformation, App.Title
       RoombaInit = False
    End If

    ' Percakton porten e komunikimit me robotin Roomba
    comm = "COM" & my_COM
    objComport.Device = comm
    objComport.ComTimeout = 1000
    objComport.LogFile = App.Path & "\Errors.Log"
    objComport.BaudRate = 57600
    objComport.HardwareFlowControl = 0
    objComport.SoftwareFlowControl = 0
    objComport.DataBits = objComport.asDATABITS_8
    objComport.StopBits = objComport.asSTOPBITS_1
    objComport.Parity = objComport.asPARITY_NONE
    objComport.Open

    confirmation = GetResult

    If objComport.IsOpened = True And objComport.LastError = 0 Then
       Key = 128: objComport.WriteByte (Key):DoSleep 0.1
       Key = 132: objComport.WriteByte (Key): DoSleep 0.1

      ' light LEDS
       Key = 139: objComport.WriteByte (Key)
       Key = 25: objComport.WriteByte (Key)
       Key = 0:objComport.WriteByte (Key)
       Key = 128:objComport.WriteByte (Key)
       ' set song
       Key = 140:objComport.WriteByte (Key)
       Key = 1:objComport.WriteByte (Key)
       Key = 1:objComport.WriteByte (Key)
       Key = 48:objComport.WriteByte (Key)
       Key = 20:objComport.WriteByte (Key):DoSleep 0.05
       ' sing it
       Key = 141:objComport.WriteByte (Key)
       Key = 1:objComport.WriteByte (Key)
       RoombaInit = True
    Else
       MsgBox  "Pamundesi ne hapjen e portes se komunikimit.", vbInformation, App.Title
       RoombaInit = False
    End If
End Function
```

Funksioni kthen mbrapsh vlerat True ose False për rastet kur lidhja realizohet me robotin ose jo. Lëvijet para, prapa si dhe majtas, djathtas me shpejtësi lineare të përcaktuara për secilën rrotë të robotit jepet nga komanda :

```
SetDriveWheelsCreate ShpejtesiaRrotesMajtas, ShpejtesiaRrotesDjathtas
```

ku parametrat `ShpejtesiaRrotesMajtas` dhe `ShpejtesiaRrotesDjathtas` përcaktojnë shpejtësitë e rrotës së majtë dhe të djathtë të robotit (Fig 5.19). Shpejtësia lineare e rotave të robotit përcaktohet në vlerat (-500 – 500 mm/s) [37].

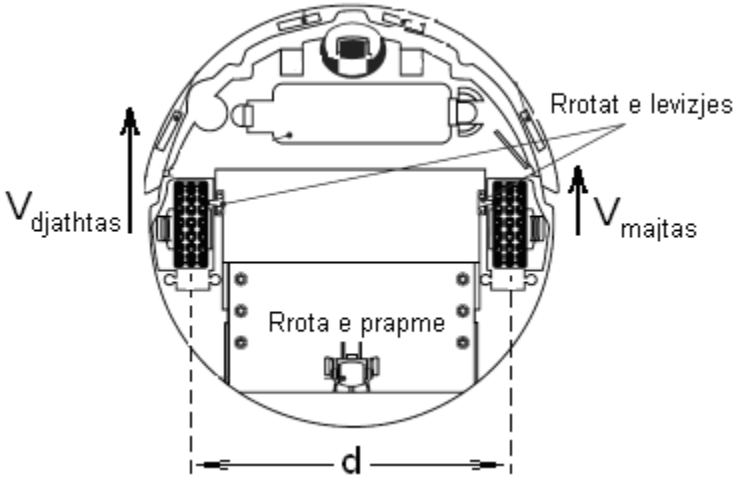


Fig 5.19. *Rrotat e iRobot Create dhe shpejtesite lineare te tyre*

Kur kërkohet lëvizja perpara e robotit pas përcaktimit të shpejtësisë me të cilën ai do të lëvize, mjafton të ekzekutojmë komandën :

```
SetDriveWheelsCreate vLevisjeje, vLevisjeje          'Shpejtesia  vLevisjeje =[0, 500] mm/s
```

Për lëvizjen prapa do të ekzekutonim komandën :

```
SetDriveWheelsCreate vLevisjeje, vLevisjeje          'Shpejtesia  vLevisjeje =[-500 , 0] mm/s
```

Rrotullimet me 90 gradë majtas dhe djathtas do të kryheshin nga ekzekutimi i komandave :

```
SetDriveWheelsCreate 0, vLevisjeje          'Shpejtesia  vLevisjeje =[0 , 500] mm/s
SetDriveWheelsCreate vLevisjeje, 0          'Shpejtesia  vLevisjeje =[0 , 500] mm/s
```

Rrotullimi me 180 gradë majtas dhe djathtas do të kryheshin nga ekzekutimi i komandave :

```
SetDriveWheelsCreate - vLevisjeje, vLevisjeje        'Shpejtesia  vLevisjeje =[0 , 500] mm/s
SetDriveWheelsCreate vLevisjeje, - vLevisjeje        'Shpejtesia  vLevisjeje =[0 , 500] mm/s
```

Duke njohur distancën midis rrotave të robotit si dhe shpejtësitë lineare të rrotave jemi në gjendje të përcaktojmë rrotullime në kënde të dëshiruara të sistemit robotik. Për këtë do të shfrytëzojmë disa formula të tjeshta të fizikës. Problemi do të shtrohej në formën : sa duhet të jetë shpejtësia lineare e lëvizjes së rrotave në momentin kur kërkohet rrotullimi i robotit me këndin α ?
Duke u mbështetur në formula të njohura që lidhin shpejtësinë këndore me atë lineare dhe shpejtësine lineare me kohën mund të shkruajmë :

$$\omega = \frac{v}{r} = \frac{2 \cdot v}{d} \qquad (1)$$

$$\omega = \frac{\alpha}{t} \qquad (2)$$

Nisur nga 1 dhe 2 mund të llogaritim shpejtësinë lineare të rrotës për një rrotullim me këndin α të robotit gjatë kohës t.

$$\alpha = \frac{2 \cdot v \cdot t}{d} \qquad (3)$$

Ku d është distanca midis rrotave të robotit Fig 5.20.

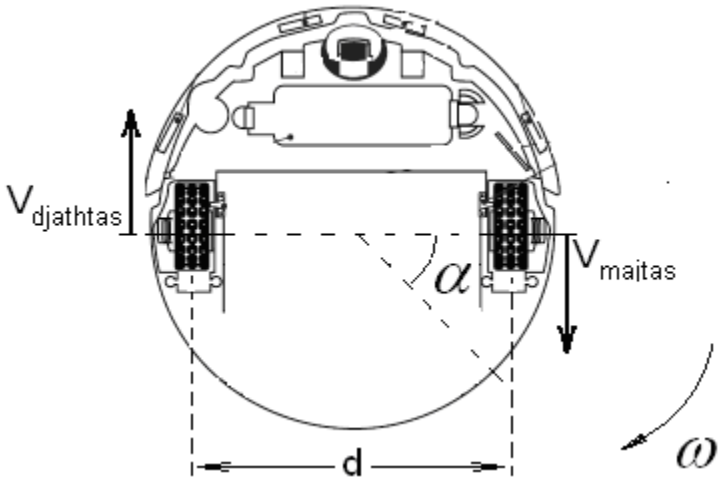


Fig 5.20. *Rrotullimi i iRobot Create*

Një komandë për lëvizje të përbërë në formë harku rrethi për iRobotCreate jepet si më poshtë :

```
SetFwdVelRadiusRoomba  vLevisjeje, rRezja
```

ku : `vLevisjeje` është shpejtësia lineare e robotit
`rRezja` është rrezja e trajektores që do të përshkruajë roboti (Fig 5.21).

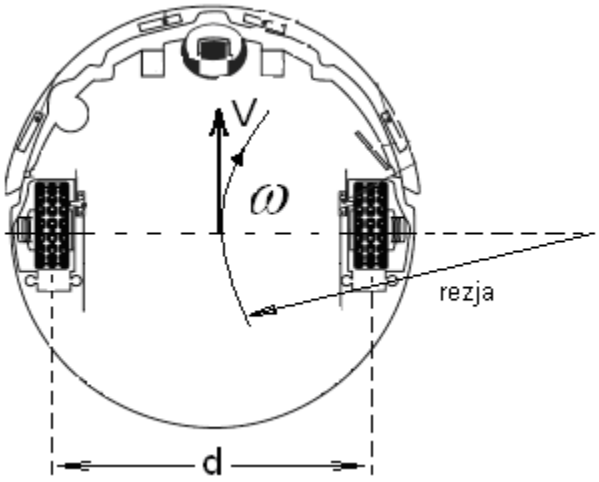


Fig 5.21. *Lëvizja e përbërë e iRobot Create*

Për të ndaluar robotin mund të përdoren komandat :
`SetDriveWheelsCreate 0, 0` ose `SetFwdVelRadiusRoomba 0,1`

Për të dy komandat e përshkruara më sipër madhësitë `vLevisjeje`, `rRezja` janë numra nga 16 bit pra 4 byte në total. Për rastin kur shpejtësia do të jetë -200 mm/s dhe rrezja 500 mm, informacioni që do të dërgohet në portë do të jetë i renditur në byte si në vijim :
`[137] [255] [56] [1] [244]`
Komanda `[137]` i vendos rrotat e robotit në kontroll për një lëvizje në formë harku.
Vargu i mësipërm përfitohet nga zbërthimi i madhësive sipas formës së mëposhtme [38] :
`vLevisjeje = -200 = hex FF38 = [hex FF] [hex 38] = [255] [56]`
`rRezja = 500 = hex 01F4 = [hex 01] [hex F4] = [1] [244]`

Për të përfituar zbërthimin e mësipërm në Visual Basic 6.0 dhe Visual Basic .NET do të shfrytëzojmë funksionet HIBYTE (intParameter) dhe LOBYTE(intParameter) të përshkruara më poshtë :

```
Public Function HIBYTE(ByVal Intg As Integer) As Byte
          HIBYTE = (Intg And &HFF00&) / 256
End Function

Public Function LOBYTE(ByVal Intg As Integer) As Byte
          LOBYTE = Intg And &HFF&
End Function
```

Për të përcaktuar distancën që përshkon roboti thirret funksioni DistanceSensorRoomba()
i cili kthen mbrapsh metrat e përshkruara.

Për të vënë robotin Roomba në gjendjen beep (tingull zanor) ekzekutohet komanda :
`BeepRoombaVB6VBNet ().`

Për të marë informacion mbi nivelin e baterive të robotit ekzekutohen komandat :
`BatteryChargeReaderRoombaVB6VBNet()` ose `BatteryVoltageRoombaVB6VBNet()` të cilat kthejnë vlerën e tensionit të paketës burim.

Realizimi eksperimental i komunikimit të PC me iRobot Create.

Për të realizuar komandimin e robotit programi i ndërtuar në Visual Basic 6.0 përbëhet nga një formë (dritare) e paraqitur në Fig 5 dhe prej 12 modulesh ku secili prej tyre përmban komanda , funksione që lidhen me komunikimin e robotit si dhe funksione ndihmëse të tjera. Funksionimi i programit është i thjeshtë. Në dritaren e portës së komunikimit (Fig 5.22) zgjidhet numri i portës seriale. Në rastin kur PC lidhet me robotin me ndihmën e kabllit serial, në portën e komunikimit zgjidhet COM1 ose COM2. Kur lidhjen midis PC dhe robotit e kryejmë në mënyrë valore atëherë në portën e komunikimit zgjidhet COM3 – COM8 (kjo zgjedhje përcaktohet nga bluetooth pas instalimit të saj).

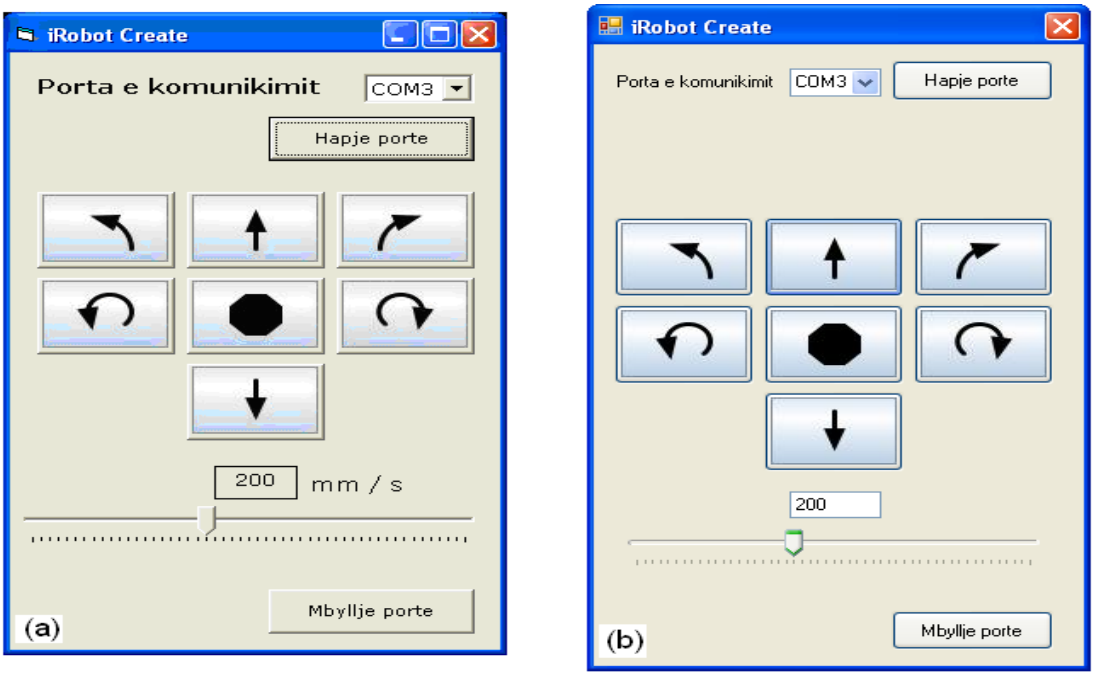


Fig 5.22. *Programi i komandimit te levizjes se robotit (a)-VB6, (b)-VB .NET*

Si përfundim :

- Nëpërmjet këtij paragrafi përshkruam se si mund të realizohet komandimi i robotit prej portës seriale duke shfrytëzuar librarinë e komandave të ndërtuar për këtë qëllim.
- Programi i ndërtuar tregon mënyrën se si mund të shfrytëzohet ActiveComport Serial Port në Visual Basic 6.0 dhe Visual Basic .NET.
- Nëpërmjet programit të ndërtuar në Visual Basic 6.0 dhe Visual Basic .NET tregohet se si mund të thirren komandat për vënien e robotit në lëvizje drejtvizore dhe rrotulluese në mënyrë të kontrolluar.

Në Shtojcën 10 po japim librarinë e komandimit të lëvizjeve të robotit në Visual Basic 6.0 dhe Visual Basic .NET bazuar në librarinë ActiveComport Serial Port.

5.6. Specifikime dhe libraria e komandave në Basic4Androide (Java). Komunikimet me kabëll (wire) dhe me valë (wireless and bluetooth)

Për komandimin e robotit prej PC është e nevojshme shfrytëzimi i portës seriale (7 pin Mini - DIN connector) ose i portës DB 25 pin ku montohet paketa Bluetooth (Fig 5.23). Programet që përbëjnë librarinë e ndërtuar paraqiten nga një grup komandash dhe funksionesh të cilat realizojnë detyra specifike. Kjo librari do të shërbejë si një urë lidhëse ndërmjet robotit dhe përdoruesit kur ai programon detyra specifike në Basic4Androide, në dërgimin e komandave drej robotit. Komandat përgjithësisht venë robotin në veprim lëvizjeje ndërsa funksionet sjellin tek përdoruesi vlerat e madhësive që karakterizohen si parametra të robotit si psh madhësia e tensionit të baterive, gjendjet e sensorëve të robotit, rrugën e kryer prej tij etj.

Paisjet fizike në realizimin e lidhjes midis paisjes Android dhe robotit.

Dërgimi i instruksioneve dhe komandave drej robotit do të realizohet fizikisht nëpërmjet paisjes së komunikimit bluetooth e cila në paisjen Androide është e inkorporuar ndërsa në iRobot Create lidhet në portën 25 pin (DB 25 pin connector) ku montohet paketa bluetooth i quajtur Element Direct BAM (Bluetooth Adaptor Module) (Fig 5.23). Në paisjen Android hapi i parë është deklarimi i një objekti serial i cili do të deklarohet si një objekt global. Pas inicializimit të objektit serial kërkohet lidhja në paisjen robotike.

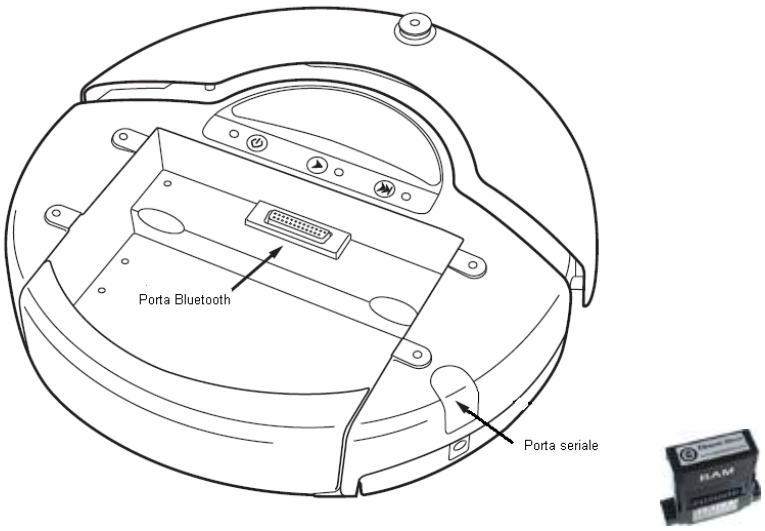


Fig 5.23. *iRobot Create dhe elementi BAM*

Programet lidhëse midis paisjes Android dhe robotit.

Komandimi i iRobot Create kryhet nëpërmjet një programi të thjeshtë i cili përbëhet nga një dritare ku janë vendosur disa butona me simbolikat e lëvizjeve më të mundshme të robotit.
Pas cdo butoni që klikohet ekzekutohen komanda dhe funksione të cilat janë në përbërje të librarisë së ndërtuar në Basic4Androide [40]. Për realizimin e komunikimeve me Roomba 4400 në këtë material do të shfrytëzohet komunikimi bluetooth, i cili shfrytëzon librarinë Androide 2.0 (API level 5) ose versione më lart. Programi që realizon këto komunikime i njohur si Basic for Androide

Programimi me librarinë SERIALE ANDROID 2.0.

Problemi i komunikimit nëpërmjet portës seriale edhe pse është një problem i trajtuar në shumë materiale, përsëri ka specifikat e veta. Realizimi i komunikimit duke qenë kompleks ka bërë që kompani të specilizuara të ndërtojnë programe ndihmëse që shërbejnë si ura lidhëse midis programuesve dhe portës seriale. Përdorimi i librarisë seriale Androide 2.0 është me qëllime të ndryshme si psh komandimi i paisjeve industriale të ndryshme që mund të komunikojnë me paisjet Android nëpërmjet portës seriale dhe Bluetooth [40]. Kjo siguron që përdorimi i teknikave të menaxhimit të portave seriale të jetë i rëndësishëm.

Për të krijuar një objekt serial dhe për të kryer inicializimet e nevojshme duke përdorur librarinë seriale Androide 2.0 janë të domosdoshme linjat e mëposhtme :

```
Sub Process_Globals
    Dim admin     As BluetoothAdmin
    Dim serial1   As Serial              ' deklarim i objktit serial
    Dim out       As OutputStream
    Dim Inp       As InputStream
End Sub
Sub Activity_Create(FirstTime As Boolean)
      Activity.LoadLayout("bluetooth")
      If FirstTime Then
            admin.Initialize("admin")
            serial1.Initialize("serial1")
      End If
End Sub
```

Objekti serial i krijuar duhet të konfigurohet dhe kryhet hapja e portës së komunikimit. Për rastin konkret për komunikimin me iRobot Create, konfigurimi i tij përcaktohet më poshtë. Pas klikimit të butonit `btnSearchForDevices`, nis procesi i identifikimit të paisjeve që mund të lidhen me paisjen Android. Në ekran të paisjes Android shfaqet lista e mundësive të lidhjeve nga e cila përdoruesi zgjedh atë me emërtim Element Direct BAM [39]. Kjo e fundit përfaqëson lidhjen bluetooth me iRobot Create.
Komanda `serial1.Connect(connectedDevice.Mac)` realizon lidhjen me paisjen me emër `connectedDevice.Name` dhe mac adresë `connectedDevice.Mac`

```
Sub btnSearchForDevices_Click
      foundDevices.Initialize
      If admin.StartDiscovery    = False Then
         ToastMessageShow("Error starting discovery process.", True)
      Else
         ProgressDialogShow("Searching for devices...")
      End If
End Sub
Sub Admin_DiscoveryFinished
```

```
        ProgressDialogHide
        If foundDevices.Size = 0 Then
           ToastMessageShow("Nuk gjendet paisje per lidhje.", True)
        Else
               Dim l As List
               l.Initialize
               For i = 0 To foundDevices.Size - 1
                      Dim nm As NameAndMac
                      nm = foundDevices.Get(i)
                      l.Add(nm.Name)
               Next
               Dim res As Int
               res = InputList(l, "Zgjidh paisjen per lidhje", -1)
               If res <> DialogResponse.CANCEL Then
                      connectedDevice = foundDevices.Get(res)
                      ProgressDialogShow("Lidhja realizuar tek : " &
                      connectedDevice.Name & " (" & connectedDevice.Mac & ")")
                      serial1.Connect(connectedDevice.Mac)
               End If
        End If
End Sub
```

Konfirmimi i lidhjes përcaktohet nga pjesa e kodeve në vazhdim :

```
Sub Serial1_Connected (Success As Boolean)
        ProgressDialogHide
        Log("Konfirmi lidhjeje : " & Success)
        LogMessage("", " Konfirmi lidhjeje : " & Success)
        If Success = False Then
           Log(LastException.Message)
           ToastMessageShow("Gabim lidhjeje : " & LastException.Message, True)
           SuccessConnection=False
        Else
           out = serial1.OutputStream
           Inp = serial1.InputStream
           SuccessConnection=True
        End If
End Sub
```

Për të kryer procesin e dërgimit të datave në portë komanda e përdorur është si më poshtë :

```
out.WriteBytes(wByte,0,uBound(wByte)-1)
```

Instruksioni `out.WriteBytes(wByte,0,uBound(wByte)-1)` dërgon në portë `uBound(wByte)-1` byte informacion (`wByte`). Në rastin e komunikimit me iRobot Create instruksioni që do të përdoret për dërgimin e datave në portë do të jetë dërgimit byte pas byte. Instruksionet do të jenë si më poshtë :

```
Dim bufferarray() As Byte = Array As Byte(128)
Main.out.WriteBytes(bufferarray,0,1)
```

Për të kryer procesin e marrjes së të dhënave, të dërguara nga roboti iRobot Create, përdoren instruksionet e mëposhtme:

```
If Main.inp.BytesAvailable >0 Then
   Dim buffer(Main.inp.BytesAvailable) As Byte
   Main.inp.ReadBytes( buffer,0, Main.inp.BytesAvailable )
```

```
   s = BytesToString(buffer,0,buffer.Length ,"Windows-1253")
   Msgbox( s,"")
End If
```

Mbyllja e portës seriale të komunikimit kryhet nga komanda :

```
serial1.Disconnect                              ‘ Komanda per mbylljen e portes
```

Mbështetur në sa u përshkruan më sipër po jepen në vijim disa komanda dhe funksione që do të përdoren për të vënë në lëvije iRobot Create.

Komandat e lëvizjes dhe funksionet e gjëndjes së robotit.

Disa nga komandat dhe funksionet më të përdorëshme do të përshkruhen më poshtë.
Hapja e lidhjes për komunikimin me robotin kryhet nga funksioni :

```
BeepRoombaConnect
```

Ndërtimi i këtij funksioni jepen në vijim :

```
Public Sub BeepRoombaConnectB4A() as Boolean
       Dim boolBeepRoombaConnectB4A     as Boolean

       If Main.out.IsInitialized = True Then

              Dim bufferarray () As Byte

              Bufferarray(0) = 128: Main.out.WriteBytes(Bufferarray,0,1)
              modMAin.Sleep(100)
              Bufferarray(0) = 132: Main.out.WriteBytes(Bufferarray,0,1)
              modMAin.Sleep(100)
              Bufferarray(0) = 139: Main.out.WriteBytes(Bufferarray,0,1)
              Bufferarray(0) = 25: Main.out.WriteBytes(Bufferarray,0,1)
              Bufferarray(0) = 0: Main.out.WriteBytes(Bufferarray,0,1)
              Bufferarray(0) = 128: Main.out.WriteBytes(Bufferarray,0,1)
              modMAin.Sleep(100)
              Bufferarray(0) = 140: Main.out.WriteBytes(Bufferarray,0,1)
              Bufferarray(0) = 1: Main.out.WriteBytes(Bufferarray,0,1)
              Bufferarray(0) = 48: Main.out.WriteBytes(Bufferarray,0,1)
              Bufferarray(0) = 20: Main.out.WriteBytes(Bufferarray,0,1)
              modMAin.Sleep(100)
              Bufferarray(0) = 141: Main.out.WriteBytes(Bufferarray,0,1)
              Bufferarray(0) = 1: Main.out.WriteBytes(Bufferarray,0,1)
              modMAin.Sleep(100)

              boolBeepRoombaConnectB4A = True
       else
              boolBeepRoombaConnectB4A = False
       End If

       RETURN boolBeepRoombaConnectB4A
End Sub
```

Funksioni kthen mbrapsh vlerat True ose False për rastet kur lidhja realizohet me robotin ose jo.

Lëvijet para, prapa si dhe majtas, djathtas me shpejtësi lineare të përcaktuara për secilën rrotë të robotit jepet nga komanda :

```
SetDriveWheelsCreateB4A ShpejtesiaRrotesMajtas, ShpejtesiaRrotesDjathtas
```

Ku parametrat `ShpejtesiaRrotesMajtas` dhe `ShpejtesiaRrotesDjathtas` përcaktojnë shpejtësitë e rrotës së majtë dhe të djathtë të robotit (Fig 5.24). Shpejtësia lineare e rotave të robotit përcaktohet në vlerat (-500 – 500 mm/s) [36].

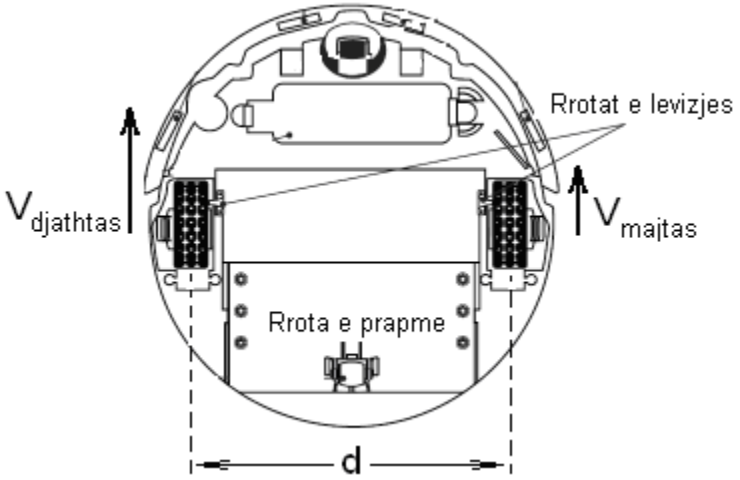


Fig 5.24. Rrotat e iRobot Create dhe shpejtesitë lineare të tyre

Kur kërkohet lëvizja perpara e robotit pas përcaktimit të shpejtësisë me të cilën ai do të lëvizë, mjafton të ekzekutojmë komandën :

```
SetDriveWheelsCreateB4A vLevisjeje, vLevisjeje     'Shpejtesia  vLevisjeje =[0, 500] mm/s
```

Për lëvizjen prapa do të ekzekutonim komandën :

```
SetDriveWheelsCreateB4A vLevisjeje, vLevisjeje     'Shpejtesia   vLevisjeje  =[-500  ,  0]
mm/s
```

Rrotullimet me 90 gradë majtas dhe djathtas do të kryheshin nga ekzekutimi i komandave :

```
SetDriveWheelsCreateB4A 0, vLevisjeje       'Shpejtesia  vLevisjeje =[0 , 500] mm/s
SetDriveWheelsCreateB4A vLevisjeje, 0       'Shpejtesia  vLevisjeje =[0 , 500] mm/s
```

Rrotullimi me 180 gradë majtas dhe djathtas do të kryheshin nga ekzekutimi i komandave :

```
SetDriveWheelsCreateB4A - vLevisjeje, vLevisjeje   'Shpejtesia vLevisjeje =[0 , 500] mm/s
SetDriveWheelsCreateB4A vLevisjeje, - vLevisjeje   'Shpejtesia vLevisjeje =[0 , 500] mm/s
```

Duke njohur distancën midis rrotave të robotit si dhe shpejtësitë lineare të rrotave jemi në gjendje të përcaktojmë rrotullime në kënde të dëshiruara të sistemit robotik. Për këtë do të shfrytëzojmë disa

formula të tjeshta të fizikës. Problemi do të shtrohej në formën : sa duhet të jetë shpejtësia lineare e lëvizjes së rrotave në momentin kur kërkohet rrotullimi i robotit me këndin α ?
Duke u mbështetur në formula të njohura që lidhin shpejtësinë këndore me atë lineare dhe shpejtësine lineare me kohën mund të shkruajmë :

$$\omega = \frac{v}{r} = \frac{2 \cdot v}{d} \qquad (1)$$

$$\omega = \frac{\alpha}{t} \qquad (2)$$

Nisur nga 1 dhe 2 mund të llogaritim shpejtësinë lineare të rrotës për një rrotullim me këndin α të robotit gjatë kohës t .

$$\alpha = \frac{2 \cdot v \cdot t}{d} \qquad (3)$$

Ku d është distanca midis rrotave të robotit Fig 5.25.

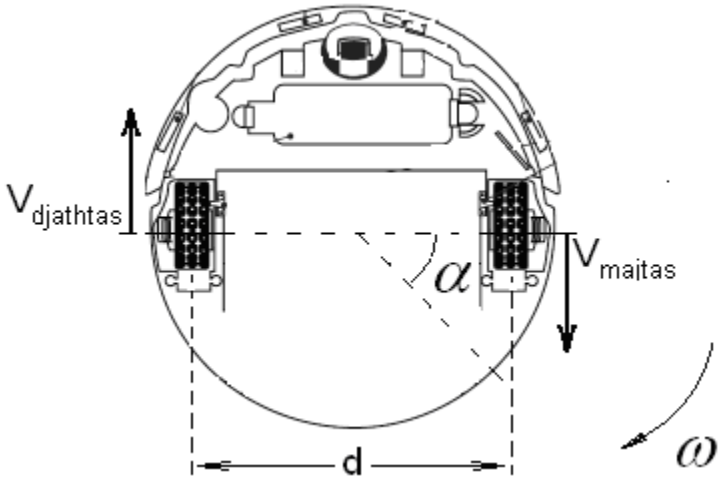


Fig 5.25. *Rrotullimi i iRobot Create*

Një komandë për lëvizje të përbërë në formë harku rrethi për iRobotCreate jepet si më poshtë :

```
SetFwdVelRadiusRoombaB4A  vLevisjeje, rRezja
```

ku `vLevisjeje` është shpejtësia lineare e robotit dhe `rRezja` është rrezja e trajektores që do të përshkruajë roboti (Fig 5.26).

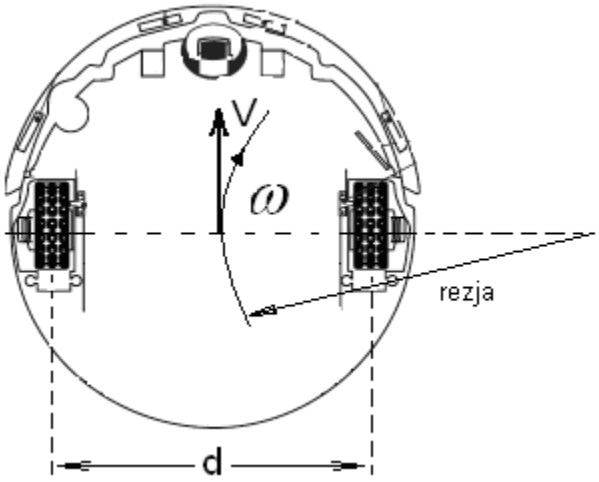


Fig 5.26. *Lëvizja e përbërë e iRobot Create*

Për të ndaluar robotin mund të përdoren komandat :

```
SetDriveWheelsCreateB4A  0, 0        ose    SetFwdVelRadiusRoombaB4A 0,1
```

Për të dy komandat e përshkruara më sipër madhësitë `vLevisjeje`, `rRezja` janë numra nga 16 bit pra 4 byte në total. Për rastin kur shpejtësia do të jetë -200 mm/s dhe rezja 500 mm, informacioni që do të dërgohet në portë do të jetë i renditur në byte si në vijim :

```
[137]  [255] [56]  [1] [244]
```

Komanda `[137]` i vendos rrotat e robotit në kontroll për një lëvizje në formë harku.
Vargu i mësipërm përfitohet nga zbërthimi i madhësive sipas formës së mëposhtme [38] :

```
vLevisjeje = -200 = hex FF38 = [hex FF] [hex 38] = [255] [56]
rRezja = 500 = hex 01F4 = [hex 01] [hex F4] = [1] [244]
```

Për të përfituar zbërthimin e mësipërm në Basic4Androide [37] do të shfrytëzojmë funksionet `HIBYTE (Intg)` dhe `LOBYTE (Intg)` të përshkruara më poshtë :

```
Public Sub HIBYTE( Intg As Int) As Int
       Dim andBit    As Int
       andBit = Bit.AND(Intg ,0xFF00)
       andBit = andBit / 256
    Return andBit
End Sub

Public Sub LOBYTE( Intg As Int) As Int
   Dim andBit As Int
   andBit = Bit.AND(Intg ,0xFF)
   Return andBit
End Sub
```

Për të përcaktuar distancën që përshkon roboti thirret funksioni `DistanceSensorRoombaB4A()` i cili kthen mbrapsh metrat e përshkruara.

Për të vënë robotin Roomba në gjendjen beep (tingull zanor) ekzekutohet komanda :

```
BeepRoombaB4A().
```

Për të marë informacion mbi nivelin e baterive të robotit ekzekutohen komandat :
`BatteryChargeReaderRoombaB4A()` ose `BatteryVoltageRoombaB4A()` të cilat kthejnë vlerën e tensionit të paketës burim.

Realizimi eksperimental i komunikimit të paisjes Androide me iRobot Create.

Për të realizuar komandimin e robotit programi i ndërtuar në Basic4Androide Version 2.52 përbëhet nga nje forme (dritare) e paraqitur në Fig 5.27 dhe prej 8 modulesh ku sejcili prej tyre përmban komanda, funksione që lidhen me komunikimin e robotit si dhe funksione ndihmëse të tjera.

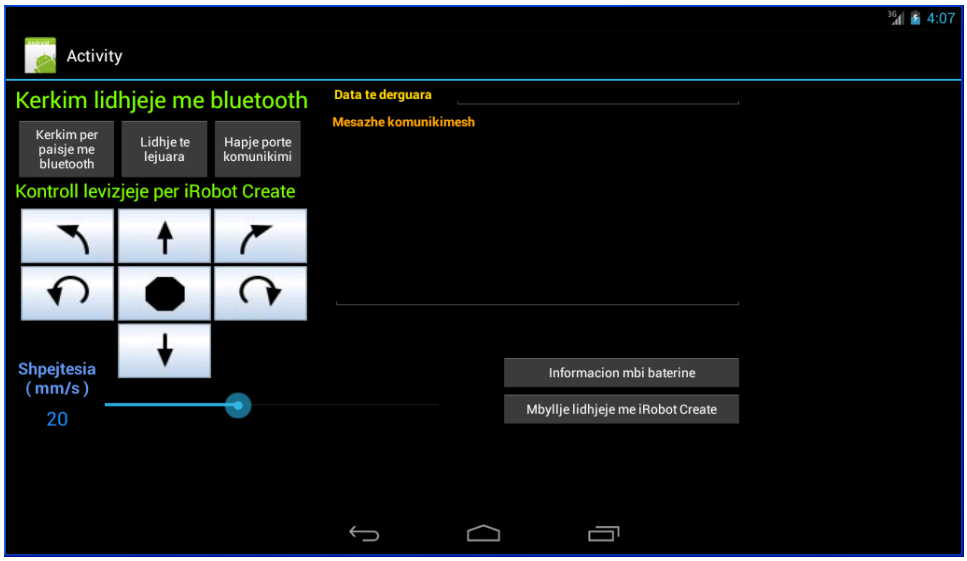


Fig 5.27. *Programi i komandimit të lëvizjes së robotit Roomba 4400 (iRobot Create)*

Funksionimi i programit është i thjeshtë. Në dritaren e portës së komunikimit (Fig 5.27) klikohet në butoni 'Kërkim per paisje me bluetooth' pas së cilës shfaqet pas një kërkimi lista e paisjeve me të cilat mund të kërkojmë komunikim (a). Pas klikimi të butonit 'Hapje porte komunikimi' sigurohet kërkesa për lidhje me paisjen e zgjedhur në listën (a). Lëvizja e robotit më tej bëhet duke punuar me butonat që si pamje të tyre kanë drejtimin e lëvizjes.

Si përfundim mund të themi se :

- Nëpërmjet këtij paragrafi përshkruam se si mund të realizohet komandimi i robotit prej portës virtuale seriale bluetooth duke shfrytëzuar librarinë e komandave të ndërtuar për këtë qëllim.
- Programi i ndërtuar tregon mënyrën se si mund të shfrytëzohet librarina seriale Androide 2.0 në Basic4Androide Version 2.52.
- Nëpërmjet programit të ndërtuar në Basic4Androide Version 2.52 tregohet se si mund të thirren komanadat për vënien e robotit në lëvizje drejtvizore dhe rrotulluese në mënyrë të kontrolluar.

Në Shtojcën 11 po japim librarinë e komandimit të lëvizjeve të robotit në Basic4Androide bazuar në librarinë seriale Androide 2.0.

Si përfundim të kapitullit mund të themi se :

Realizuam programet, pra pjesën software të përbehet nga dy pjesë :

- Programi i komandave, i cili shpreh detyrën që duhet të kryhet, nëpërmjet një algoritmi të caktuar.
- Programet e komunikimit me robotin që përfaqësojnë një librari funksionesh dhe komandash, të ndërtuara për mjedise të ndryshme programimi dhe konkretisht për mjediset Matlab, Visual Basic 6.0, Visual Studio 2005, 2008, 2010, 2012 .Net dhe Basic4Androide (Java)

KAPITULLI VI

Hyrje

iRoboti Create është një model i programueshëm i versionit Roomba Model 4400 që përdoret për qëllime mësimi dhe studimi. Për realizimin e kontrollit të pozicionit të këtij roboti prej PC është shfrytëzuar porta DB 25 pin ku montohet moduli bluetooth BAM, dhënë në Fig 6.1. Një sistem point lazer është i instaluar poshtë pllakës të mbështetur në katër shufrat vertikale mbajtëse të konstruksionit.
Dërgimi i komandave prej kompjuterit drejt robotit do të kryhet nëpëmjet paisjeve bluetooth të montuara në kompjuter dhe robot. Në robot është instaluar një kamera (ose dy të tilla) Wi-Fi si në Fig 6.1, e cila do të japë imazhe digitale që do të përpunohen në kompjuter prej nga do të formulohen dhe komandat për drejtimin e robotit drejt pozicionit të dëshiruar.
Programi në kompjuter në mënyrë të përmbledhur realizon këto detyra :

- përpunon imazhet grafike të dërguara prej kamerës Wi-Fi.
- formulon komandat për robotin.
- dërgon komandat për ekzekutim nëpërmjet paisjeve Bluetooth në robot.

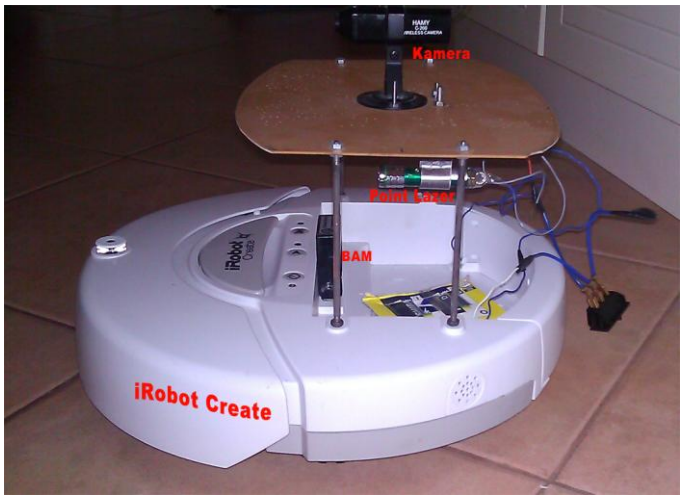


Fig 6.1. iRobot Create, kamera, gjeneruesi lazer dhe elementi BAM

Detyra që mund të kryhen nëpërmjet një sistemi të tillë janë të shumta. Ndër përdorimet më të spikatura prej shfrytëzimit të sistemeve të tilla mund të përmendim disa të cilat po i paraqesim në vijim.

6.1. Përdorimi i sistemit robotik iRobot Create në fusha të ndryshme sociale.

Disa prej mundësive të përdorimit të sistemit robotik iRobot Create në fusha të ndryshme sociale po i përshkruajmë si më poshtë :

- Komandimi prej kompjuterit me kabëll ose valor (bluetooth) i iRobot Create si një sistemi robotik vëzhgimi dhe komunikimi Fig 6.3.

Fig 6.2 *Përdorimi i iRobot Create si një sistem robotik vëzhgimi.*

- Përdorimi i sistemit robotik për rastet kur ndërtesat janë të dëmtuara nga tërmetet dhe njerëzit e kanë të pamundur të hyjnë në këto objekte.

- Komandimi i robotit në distancë dhe marrja e pamjeve të kamerave të montuara mbi robot, në kompjuter Fig 6.3.

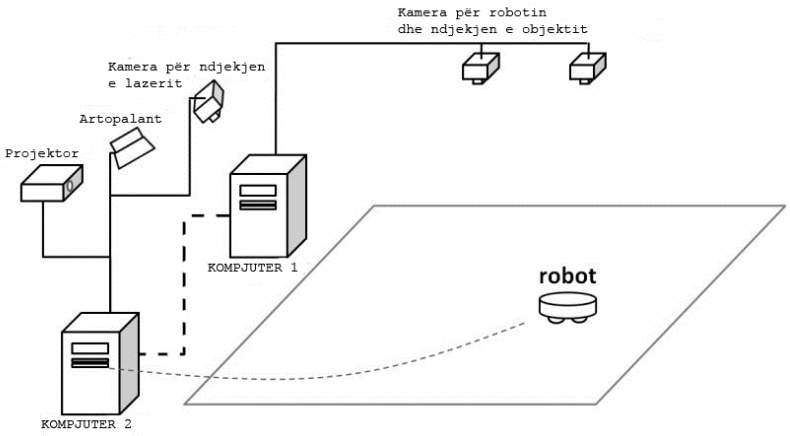


Fig 6.3 *Skema bazë për përdorimin i iRobot Create si një sistem robotik vëzhgimi dhe komunikimi.*

- Ndjekja e objekteve dhe kontrolli i distancës prej tyre si objekte në vëzhgim.

Në pjesën në vijim jepet ndjekja në rrafshin plan të një objekti qëllim, prej iRobot Create sipas metodës klasike të ndjekjes përcaktuar nga shpejtesia e lëvizjes së objektit që ndiqet.

6.2. Ndjekja e një objekti prej iRobot Create sipas metodave klasike të ndjekjes përcaktuar nga shpejtësia e lëvizjes së objektit.

Qëllimi i këtij paragrafi është dhënia e disa informacioneve mbi një mënyrë kontrolli pozicioni për robotin Roomba Model 4400 (iRobot Create) me ndihmën e një ose dy kamerash vëzhgimi. Qëllimi është realizimi i një sistemi (robot & kamera) si në Fig 6.1 i cili përpiqet të ndjekë në një distancë të pa ndryshuar një objekt të paracaktuar, si në Fig 6.4. Objekti mund të jetë një sferë të vogël me ngjyrë të cilësuar apo një diodë ndricuese led, montuar në një objekt të lëvizshëm si mbi një robot tjetër.

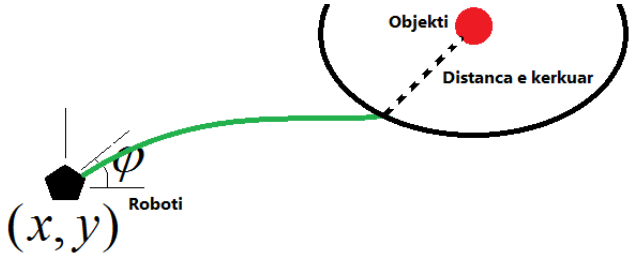


Fig 6.4. *Ndjekja e objektit duke ruajtur një distancë të kërkuar*

Për të gjtha sa u tha deri tani, qëllimi përfundimtar është realizimi i një sistemi inteligjent ndjekës.

Baza teorike.

Për të realizuar kontrollin e robotit është e nevojshme të njihet modeli matematik i tij. Për rastin e robotit në përdorim, duke qenë se ai ka dy rrota të pavarura në lëvizje nga njëra-tjetra (sejcila prej

tyre vihet në lëvizje nga një motor me hapa i veçantë), si model matematik mund të shfrytëzohet ai i robotit me rrota të diferencuara. Tipi i lëvizjes së robotit është i kushtëzuar nga shpejtësia e rrotave të tij. Konkretisht, duke i rrotulluar rrotat me shpejtësi të njëtë, roboti lëvizë në vijë të drejtë (lëvizje drejtvizore). Nëse një rrotë rrotullohet me shpejtësi më të vogël se tjetra, roboti do të rrotullohet në anën e rrotës me shpejtësi më të vogël. Rasti i fundit i lëvizjes është rasti kur një rrotë rrotullohet në kahje të kundërt me rrotën tjetër. Në këtë rast roboti rrotullohet në formë spirale.
Për modelimin matematik më të thjeshtë të robotit, në lëvizjen 2D, është e nevojshme njohja e vetëm dy parametrave konstruktivë, distanca ndërmjet rrotave d dhe rrezja e rrotave r.
Për të kontrolluar pozicionin e rrobotit në planin 2D është e nevojshme të kontrollojmë tek roboti shpejtësitë e rrotave v_{majtas} dhe $v_{djathtas}$. Dy hyrje (inpute) si parametra të kontrollit për robotin do të jenë shpejtësitë e rrotave v_{majtas} dhe $v_{djathtas}$, dhe dalje (output) janë koordinatat x, y dhe këndi i rrotullimit φ (shpejtësi këndore ω). Modeli matematik do të realizojë lidhjen midis variablave të hyrjes dhe të daljes.

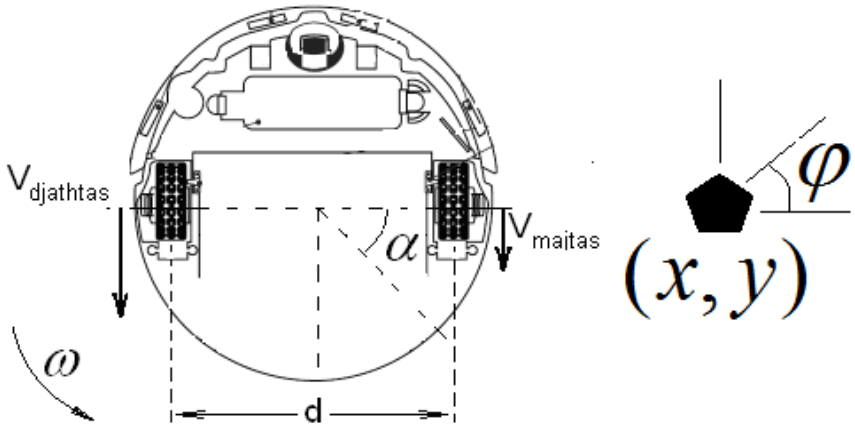


Fig 6.5. *Rrotullimi i iRobot Create dhe parametrat për pozicionimin në 2D.*

Madhësitë x, y dhe këndi i rrotullimit φ, përbëjnë variablat e gjendjes së sistemit.
Kinematika e lëvizje së rrobotit do të përshkruhej nga ekuacionet e mëposhtëme :

$$\dot{x} = \frac{r}{2} \cdot (v_{majtas} + v_{djathtas}) \cdot \cos(\varphi)$$
$$\dot{y} = \frac{r}{2} \cdot (v_{majtas} + v_{djathtas}) \cdot \sin(\varphi) \qquad (1)$$
$$\dot{\varphi} = \frac{r}{d} \cdot (v_{djathtas} - v_{majtas})$$

Në rastin e lëvizjes së një pike materiale (qëndra e gravitetit e një trupi) me shpejtësi v dhe kënd rrotullimi φ (shpejtësi këndore ω), ekuacionet e kinematikës do të jenë :

$$\begin{aligned} \dot{x} &= v \cdot \cos(\varphi) \\ \dot{y} &= v \cdot \sin(\varphi) \\ \dot{\varphi} &= \omega \end{aligned} \qquad (2)$$

Duke qenë se robotin ne mund ta trajtojmë në disa raste si një pikë mase, duke barazuar anë për anë ekuacionet (1) dhe (2) kemi relacionet e mëposhtëme :

$$\begin{aligned} v &= \frac{r}{2} \cdot (v_{majtas} + v_{djathtas}) \\ \omega &= \frac{r}{d} \cdot (v_{djathtas} - v_{majtas}) \end{aligned} \qquad (3)$$

Prej ekuacioneve (3) mund të përcaktojmë v dhe ω në funksion të d , r , v_{majtas} dhe $v_{djathtas}$.

$$\begin{aligned} v_{djathtas} &= \frac{1}{r} \cdot (v + \frac{d}{2} \cdot \omega) \\ v_{majtas} &= \frac{1}{r} \cdot (v - \frac{d}{2} \cdot \omega) \end{aligned} \qquad (4)$$

Ekuacionet (4) tregojnë mardhënien ndërmjet shpejtësisë lineare të rrotave të robotit dhe shpejtësisë lineare dhe këndore të qendrës së masës v dhe ω.

Për llogaritjen e pozicionit të robotit $x, y, \varphi(orientimi)$ në 2D, kur rrotat e tij kanë shpejtësitë lineare v_{majtas} dhe $v_{djathtas}$, do të shfrytëzojmë kalkulimet e mëposhtëme. Gjatë intervalit kohor Δt, secila rrotë gjatë lëvizjes përshkruan një hark. Shënojmë D_{rm} gjatësinë e harkut të kryer nga rrota e majtë dhe D_{rd} është gjatësia e harkut e kryer nga rrota e djathtë. Qëndra e robotit do të përshkruajë gjatësinë e harkut D_c. Llogaritja e D_c bëhet sipas formulës së mëposhtëme :

$$D_c = \frac{D_{rm} + D_{rd}}{2} \qquad (5)$$

Duke njohur gjendjen e vjetër $x_v, y_v, \varphi_v(orientimi)$ si dhe distancën D_c të kryer nga qendra e robotit gjatë intervalit kohor Δt, mund të njihen vlerat e reja të gjendjes së robotit $x_r, y_r, \varphi_r(orientimi)$ në fund të këtij intervali kohe**.**

$$\begin{aligned} x_r &= x_v + D_c \cdot \cos(\varphi) \\ y_r &= y_v + D_c \cdot \sin(\varphi) \\ \varphi_r &= \varphi_v + \frac{D_{rl} - D_{rd}}{d} \end{aligned} \qquad (6)$$

Problemin e shfrytëzimit të të dhënave për të përcaktuar pozicionin e objektit e trajton odometria.

Odometria

Odometria është përdorimi i të dhënave të përftuar nga sensorët, për të përcaktuar ndryshimin e pozicionit të objektit në varësi të kohës. Nëpërmjet odometrisë mund të marrim informacione mbi pozicionimin e robotit në hapësirën dy dimensionale (2D) pra koordinatat $x, y, \varphi(orientimi)$ ose tre dimensionale (3D) pra koordinatat x, y, z dhe për orientimin φ, r **.**

Për marrjen e këtij informacioni duhen sensorët të cilët mund të jenë të jashtëm ose të brendshëm. Sensorë të jashtëm, janë sensorët që japin informacion për mjedisin rrethues të robotit, si p.sh distancën e robotit prej një objekti ose distancën nga destinacioni qëllim. Si të tillë mund të përmendim sensorët ultratinguj, infrared, vizualë (kamerat), skanerat me lazer, pajisjet GPS etj.
Në grupin e sensorëve të brendshëm futen sensorët që japin informacion për gjendjen faktike të robotit, pozicionin dhe orientimin e tij. Të tillë sensorë janë akselerometrat, xhiroskopët, enkoderët.

Realizimi praktik.

Objekti në vëzhgim i treguar në Fig 6.6 dhe Fig 6.7 paraqet një trup sferik të vogël me ngjyrë të kuqe i vendosur mbi një objekt të lëvizshëm si psh mbi një tjetër robot. Në këtë rast problemi do të shtrohej në ndjekjen e një roboti prej një roboti të dytë. Për kamerën dhe objektin në vëzhgim njihen lartësitë prej bazës D_1 dhe D_2 (Fig 6.8). Në këto kushte mund të themi $h = D_1 - D_2$.

Zgjidhja e problemit do të realizohet në disa etapa :

1. Lokalizohet objekti që ndiqet në imazhin e sjellë prej kamerës.
2. Rrotullim i sistemit robotik duke sjellë vendosjen e objektit në vëzhgim në mes të imazhit (llogaritja e këndit të rrotullimit dhe shpejtësisë këndore të rrotullimit)
3. Përcaktimi i distancës nga objekti në vëzhgim.
4. Vlerësimi i ndryshimit midis vlerës së kërkuar dhe asaj faktike.
   Llogaritja e masës së zhvendosjes kundrejt objektit në vëzhgim si dhe e shpejtësisë lineare të zhvendosjes.
5. Formulimi i komandave për robotin që ai të pozicionohet sipas kërkesës.

Sejcilën prej etapave po e trajtojmë në vazhdim.

1. Për të kryer lokalizimin e objektit në vëzhgim në imazhin e sjellë prej kamerës mund të përdoren metoda të ndryshme në funksion të problemit konkret që shtrohet. Në rastin tonë objekti që ndiqet nuk është një trup i rastësishëm por një objekt i targuar i cili është i dallueshëm prej mjedisit përreth psh me ngjyrë të kuqe në një mjedis të çelët. Duke qenë i tillë lokalizimi i tij në imazhin e kamerës është një problem më i thjeshtë. Pas lokalizimit përcaktohet qendra gjeometike e drejkëndëshit që përfaqëson dhe koordinatat e qendrës së objektit si në Fig 6.6.

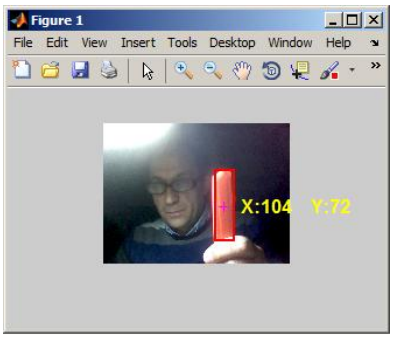


Fig 6.6. *Pamje nga lokalizimi i një objekti brenda imazhit të kamerës, në Matlab.*

2. Në Fig 6.7 jepen tre pamje të njëpasnjëshme të mara prej kameras në kohët t_0 , t_1 dhe t_2 . Gjatë intervalit kohor $\Delta t = t_2 - t_0$, objekti në vëzhgim zhvendoset me madhësinë :

$$\Delta pfch = pfch_2 - pfch_0$$

në pixel ku $pfch_2$ jep numrin e pixelëve prej qendrës sipas drejtimit horizontal për pamjen Fig 6.7 - 3, $pfch_0$ jep numrin e pixelëve prej qendrës sipas drejtimit horizontal për pamjen Fig 6.7 - 1.

Shpejtësia lineare e lëvizjes së objekti në vëzhgim do të llogaritet sipas formulës :

$$v_{pixel\_objektit} = \frac{\Delta pfch}{\Delta t}$$

ose (1)

$$v_{pixel\_objektit} = \frac{pfch_2 - pfch_1}{t_2 - t_0}$$

Nisur prej $v_{pixel\_objektit}$ jemi në gjendje të përcaktojmë $v_{cm\_objektit} = v_{pixel\_objektit} \cdot koef_{pixelTOcm}$ si dhe shpejtësinë këndore të rrotullimit të sistemit robotik e cila do të duhet të jetë sa $\varpi_{robotit}$:

$$\varpi_{robotit} = \frac{v_{cm\_objektit}}{D} = \frac{v_{pixel\_objektit} \cdot koef_{pixelTOcm}}{D}$$

ku : D distanca midis kameras dhe objekti në vëzhgim Fig 6.7.

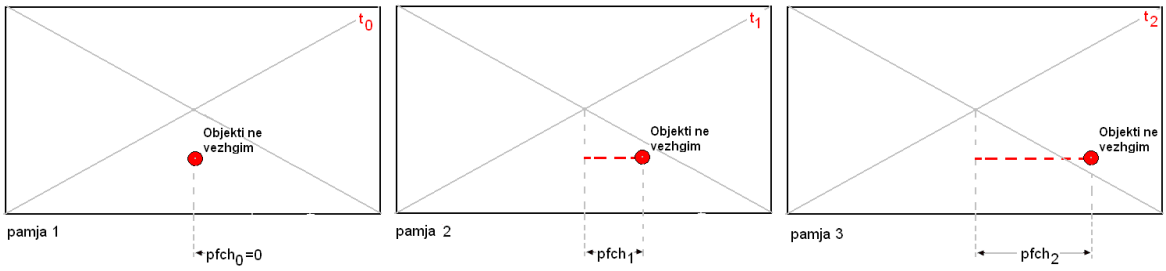


Fig 6.7. *Pamje nga kamera:*
pamja 0 koha $t_0 = 0$, *pamja A koha* t_1 *dhe pamja B koha* t_2 *si dhe objekti në vëzhgim*

Eshtë e kuptueshme se pas çdo pamjeje të marë nga kamera roboti rregullon shpejtësinë këndore, llogaritur sipas formulave të mësipërme duke tentuar të vendosë objektin në vëzhgim në pozicionin e dhënë në pamjen Fig 6.7 - 1.

3. Gjetja e distancës D mund të kryhet duke u bazuar në një nga tre metodat e përshkruara në kapitullin III.
Duke u bazuar në metoda e gjetjes së distancës me ndihmën e kameras dhe një lazeri pikësor të dhënë në Fig 6.8, formula llogaritëse mëposhtëme :

$$D = \frac{h}{\tan(\theta)}$$

Për të përcaktuar distancën e panjohur është e kuptueshme që duhet të njohim këndin θ. Përcaktimi i tij bëhet nëpërmjet formulimeve të mëposhtëme :

$$tg(\theta) = \frac{h}{D}$$

$$\theta = pfc \cdot rpc + r_0$$

ku :

pfc - pika prej qëndrës deri tek projeksioni i objektit.

rpc - radianë për pixel.

r_o - vlerë korrigjuese për mbulimin e gabimit të këndit.

Mund të themi se shprehja llogaritëse përfundimtare do të jetë si më poshtë [6], [7]:

$$pfc \cdot rpc + r_0 = arctg(\frac{h}{D})$$

Përfundimisht mund të shkruajmë :

$$D = \frac{h}{\tan(pfc \cdot rpc + r_0)}$$

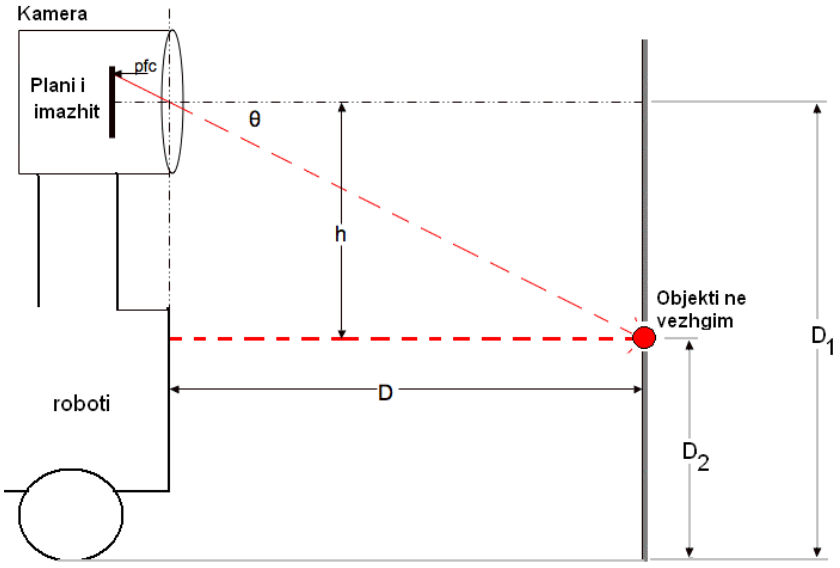


Fig 6.8. *Roboti, kamera, lazeri dhe objekti në vëzhgim.*

Madhësia pfc përcaktohet nga analiza e imazheve te siguruara nga kamera.

Madhësitë rpc dhe r_o përcaktohen në mënyrë eksperimentale duke shfrytëzuar distanca të njohura.

4. Vlerësimi i ndryshimit midis vlerës së kërkuar dhe asaj faktike pas përcaktimit të asaj faktike është një proces i thjeshtë. Madhësia përcaktohet si më poshtë :

$$\Delta D = D_0 - D$$

Ku :

D_0 - vlera e kërkuar

D - vlera e faktike

ΔD - ndryshimi midis vlerës së kërkuar dhe asaj faktike

Vlera ΔD pjesëtuar me kohën e këtij ndryshimi do të japë shpejtësinë lineare të nevojshme të robotit për lëvizjet drejtvizore para – prapa.

$$v_{robotit} = \frac{\Delta D}{\Delta t}$$

ose (2)

$$v_{robotit} = \frac{D_0 - D}{\Delta t}$$

Ku Δt përcaktohet nga intervali kohor midis dy pamjeve të kamerës.

5. Eshtë e kuptueshme se drejtim i levizjes rrotulluese dhe asaj para-prapa në Roomba 4400 do të përcaktohet nga llogaritjet e pikave 1, 2 dhe 3.

Kur do të kërkohet rrotullimi i robotit komanda që do të jepet do të jetë :

```
If ObjectIsLeftRightCenter = majtas Then
   ' levizja e robotit duhet kryhet majtas
    SetDriveWheelsCreate -ShpejtesiaLineare, ShpejtesiaLineare
    HistorikLevizje = majtas
    lMessage.AddItem "Levizja majtas (-) " & ShpejtesiaLineare
ElseIf ObjectIsLeftRightCenter = djathtas Then
   ' levizja e robotit duhet kryhet djathtas
    SetDriveWheelsCreate ShpejtesiaLineare, -ShpejtesiaLineare
    HistorikLevizje = djathtas
    lMessage.AddItem "Levizja djathtas (-) " & ShpejtesiaLineare
ElseIf ObjectIsLeftRightCenter = ndaluar Then
    ' stop the robot
    SetFwdVelRadiusRoomba 0, 1
    HistorikLevizje = ndaluar
    lMessage.AddItem "Qendrim ne vend () " & ShpejtesiaLineare
End If
```

Me shkronja italike janë paraqitur komandat për lëvizjen e robotit.

PAISJET PER REALIZIMIN E EKSPERIMENTIT.

Paisjet që u përdorën për kontrollin e pozicionit të sistemit robotik janë paisje standarte të një niveli normal. Sistemi kompjuterik së bashku me sistemin robotik përbëhet prej :

- Toshiba harman/kardon Laptop
- Bluetooth Adaptor Module for PC
- Roomba Model 4400 (iRobot Create)
- Element Direct BAM (Bluetooth Adaptor Module)
- Kamera Model Hamy M200 Wireless Camera
- Pointer Laser

PROGRAMI DHE FUNKSIONALITETI I TIJ

Për të realizuar komandimin e robotit programi i ndërtuar në Visual Basic 6.0 [4] përbëhet nga nje forme (dritare) e paraqitur në Fig 6.9 dhe prej 12 modulesh ku sejcili prej tyre përmban komanda, funksione që lidhen me komunikimin e robotit si dhe funksione ndihmëse të tjera. Funksionimi i programit është i thjeshtë. Në dritaren e portës së komunikimit (Fig 5) zgjidhet numri i portës seriale. Në rastin kur PC lidhet me robotin me ndihmën e kabllit serial, në portën e komunikimit zgjidhet COM1 ose COM2. Kur lidhjen midis PC dhe robotit e kryejmë në mënyrë valore atëhere në portën e komunikimit zgjidhet COM3 – COM8 (kjo zgjedhje përcaktohet nga bluetooth pas instalimit të saj).
Lëvizja manuale e robotit më tej bëhet duke punuar me butonat që si pamje të tyre kanë drejtimin e lëvizjes.

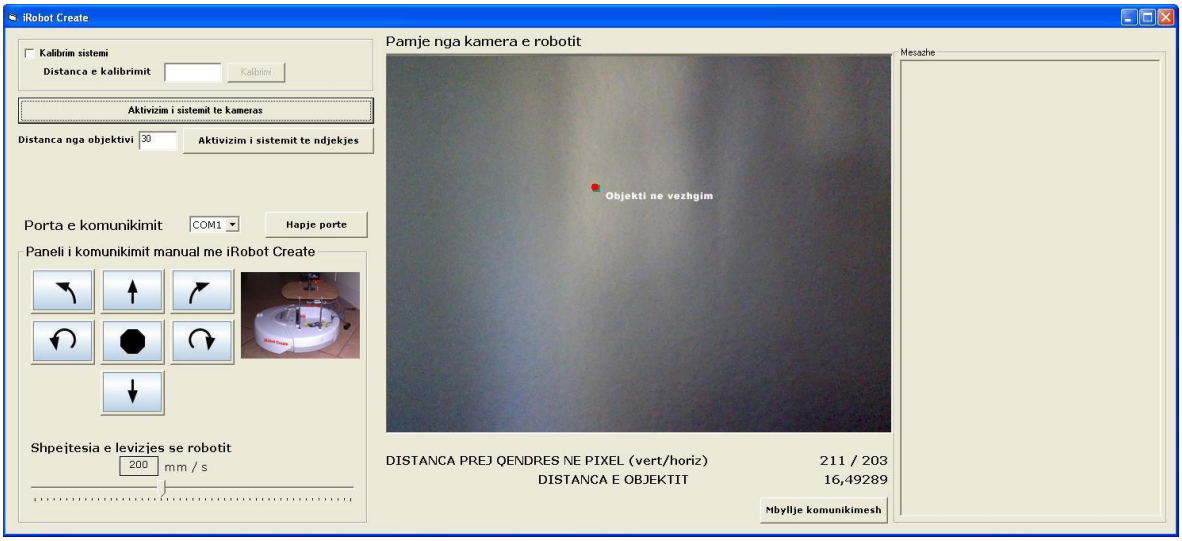


Fig 6.9. *Programi i kontrollit të pozicionit të robotit Roomba 4400 (iRobot Create)*

Funksionimi i programit për kontrollin e distancës është i thjeshtë. Në dritaren e portës së komunikimit (Fig 6.9) klikohet në butoni ‘Aktivizim i sistemit të kameras’ pas së cilës shfaqet pamja e marrë prej kamerës në dritaren e programit. Pas vendosjes së vlerës së kërkuar të distancës

në textbox_in me përshkrimin ‘Distanca nga objektivi’, klikohet te butoni ‘Aktivizim i sistemit të ndjekjes’.

Pas këtij veprimi programi kryhen dy detyra :

- Pozicionon objektin në vëzhgim në mes të pamjes së kamerës duke kryer rrotullime majtas – djathtas në vend.
- Llogarit distancën prej objektit në vëzhgim dhe pas krahasimit me vlerën e kërkuar kryhen lëvizjen para ose prapa të robotit deri sa ndryshimi nga distanca e kërkuar të jetë brenda një kufiri të pranuar si gabim i lejuar.

Modulin kryesor të programit të ndërtuar në Visual Basic 6.0 do e japim në Shtojcën 12.

Si përfundim mund të themi se :

- Nëpërmjet këtij paragrafi u tregua se si mund të realizohet komandimi i robotit prej portës seriale dhe virtuale seriale bluetooth duke shfrytëzuar librarinë e komandave të ndërtuar për këtë qëllim.

- Programi i ndërtuar tregon mënyrën se si mund të kontrollohet pozicioni i robotit kundrejt një objekti të paracaktuar.

- Ekperimentet e kryera tregojnë se algoritmi i ndërtuar mbështetur në idenë e dhënë më lart, funksionon në mënyrë korekte. Probleme shfaqen në rastin e fluktacioneve të dritës në fushën e pamjes së kamerës. Në këto raste humbet pozicioni i objektit në vëzhgim dhe si rezultat programi llogarit vlera të pa sakta për pozicionin. Mënjanimi i vlerave të gabuara në këto raste kryhet nëpërmjet një sistemi krahasimi me vlerat e një hapi përpara. Kur ndryshimet me këto vlera (paraardhëse) janë shumë të thekësuara është e kuptueshme që programi llogaritës ka probleme në gjetjen e objektit në fushën e kamerës.

- Mënjanimi i vlerave të gabuara në raste kur objekti është më i shpejtë se roboti, kryhet nëpërmjet një sistemi krahasimi me vlerat e hapve më përpara. Kur ndryshimet me këto vlera (paraardhëse) vijnë duke u rritur, është e kuptueshme që programi llogaritës ka probleme në gjetjen e objektit në fushën e kamerës ose nuk ka fuqi motorike për ndjekje.

KAPITULLI VII

Hyrje

Rrjetet neurale (NN-të) dhe algoritmat gjenetike (GA-të) janë përdorur nga disa kërkues për të zgjidhur problemin e komandimit të lëvizjeve të robotëve. Në këtë drejtim janë për t’u përmendur punët mjaft të rëndësishme të Mondada & Floreano (1995) [43], Yamada (2005) [44], Capi (2007) [45].

Performanca e një rrjeti neural bazohet në ndërtimin e tij dhe konektimin sinaptik të peshave. Zgjedhja optimale e peshave është një punë jo e lehtë. Algoritmi "back-propagation" është metoda më e mirë për të përmirësuar rrjetin neural, por mund të ketë problemin e minimumit lokal. Metodat si : simulated annealing (SA) (Goffe et al., 1994 [47]), programimi gjenetik (GP) (Koza, 1992 [48]), algoritmat gjenetike (GA) (Goldberg, 1989 [49]) janë përdorur nga disa kërkues për qëllimin e shprehur më sipër. Duhet thënë se GA-të sëbashku me rrjetet neurale kanë shtuar një dimension të ri në fushën e kërkimeve të robotikës, të quajtur robotikë evolucionare (Pratihar, 2003 [46]). Këtu, një strukturë e përshtatshme e rrjetit neural është zhvilluar duke përdorur një GA përmes bashkëveprimeve të përshtatshme me mjedisin. Pratihar (2003) mundësoi një përmbledhje të qartë në aspekte të ndryshme të robotikës evolucionare. Pas zbulimit të avantazheve të lidhjes GA–NN, Hui dhe Pratihar (2004) studiuan performancën e saj për të zgjidhur problemet e navigimit të një roboti si makinë me anë të simulimeve kompjuterike. Ndryshe nga punët më përpara, në eksperimentin e kryer në këtë material, ne marrim parasysh evolucionin e kontrolluesve neuralë për navigimin e robotit në mjedise të caktuara. Në mëtodën që përshkruhet ne vazhdim të këtij kapitulli, ne përpunojmë imazhet ose konvertojmë imazhet e marra në imazhe binare, të cilat më pas përdoren si inpute për kontrolluesin neural.
Pjesa më e madhe e kërkuesve të mësipërm testuan algoritmin e tyre të planifikimit të lëvizjes me anë të simulimeve kompjuterike. Megjithate, rëndësia e zhvillimit të eksperimenteve dukë përdorur robotë të vërtetë për të testuar performancën e planifikimit të lëvizjes është një detyrë që është shtruar nga shumë kërkues. Duhet thënë se simulimi i sensorëve vizualë është shumë i vështirë. Për shkak të kushteve të dritës, kontrolluesit neuralë të zhvilluar në simulim, shpesh kanë një performancë të dobët kur transferohen në robotë të vërtetë. Për të zgjidhur këtë problem, janë zhvilluar kontrolluesit neuralë "online" duke përdorur robotë të vërtetë.

7.1. Teoria e algoritmatve neuronikë (ANN).

Në këtë paragraf do të japim një përshkrim të shkurtër të rrjetave neuronike si dhe algoritmave neuronikë. Qëllimi kryesor është njohja e funksionimit të kontrolluesve neuralë, pra e rrjetave neuronike (NN-të), me qëllimin e vetëm, marrjen e rezultateve sa më të mira në ndjekjen e objekteve të caktuara prej robotit. Informacionet për NN-të sot janë të pafundme, rrjedhojë e një numri shumë të madh kërkuesish në këtë fushë. Materiali në vazhdim mbështetet kryesisht në 'Neural Network Toolbox' të autorëve: Howard Demuth dhe Mark Beale [50].

Ç'janë rrjetat neuronike.

Një rrjet nervor artificial (RNA) është një model i përpunimit të informacionit që është i frymëzuar nga mënyra biologjike se si sistemi nervor dhe sidomos truri, përpunon një informacion. Pika kyçe e këtij modeli është struktura e sistemit të përpunimit të informacionit. Struktura është e formuar nga një numër i madh neuronesh të lidhur me njeri-tjetrin në një mënyrë shumë efektive, të cilët punojnë së bashku për të zgjidhur probleme specifike. Rrjetat neuronike janë grupe neuronesh të organizuara në shtresa. Çdo neuron paraqet një funksion i cili duke vepruar mbi një madhësi në hyrje (input), e cila mund të jetë skalare ose vektoriale jep në dalje (output) një madhësi tjetër gjithashtu skalare ose

vektoriale. Këto output-e varen qoftë nga forma e funksionit transferues, qoftë edhe nga "peshimi" që i jepet vlerave hyrëse, para se mbi to të veprojë funksioni i transferimit.
Disa elemente, me analogët e tyre natyrorë, që do të përdorim në materialin tonë po i paraqesim në vijim :

- Neuronet - analogët e nyjeve nervore.
- Koefiçientët peshë - ekuivalentë të sinapseve nervore.
- Funksionet e transferimit - funksione analogë "imitues" me ata të përpunimit që truri i bën sinjalit nervor.

Një ose disa neurone përbëjnë shtresën neuronike. Disa shtresa neuronike përbëjnë të ashtuquajturën 'Rrjetë Artificiale Neuronike'/ Artificial Neural Networks/ ANN. Neuronet e së njëjtës shtresë nuk lidhen me njëri-tjetrin, ndërkohë që neuronet e shtresave të tjera lidhen midis tyre me anën e peshave neuronike. Sinjali hyrës në një rrjetë neuronike, shpërdahet në të gjithë neuronet e rrjetës njëkohësisht, ashtu si një informacion shpërdahet në nyjet nervore të trurit tonë.
Funksionimi në paralel është tipari kryesor i rrjetave neuronike.
Të gjithë elementët përbërës të rrjetës neuronike do të analizohen më poshtë për të dhënë një ide më të qartë mbi ndërtimin dhe funksionin e tyre.

Neuroni i thjeshtë.

Një neuron i thjeshtë jepet në Fig 7.1. Madhësia në hyrje x e cila përfaqëson një madhësi skalare e cila nëpërmjet një shumëzimi me madhësinë skalare peshë w (weight), jep produktin $w \times x$, përsëri një një madhësi skalare. Produkti $w \times x$ përbën argumentin e funksionit të transferimit f, i cili jep në dalje madhësinë skalare Y. Neuroni i paraqitur në Fig 7.1 është rasti i neuronit më të thjeshtë i cili në një model më të kompletuar jepet në Fig 7.2. Në Fig 7.2 hyrja x e amplifikuar me madhësinë skalare të peshës w, mblidhet nga nyja shumatore me madhësinë skalare $b \times 1$, ku b njihet me emrin devijues (bias). Devijuesi b lidh hyrjen e dytë me koeficient 1 me neuronin. Produkti $w \times x + b$ përbën për këtë rast argumentin e funksionit të transferimit f, i cili jep në dalje madhësinë skalare Y.

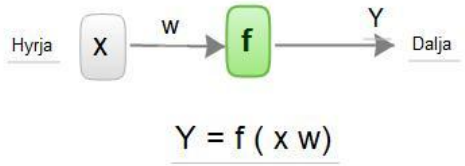


Fig 7.1. *Neuroni i thjeshtë (hyrja x-skalare, dalja Y-skalare)*

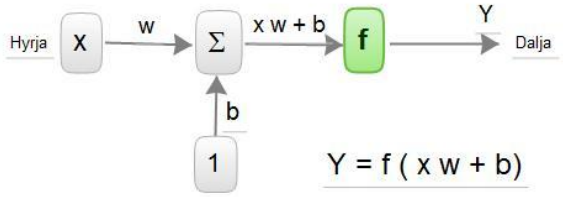


Fig 7.2. *Neuroni i thjeshtë (hyrja x-skalare, devijuesi b, dalja Y-skalare)*

Funksioni i transferimit f paraqet një funksion linear ose jolinear. Funksionet më të përdorshëm janë hardlim, purelin, sigmoid, tansig etj, që transformojnë hyrjen për të dhënë në dalje madhësinë Y. Madhësitë w dhe b janë parametra rregullues të neuronit.
Në Fig 7.3 jepet modeli i një neuroni më të zhvilluar që ofron akoma më shumë me neuronet e trurit.

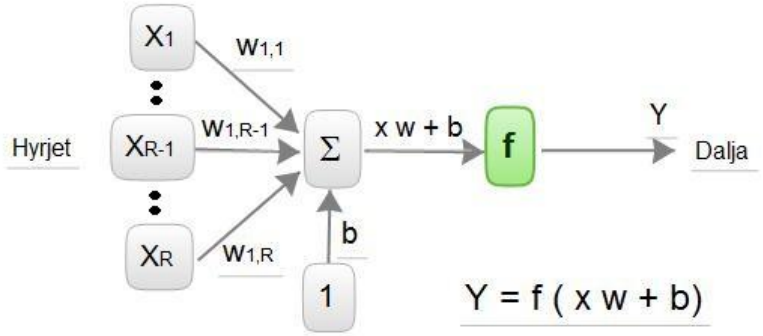


Fig 7.3. *Neuroni (hyrja X-vektoriale, devijuesi b, dalja Y-skalare)*

Në neuron hyrja $X=[x_1 \quad ... \quad x_R]$ përfaqëson një vektor me R elementë, ku secili element në hyrje shumëzohet me peshat përkatëse $w_{1,1}$, .. , $w_{1,R}$ të cilat përbëjnë vektorin e peshave W, dhe produktet shumohen në pikën e takimit. Vektori rresht i peshave $W=[w_{1,1} \quad .. \quad w_{1,R}]$ dhe vektorit kolonë $X=[x_1 \quad ... \quad x_R]'$ shumëzohen dhe më pas mblidhen me madhësinë b, duke përcaktuar madhësinë $W\times X+b$. Kjo e fundit përbën argumentin e funksionit transmetues f i cili në dalje jep madhësinë Y. Duhet thënë se $w_{1,1}$, .. , $w_{1,R}$ njihen me emrin koefiçientët peshë ($\frac{dw}{dt}=0$) ose funksione peshë ($\frac{dw}{dt}\neq 0$) dhe identifikohen si 'sinapset' e rrjetave neuronike. Pra, ashtu si nyjet nervore në trurin e njerut lidhen me anën e sinapseve, tek rrjetat neuronike elementet lidhen me njëri-tjetrin me anën e peshave. Peshat janë vlera fillimisht të vendosura në mënyrë apriori, më tej ato llogariten sipas disa algoritmave të caktuar.

Funksionet e transferimit.

Për sa i takon funksioneve të transferimit mund të themi se lista e plotë e tyre gjendet në mjaft literatura. Në Matlab ai gjendet në librarinë TFG (transfer function graphs).
Funksionet e transferimit më të përdorshme jepen të përshkruara më poshtë :

- Funksioni transferues *hardlim* njihet me emrin funksioni shkallë dhe kufizon vlerat në dalje të neuronit midis 0 dhe 1. Në qoftë se argumenti hyrës në funksionin transferues është ≥ 0 atëhere limiti është 1, në të kundërt kur argumenti hyrës <0, dalja është 0.
- Funksioni i transformimit linear 'Linear Transfer Function'. Neuronet e këtij tipi përdoren si përafrues linearë në filtrat linearë.
$$'\text{purelin}' \rightarrow Linear \rightarrow f(x)=f_0(x)=x$$
- Funksioni i transformimit *sigmoid*, merr në hyrje të gjitha vlerat e intervalit $]-\infty,+\infty[$.

- Për funksionin *logsig* daljet janë në intervalin $]0,1[$.
- Për funksionin *tansig* daljet janë në intervalin $]-1,+1[$.

Funksionet transferues *logsig* dhe *tansig* përdoren shpesh në rrjetat "*feed-forward"* në të cilat përdoret algoritmi i llogaritjes së peshave "*backpropagation*". Meqënëse në këtë algoritëm përdoren derivatet e para të funksioneve transferues, kushti kryesor është që këta funksione të jenë të derivueshëm.

$$\text{'logsig'(Logistic Sigmoid)} \rightarrow f(x) = f_0(x) = \frac{1}{1+e^{-\lambda \cdot x}}$$

$$\text{'tansig'(Hyperbolik Tangent Sigmoid)} \rightarrow f(x) = f_0(x) = \frac{e^{\lambda \cdot x} - e^{-\lambda \cdot x}}{e^{\lambda \cdot x} + e^{-\lambda \cdot x}}$$

Shpesh shënimi i funksionit f zëvendësohet me simbolet e dhëna në Fig 7.4 për të dhënë më qartë rolin e këtyre funksioneve transferues në strukturën e rrjetave neuronike.

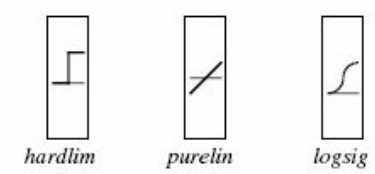


Fig 7.4. *Funksionet transferues hardlim, purelin, logsig*

Rrjetat "*feed-forward*"

Rrjetat "*feed-forward"* janë rrjeta që lejojnë sinjalin të kalojë vetëm në një anë, nga hyrja në dalje. Nuk ka kthime mbrapa. Rrjetat "*feed-forward"* kanë tendence të jenë të drejtperdrejta në lidhjen e hyrjeve me daljet. Ato përdoren gjerësisht në trajtimet e problemeve të ndryshme.

Rrjetat "*feedback"*

Rrjetat "*feedback"* mund të kenë sinjale që qarkullojnë në të dy drejtimet. Rrjetat "*feedback"* janë shumë të fuqishme dhe gjithashtu shumë të komplikuara. Ato janë dinamike, gjendja e tyre ndryshon vazhdimisht derisa arrijnë në një gjendje ekuilibri. Qëndrojnë në gjendje ekuilibri derisa të dhënat hyrëse ndryshojnë, e me tej kalohet në një ekuilibër të ri. Arkitekturat "*feedback"* gjithashtu njihen si interactive ose të ripërsëritura, megjithëse termi i fundit shpesh përdoret për të përcaktuar lidhjet "*feedback*" në organizime njështresore.

Ndërtimi i modeleve dhe arkitektura e rrjetës neuronike.

Dy ose më shumë neurone mund të kombinohen në një strukturë që quhet shtresë. Një rrjetë neuronike mund të përmbajë një ose më shumë shtresa të tilla. Në fillim le të trajtojmë një shtresë neuronike me R elementë në hyrje dhe me S neurone në shtresë, si në Fig 7.5.

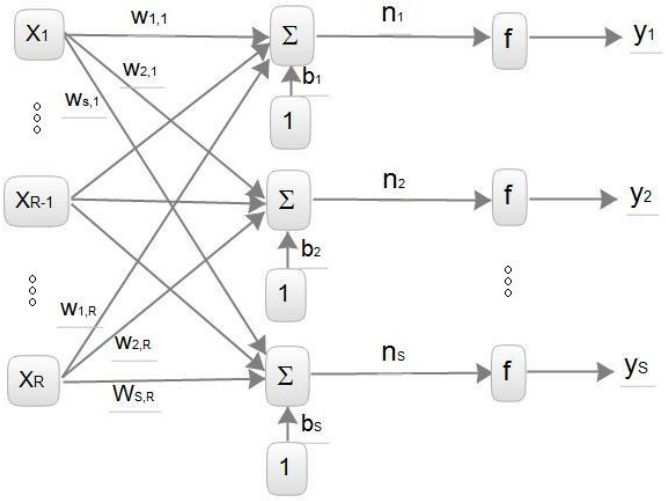


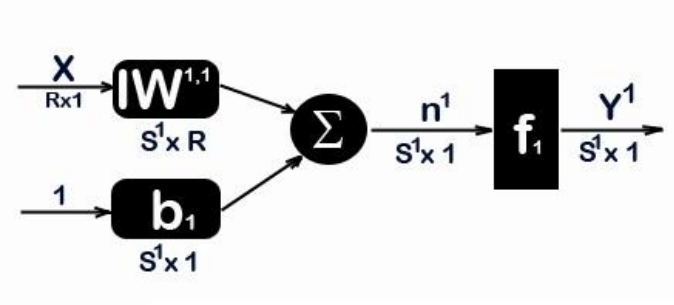
Fig 7.5. *Rrjeta me një shtresë neuronike.*

Në këtë rrjetë çdo element i vektorit X në hyrje, është lidhur me çdo neuron nëpërmjet matricës peshë W. Elementët e matricës peshë W jepen të organizuar si më poshtë :

$$W = \begin{vmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{2,1} & w_{2,2} & \dots & w_{2,R} \\ \dots & \dots & \dots & \dots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{vmatrix}$$

Indekset e rreshtit në elementet e matricës W tregojnë destinacionin neuronik të peshës, pra indeksi rjesht tregon neuronin me të cilin do të lidhet pesha. Indekset e kolonës tregojnë se kush janë burimet hyrëse të këtyre peshave. Kështu, madhësia $w_{m,n}$ përcakton peshën që lidh meuronin e m-të me hyrjen e n-të. Vektori X është me përmasa $R \times 1$, vektori ndikues b me përmasa $S \times 1$ dhe matrica W me përmasa $S \times R$. Çdo shtresë ka një matricë peshë W, një vektor ndikues b, dhe vektorin që merret në dalje Y. Për të bërë dallimin midis matricave peshë W, vektorëve ndikues b, vektorëve dalës Y dhe funksioneve të transferimit f të shtresave të ndryshme, do t'u shtojmë këtyre vektorëve dhe matricave nga një indeks në pjesën lart i cili do të tregojë numrin e shtresës së cilës i përket variabli që na intereson. Kjo bëhet pasi duhet të bëjmë një dallim midis matricave peshë të cilat janë të lidhura në hyrje dhe matricave peshë që janë të lidhura midis shtresave, gjithashtu duhet të identifikojmë burimin dhe destinacionin për një matricë peshë. Do të emërtojmë matrica peshë të lidhura në hyrje si pesha hyrëse dhe do të emërtojmë pesha të shtresës matricat

peshë që dalin nga shtresat. Do të përdorim indekset lart për të identifikuar burimin (indeksi i dytë) dhe destinacionin (indeksi i parë) për peshat e ndryshme dhe elementë të tjerë të rrjetës.



$$Y^1 = f^1(IW^{1,1} \times X + b^1)$$

Fig 7.6. *Rrjeta me një shtresë neuronike e grupuar.*

Në Fig 7.6 përshkruhet një shtresë neuronike dhe pikërisht shtresa e parë së bashku me elementet e saj si :

- Matrica peshë e lidhur me vektorin X si matrica peshë hyrëse (Input Weight matrix $IW^{1,1}$, matrica me peshat fillestare që lidhen me vlerat hyrëse) që ka si burim 1 dhe si destinacion 1.
- Vektorët b, n dhe X kanë indeksin 1 lart për të treguar se të gjithë këta elementë i përkasin shtresës së parë.

Shënimi që përdoret në këtë rast në Matlab për një peshë të dhënë është :

$$IW^{1,1} \rightarrow net.IW\{1,1\}$$

Llogaritja e madhësisë pas shumatores do të kryhej nga kodi i dhënë në vazhdim :

$$n\{1\} = net.IW\{1,1\} * X + net.b\{1,1\};$$

Në Fig 7.7 jepen rrjeta neuronike tre shtresore me të gjithë elementët e saj. Mund të bëjmë një ndarje në disa pjesë të rrjetës si vijon :

- Hyrjet
- Shtresa 1 ose shtresa hyrëse në rrjetë.
- Shtresa 2 ose shtresa e fshehur.
- Shtresa 3 ose shtresa dalëse në rrjetë.
- Daljet

Vihet re se shtresat janë të lidhura midis tyre në një formë të tillë që daljet e njërës shtresë janë hyrje për shtresën tjetër. Shtresat kanë numër të ndryshëm neuronesh, kështu p.sh. shtresa hyrëse ka S^1 neurone, shtresa e dytë ka S^2 neurone dhe e treta S^3. Mardhëniet midis madhësive në dalje kundrejt atyre në hyrje përshkruhen nga ekuacionet e mëposhtme :

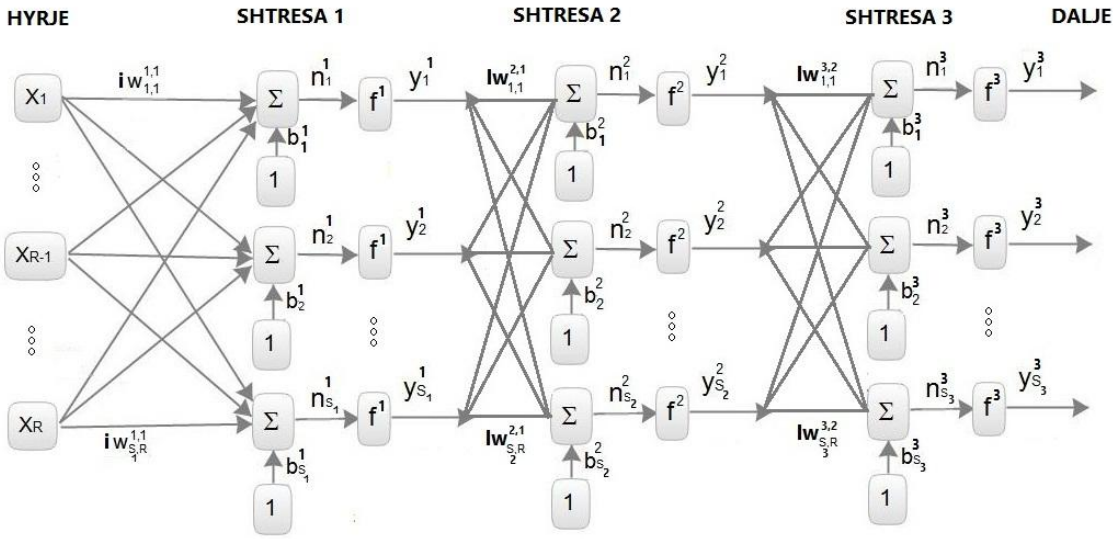
Shtresa 1 : $Y^1 = f^1(IW^{1,1} \times X + b^1)$

Shtresa 2 : $Y^2 = f^2(LW^{2,1} \times Y^1 + b^2)$

Shtresa 3 : $Y^3 = f^3(LW^{3,2} \times Y^2 + b^3)$

Mardhënien midis daljes dhe hyrjes, duke u mbështetur në ekuacionet e mesipërme, e shprehim në formën e dhënë më poshtë :

$$Y^3 = f^3(LW^{3,2} \times (f^2(LW^{2,1} \times (f^1(IW^{1,1} \times X + b^1)) + b^2)) + b^3)$$

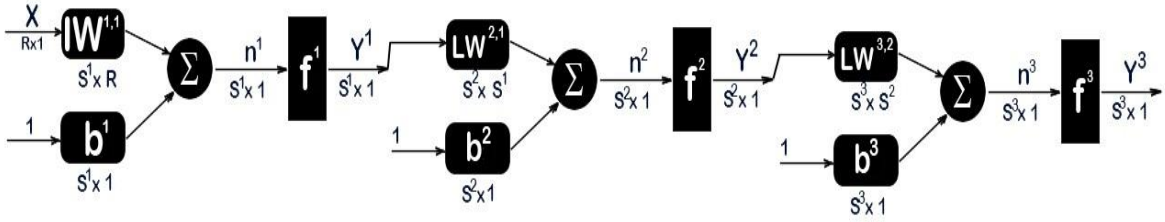


$$Y^1 = f^1(IW^{1,1} \times X + b^1);\ \ Y^2 = f^2(LW^{2,1} \times Y^1 + b^2);\ \ Y^3 = f^3(LW^{3,2} \times Y^2 + b^3)$$

$$Y^3 = f^3(LW^{3,2} \times (f^2(LW^{2,1} \times (f^1(IW^{1,1} \times X + b^1)) + b^2)) + b^3)$$

Fig 7.7. *Rrjeta neuronike tre shtresore.*

Në Fig 7.8 jepet rrjeta neuronike tre shtresore e grupuar duke përshkruar madhesitë vektoriale dhe matricore me përmasat e tyre.



$$Y^1 = f^1(IW^{1,1} \times X + b^1);\ \ Y^2 = f^2(LW^{2,1} \times Y^1 + b^2);\ \ Y^3 = f^3(LW^{3,2} \times Y^2 + b^3)$$

$$Y^3 = f^3(LW^{3,2} \times (f^2(LW^{2,1} \times (f^1(IW^{1,1} \times X + b^1)) + b^2)) + b^3)$$

Fig 7.8. *Rrjeta neuronike tre shtresore e grupuar.*

Si përfundim mund të themi se rrjetat janë kombinim i neuroneve pra i peshave, vektorëve zhvendosës, funksioneve të transferimit etj. Por e veçanta në ndërtimin e rrjetës qëndron në funksionin që ajo do të kryejë. Nisur nga ky funksion, rrjetat klasifikohen në rrjetat e perceptimit,

rrjetat e parashikimit etj. Këto të fundit janë përdorur në studimin tonë dhe do të trajtohen gjërësisht.

Rrjeti neural në zgjidhje të një detyre të caktuar.

Për të bërë një rrjet neural që performon një detyrë të caktuar, duhet të zgjedhim sesi janë të lidhura njësitë me njëra-tjetrën dhënë në Fig 7.7, Fig 7.8 dhe duhet të vendosim në mënyrën e duhur peshat në këto lidhje. Lidhjet vendosin nëse është e mundur që një njësi të ndikojë mbi një tjetër. Peshat përcaktojnë fuqinë e ndikimit.
Ne mund të "mësojmë" një rrjet neural prej tre shtresash të bëjë një detyrë të caktuar duke përdorur procedurën e mëposhtme :

- I "*prezantojmë*" rrjetit shembuj trajnimi, të cilët konsistojnë në formën e aktivitetit të njësive hyrëse (inputs) se bashku me formën e dëshiruar të aktivitetit të njësive dalëse (outputs).
- Përcaktojmë se sa afër output-it aktual të rrjetit janë output-et e dëshiruara.
- Ndryshojmë peshat e lidhjeve në mënyrë që rrjeti të prodhojë në mënyrë më të përafërt output-in e kërkuar.

Për të trajnuar rrjetin, ne japim një gjendje hyrjeje dhe e krahasojmë aktivitetin aktual të daljeve me aktivitetin e dëshiruar. Më pas llogarisim gabimin, që shprehet si katrori i ndryshimit midis aktivitetit actual dhe atij të dëshiruar. Më pas ne ndryshojmë peshat e çdo lidhjeje në mënyrë që të zvogëlojmë gabimin. E përsërisim këtë proçes për hyrje të ndryshme deri kur rrjeti të arrijë që të klasifikoje çdo dalje në mënyrën e duhur.
Për ta implementuar këtë procedure ne duhet të llogarisim ndryshimin e gabimit për peshën (EW) në mënyrë që të ndryshojmë peshën në një masë e cila do të jetë në përpjestim të drejtë me shkallën me të cilin ndryshon gabimi, ndërkohë që ndryshon dhe pesha. Një mënyrë për të llogaritur EW-në është duke influencuar nga pak në vlerat e peshave duke vëzhguar se si ndryshon gabimi. Por kjo mënyrë është e pamjaftueshme sepse kërkon një influencim të veçantë në secilën peshë.
Një mënyrë tjetër për të llogaritur EW-në është duke përdorur algoritmin "*back-propagation*" i cili është përshkruar më poshtë dhe në ditët e sotme është një nga mënyrat më të mira për të trajnuar rrjetet neurale. Ky algoritëm është zhvilluar në mënyrë të pavarur nga dy skuadra, njëra : Fogelman-Soulie, Gallinari dhe Le Cun në Francë dhe tjetra : Rumelhart, Hinton dhe Williams në SH.B.A.

Algoritmi "*Back-propagation*"

Në mënyrë që të trajnohet nje rrjet neural për të performuar një detyrë te caktuar, duhet që të rregullohen peshat e secilës njësi në mënyrë që gabimi midis output-it të dëshiruar dhe atij aktual të zvogëlohet. Kjo diferencë shënohet shkurtimisht me EA. Për këtë duhet që rrjetat neurale të llogarisin ndryshimin e gabimeve të peshave (EW-ve) d.m.th të llogaritin sesi ndryshon gabimi (EA-ja) me rritjen ose uljen e secilës peshë. Algoritmi "*back-propagation*" është metoda më e përdorur për të përcaktuar EW-në.
Algoritmi "*back-propagation*" është më i lehtë për t'u kuptuar sikur të gjitha njësitë e rrjetit të ishin lineare. Algoritmi llogarit çdo EW-vë duke llogaritur si fillim EA-në, ritmin me të cilin ndryshon gabimi ndërsa aktiviteti i një njësie ndryshon. Për njësitë dalëse, EA-ja është thjesht diferenca midis output-it aktual dhe atij të dëshiruar. Për të llogaritur EA-në për një njësi të fshehur në shtresën para shtresës së output-it, ne si fillim identifikojmë të gjitha peshat midis asaj njësie të fshehur dhe njësive dalëse me të cilat është e lidhur. Më pas i shumëzojmë ato pesha me EA-të e atyre njësive dalëse dhe mbledhim prodhimet. Kjo shumë është e barabartë me EA-në e njësisë së fshehtë. Pasi

kemi llogaritur të gjitha EA-të në shtresën e fshehtë përpara shtresës së output-it, ne mund të llogarisim EA-të për shtresat e tjera, duke lëvizur nga shtresa në shtresë në drejtim të kundërt me shpërhapjen e aktivitetit të rrjetit. Prej këtej e merr emrin dhe ky algoritëm. Në çastin që kemi llogaritur EA-në për një njësi, mund të llogarisim EW-të për çdo lidhje të ardhshme të njësisë. EW-ja është prodhimi i EA-së dhe aktivitetit të lidhjeve pasuese.
Duhet vene re që për njësi jo lineare, algoritmi "*back-propagation*" ka nevojë për hapa shtesë [54].

7.2. Teoria e algoritmatve gjenetikë (GA).

Algoritmi Gjenetik (GA) është një algoritëm optimizimi me disa përdorime të rëndësishme në fusha të ndryshme të teknikës. Algoritmi gjenetik bazohet parimisht në teorinë klasike të evolucionit të Darwinit, në teorinë e seleksionimit natyral të Weismannit dhe konceptin e gjenetikës së Mendelit, të cilat së bashku njihen si teoria Neo-Darwiniane [51].
Neo-Darwinismi bazohet në proceset e riprodhimit, mutacionit dhe selektimit. Aftësia për të riprodhuar është një cilësi thelbësore e jetës. Aftësia për të ndryshuar (mutacioni) garantohet në çdo organizëm të gjallë që riprodhon veten në një mjedis që ndryshon vazhdimisht. Proçeset e selektimit zenë vend në botën natyrale ku popullata që zgjerohen të llojeve të ndryshme kufizohen në një hapsirë të fundme.
Në vitet 1970 John Holland prezantoi konceptin e algoritmave gjenetike (GA). Qëllimi i tij që të realizonte nëpërmjet kompjuterit procese përzgjedhjesh dhe evolucioni, të njëjta si ato që realizohen prej natyrës. Algoritmat Gjenetike të dhënë nga Holland mund të paraqiten nëpërmjet një procedure për kalimin nga një popullatë e "kromozomeve" artificiale tek një popullatë e re. Ky algoritëm përdor selektimin "natyror" dhe teknika të përdorura nga gjenetika të njohura si crossover (kryqëzimi) dhe mutacion (ndyshimet). Çdo kromozon konsiston në një numër "genesh" dhe cdo gen paraqitet me 0 ose 1. Meqënese GA-të përdorin një metodë kërkimi stokastik, përshtatshmëria e një popullate mund të mbetet e pandryshueshme për një numër të gjeneratave përpara se të shfaqet një kromozom superior. Kjo në disa raste e bën problematik plotësimin e një kriteri përfundimtar. Një praktikë e zakonshme është të përfundosh algoritmin gjenetik pas një numri të përcaktuar të gjeneratave dhe pastaj të zgjedhësh kromozomet më të mira në popullatën e fundit. Nëse një zgjidhje e kënaqshme nuk gjendet, algoritmin gjenetik rifillon me një popullatë te re. Algoritmi gjentik siguron mbajtjen konstante të numrit të individëve të popullatës, dhe në të njëjtën kohë siguron përmirësimin e përshtatshmërisë së saj në vlerë mesatare. Gjatë ekzekutimit të algoritmit gjenetik, kryhen dy procese të rëndësishme, përcaktuse ne rezultatin perfundimtar, që njihen si crossover dhe mutacion. Operatori crossover zgjedh rastësisht një pozicion te të dy kromozonet 'prinder'dhe më tej shkëmben informacioni pas këtij pozicioni. Si rezultat krijohen dy pasardhës të rinj. Vlerësimi i kryqëzimeve jepet zakonisht nga vlera propabilitare 0.7 që konsiderohet një vlerë optimale për ecurinë normale të algoritmit. Nëse një çift kromozomesh nuk kryqëzohen atëhere shfaqen kromozomet e klonuara dhe pasardhësit krijohen kopje të sakta të cdo 'prindi'. Mutacioni i cili ndodh rrallë në natyrë, paraqet një ndryshim në gene, i cili mund të sigurojë përmirësime domethënëse në përshtatshmërinë e individeve, por edhe mund të sjellë rezultate të dëmshme. Mutacioni si një element i rëndësishëm siguron që algoritmi i kërkimit të mos mbetet në një optimum lokal. Hapat e zgjedhjeve të popullatave dhe operacionet crossover mund të ndalin në ndonjë bashkësi homogjene të zgjidhjeve. Nën kushte të tilla të gjitha kromozomet janë identike dhe kështu përshtatshmëria mesatare e popullatës nuk mund të përmirësohet. Pra algoritmi i kërkimit nuk është i aftë të proçedojë më tej. Mutacioni është i njëvlershëm me një kërkim të rastit dhe na ndihmon në shmangien e shkatërrimit të ndryshimeve gjenetike. Mutacionet gjenetike në 'kromozomet' e popullatës janë të ralla dhe kjo shprehet me një propabilitet në kufijtë nga 0.001

deri në 0.01. Algoritmi gjenetik do të përfundojë llogaritjet atëhere kur do të jetë plotësuar një prej dy kritereve të përcaktuara. Kriteri i parë është plotësimi i kushtit i funksionit qëllimor dhe i dyti plotësimi i numrit maksimal të interacioneve të përcaktuara.
Përfundimisht mund të themi se GA-ja vepron nëpërmjet një cikli të thjeshtë të përbërë nga 4 faza siç tregohet në Fig 7.9. Çdo cikël prodhon një brez të ri të zgjidhjeve të mundshme për një problem. Në fazën e parë, një popullsi fillestare me zgjidhje potenciale është krijuar si pikë fillimi për kërkimin. Në fazën pasardhëse, përformanca e çdo individi vlerësohet në bazë të kufizimeve të vendosura nga problemi. Bazuar në performancën e secilit individ, një mekanizëm selektiv zgjedh prindërit për operatorët e kryqëzimit dhe mutacionet. Operatori i kryqëzimit merr dy kromozome dhe shkëmben pjesë të informacionit të tyre gjenetik në mënyrë që të prodhojë kromozome të reja. Operatori i mutacionit krijon struktura të reja në popullsi duke modifikuar në mënyrë rastësore disa nga gjenet, duke ndihmuar në këtë mënyrë algoritmin kërkimor që të arrijë të shmanget nga pengesat e minimumit local. Pasardhësit e prodhuar nga proceset gjenetike të manipulimit janë popullsia e ardhshme e cila do vlerësohet. GA-ja mund të zëvendësojë ose të gjithë popullsinë ose vetëm pjesëtarët më pak të përfshirë. Cikli krijim-vlerësim-zgjedhje-manipulim vazhdon deri kur një zgjidhje e kënaqëshme gjendet ose kur disa kërkesa të tjera takohen. Për çdo teknikë revolucionare duhet një përfaqësim kromozomi për të përshkruar çdo individ në popullatë [52] [53].

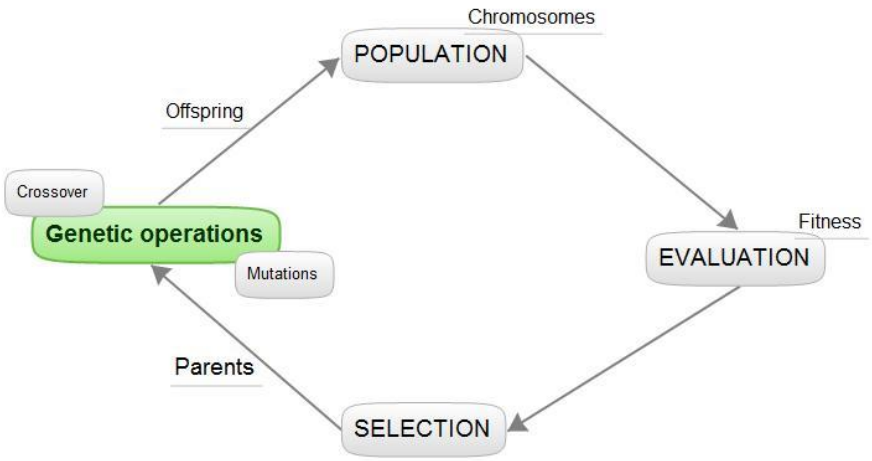


Fig 7.9. *Operacionet në GA*

Bllokskema dhe algoritmi për përcaktimin e koefiçientëve të modelit.

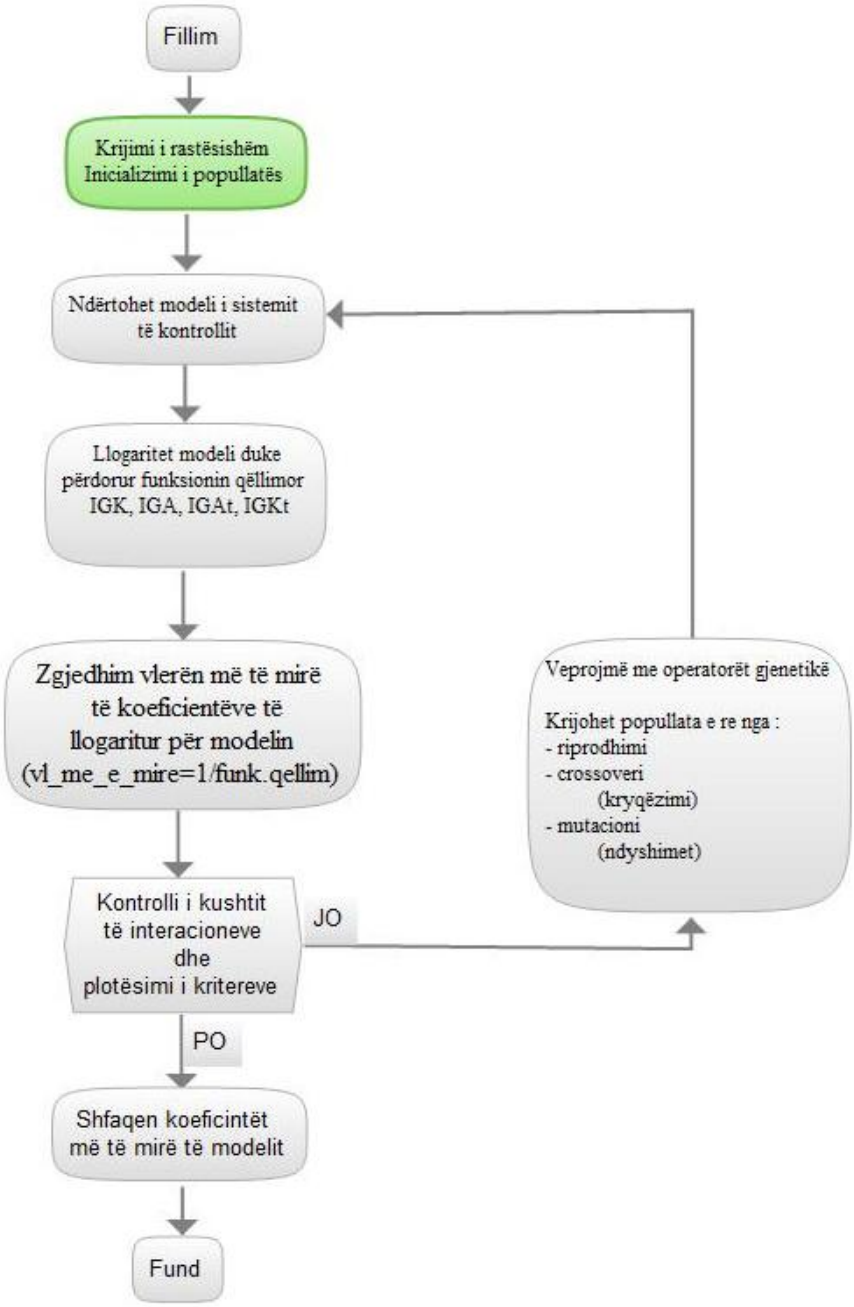


Fig 7.10. *Përcaktimi i koeficientëve të modelit me GA*

Funksioni qëllimor për përcaktimin e koeficienteve të modelit mbështetur në algoritmin gjenetik .

Funksionet qëllimore që do të përdoren për përcaktimin e koeficientëve të modelit, nuk do të jenë veçse kritert integrale dhe mesatare të gabimit në përgjigjen në dalje të modelit, të cilat janë jo pak të njohura.

Kriteret integerale dhe mesatare të gabimit shprehen në këto forma:

- Integrali i gabimit kuadratik (IGK) e cila në literaturën e huaj njihet me emrin ISE.

$$IGK = \int_0^\infty \varepsilon^2(t)dt$$

- Integrali i gabimit absolut (IGA) e cila në literaturën e huaj njihet me emrin IAE.

$$IGA = \int_0^\infty |\varepsilon(t)|\, dt$$

- Integrali i gabimit absolut sipas kohës (IGAt) e cila në literaturën e huaj njihet me emrin ITAE.

$$IGAt = \int_0^\infty t \cdot |\varepsilon(t)|\, dt$$

- Integrali i gabimit kuadratik sipas kohës (IGKt) e cila në literaturën e huaj njihet me emrin ITSE.

$$IGKt = \int_0^\infty t \cdot \varepsilon^2(t)dt$$

- Mesatarja e katrorit të gabimit (MKG) e cila në literaturën e huaj njihet me emrin MSE.

$$MSE = \frac{1}{n} \cdot \sum_{t=1}^{n} \varepsilon^2(t)$$

ku $\varepsilon(t) = r(t) - y(t)$

Përdorimi i kësaj teknike do të thotë përcaktimi i koefiçientëve të modelit, në mënyrë të tillë që të minimizojnë IGK , IGA , $IGAt$, $IGKt$ ose MSE .

7.3. Përdorimi i algoritmave neuronikë (ANN) në realizimin e orientimit dhe kontrollit të distancës së iRobot Create prej një objekti në lëvizje.

Përdorim i algoritmave neuronikë në kontrollin e distancës së një roboti prej një objekti në lëvizje lidhet me llogaritjen nëpërmjet tyre të trajektores dhe shpejtësisë të lëvizjes së robotit. Për të realizuar këtë detyre rrjetat neuronike që mund të përdoren janë të ndryshme, në funksion të modelit robotik në përdorim dhe rastit konkret të problemit që do të shtrohet për zgjidhje. Do të trajtojmë dy raste të përdorimit të rrjetave neuronike (NN), që lidhen me problemin në shqyrtim, për llogaritjen e shpejtësive të rrotave të robotit në ndjekjen e një objekti në lëvizje dhe orientimit të lëvizjes së robotit drejt një zonë të paracaktuar.

Llogaritja e shpejtësive të rrotave të robotit në ndjekjen e një objekti në lëvizje me ndihmën e rrjetave neuronike (NN).

Modeli i robotit në shqyrtim jepet në Fig 7.11 për të cilin është folur në kapitujt e mësipërm. Ai përbëhet nga dy rrota të lidhura me dy motorë të pavarur në lëvizjen e tyre. Për të realizuar lëvizjen e robotit, sipas një trajektoreje të caktuar dhe me shpejtësi lëvizjeje të ndryshueshme, mjafton që të kontrollojmë madhësitë v_{majtas} dhe $v_{djathtas}$. Rasti më i përgjithshëm do të qe ai i paraqitur nëpërmjet pamjeve të simulimeve të dhëna në Fig 7.12.

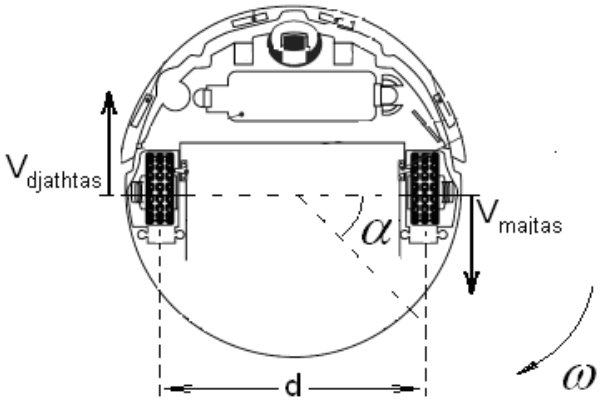


Fig 7.11. *Modeli i robotit.*

Në Fig 7.12 jepen dy objekte të cilët kanë kryer trajektore të pavarura. Objekti me ngjyrë blu përfaqëson objektin lëvizës. Objekti me ngjyrë të kuqe (të plotë), përfaqëson robotin i cili ka për detyrë të qëndrojë në një distancë të caktuar nga objekti lëvizës. Sipas Fig 7.12 është e kuptueshme që pozicionet e qëndrimit të robotit për të ruajtur distancën nga objekti në lëvizje janë të pafundme. Ato teorikisht janë në të gjitha pikat e rrethit me rreze sa distanca e kërkuar me qendër objektin lëvizës. Praktikisht për modelin në shqyrtim pozicionet e qëndrimit të robotit janë në harkun që kalon nëpër qendrën e rrathëve të kuq të pa mbushur, brenda zonës së pamjes së kameras (zona me blu të çelët). Rrathët e kuq të vegjel përfaqësojnë disa pozicionime të robotit kundrejt objektit lëvizës. Përfundimisht mund të themi se pozicioni i robotit kundrejt objektit do të përcaktohet nga dy parametra : distanca e kërkuar dhe nga këndi kundrejt objektit lëvizës. Në simulimet dhe realizimet ekperimentale do të kryejmë kontrollin e distancës për tre mundësi pozicionimi të robotit kundrejt objektit :

- në të majtë të objektit.
- në qendër të objektit.
- në të djathtë të objektit.

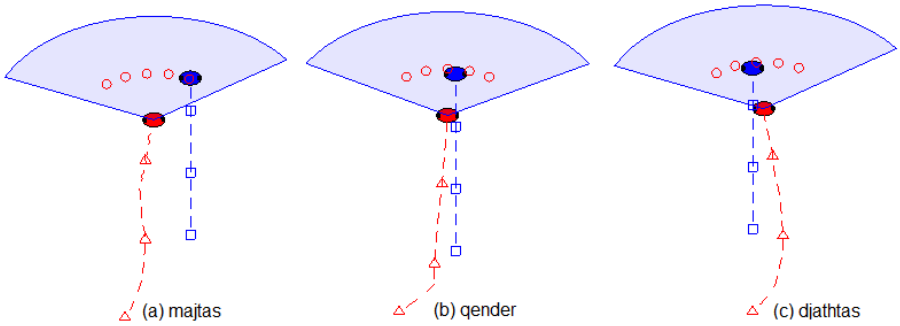


Fig 7.12. *Simulimi i ndjekjes prej robotit i një objekti në lëvizje.*

Rrjeta neuronike (NN) që do të përdorim në këtë rast do të jetë një rrjete model NN (input 3/ hidden 3 / output 2). Ajo do të ketë në hyrje (inputs) tre madhësi : distancën (distance), këndin (angle) dhe drejtimin (direction) të konvertuara në njësi relative në përputhje me kërkesat e NN për shtresën hyrëse të përbërë nga tre neurone, një shtresë të fshehur me tre neurone dhe në dalje një shtresë me dy neurone, pra dy dalje që janë madhësitë v_{majtas} dhe $v_{djathtas}$ për rotat e robotit. Vlerat në dalje konvertohen në $\frac{cm}{sek}$ përfundimisht. Rrjeta neuronike (NN) e përdorur jepet në Fig 7.13.

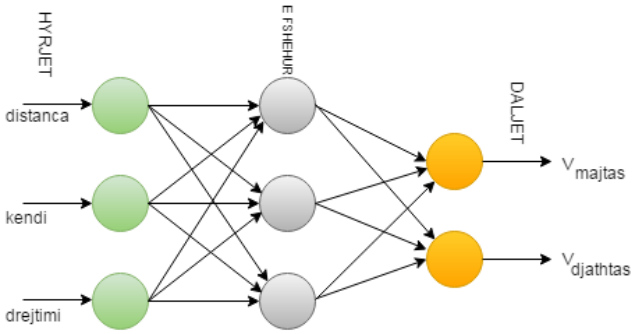


Fig 7.13. *Rrjeta neuronike e përdorur për kontrollin e shpejtësisë së rrotave, në kontrollin e distancës.*

Programi që realizon në Matlab 2013b llogaritjen e madhësive në dalje prej atyre të hyrjes dhe peshave të rrjetës $W\_i\_h(15)$ jepet në vijim :

```
% Funksioni i ndertuar per llogaritjen e NN.
% nn = struct('W_i_h',[], 'nr_i',3, 'nr_h',3, 'nr_o',2);
% W_i_h(1:1:15) - peshat
% nr_i - numri i neuroneve ne shtresen e hyrjes (distanca, kendi, drejtimi)
% nr_h - numri i neuroneve ne shtresen e fshehur
% nr_o - numri i neuroneve ne shtresen e daljes (Vmajtas, Vdjathtas)

%function [aVelL, aVelR, n_act]=NN_A(nn, MaxSpeed)
function [aVelL, aVelR]=NN_A(nn, MaxSpeed)

for i=1:nn.nr_h
   Out_H(i)=0;
   for j=1:nn.nr_i
       Out_H(i)=Out_H(i)+nn.Input_V(j)*nn.W_i_h(i,j);
   end
```

```
    Out_H(i)=logsig(Out_H(i));
end

for i=1:nn.nr_o
  Out_Sum(i)=0;

  for j=1:nn.nr_h
    Out_Sum(i)=Out_Sum(i)+Out_H(j)*nn.W_h_o(i,j);
  end
  if i==1
     aVelR=MaxSpeed*logsig(Out_Sum(i));
  else
     aVelL=MaxSpeed*logsig(Out_Sum(i));
  end
end
%n_act=[nn.Input_V Out_H logsig(Out_Sum(1)) logsig(Out_Sum(2))];
```

Bllok-skema bazë që realizon ndjekjen e një objekti në lëvizje me ndihmën e rrjetave neuronike (NN) jepet në Fig 7.14.

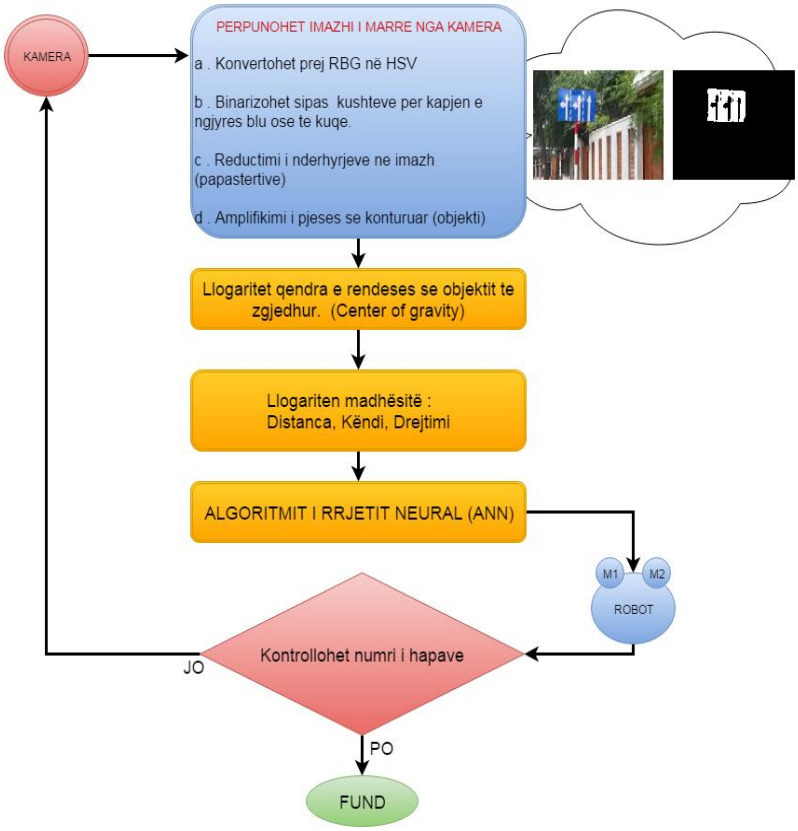


Fig 7.14. *Ndjekja e një objekti në lëvizje me ndihmën e rrjetave neuronike (NN)*

Që një sistem i tillë i përshkruar të funksionojë duke plotësuar kërkesat e shtruara është absolutisht i domosdoshëm zgjedhja e saktë e peshave të duhura për rrjetin neural (NN) të zgjedhur. Gjetja e vlerave të sakta të tyre është një detyrë jo e lehtë, të cila për rastin konkret llogariten sipas algoritmit gjenetik (GA).

Programin e ndërtuar në Matlab2013b do e japim në Shtojcën 13.

Orientimi i lëvizjes së robotit drejt një zone të paracaktuar me ndihmën e rrjetave neuronike (NN).

Një rast i rëndësishëm që lidhet me orjentimin e lëvizjes së robotit drejt një zone destinacion sipas rrugës më të shkurtër, përshkruhet nëpërmjet Fig 7.18. Eshtë e kuptueshme dhe pa asnjë diskutim se të gjithë do të ndërtonin një drejtëz që lidh robotin me zonën destinacion për të treguar rrugën më të shkurtër.

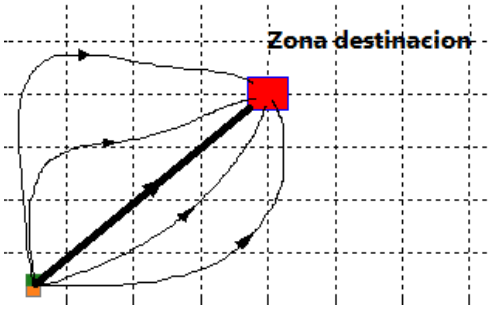


Fig 7.15. *Orjentimet e lëvizjes drejt një zone destinacion.*

Problemi do të ndërlikohej nqs në këtë rrugë do të shfaqeshin pengesa për robotin. Në këtë rast do të kërkohej që ato të anash kaloheshin. Kjo do të siguronte që roboti të arrinte në destinacion sipas rrugës më të shkurtër. Kjo pjesë do të përbënte për ne një sfide që do të qe punë e së ardhmes. Në pjesën në vazhdim do të tregojmë se si përdoren rrjetat neural (NN) në orientimin e lëvizjes së robotit drej zonës destinacion, kur mungojnë pengesat në rrugën e robotit.
Rrjeta neuronike (NN) që do të përdorim në këtë rast do të jetë një rrjete model NN (input 2/ hidden 4 / output 4). Ajo do të ketë në hyrje (inputs) dy madhësi : rob-x dhe rob-y për shtresën hyrëse të përbërë nga dy neurone, një shtresë të fshehur me katër neurone dhe në dalje një shtresë me katër neurone, pra katër drejtimet e lëvizjes që janë madhësitë $h_{lëviz\_majtas}$, $h_{lëviz\_djathtas}$, $h_{lëviz\_lart}$ dhe $h_{lëviz\_poshtë}$. Rrjeta neuronike (NN) e përdorur jepet në Fig 7.16.

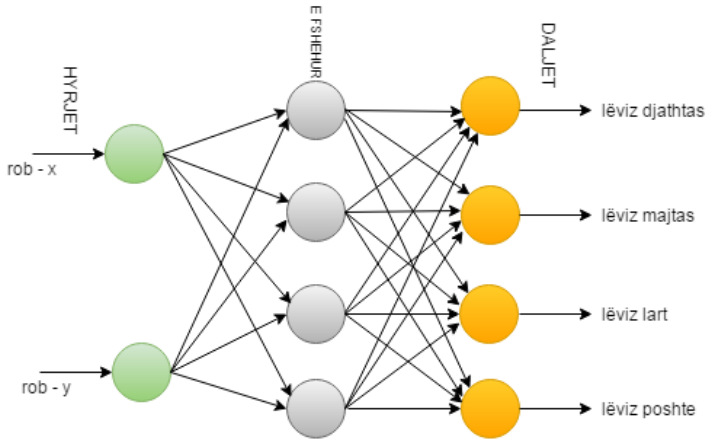


Fig 7.16. *Rrjeta neuronike e përdorur për kontrollin e distancës.*

Programi që realizon në Matlab 2013b llogaritjen e madhësive në dalje prej atyre të hyrjes dhe peshave të rrjetës $vv(1,24)$ jepet në vijim :

```
function [action]=NN_A(Input_V,vv,nr_i,nr_h,nr_o)
% Input_V=[1 2];
% nr_i=2;
% nr_h=4;
% nr_o=4;
% vv=1:1:24;
for i=0:nr_h-1
    W_i_h(i+1,:)=vv(i*nr_i+1:i*nr_i+nr_i);
end
for i=0:nr_o-1
    W_h_o(i+1,:)=vv(nr_h*nr_i+i*nr_h+1:nr_h*nr_i+i*nr_h+nr_h);
end
for i=1:nr_h
        sum_h = 0.0;
        for j = 1:nr_i
            sum_h = sum_h+W_i_h(i,j) * Input_V(j);
        end
        out_h(i) = 1.0 / (1.0 + exp(-sum_h));
end

sum_eQ=0;
for i = 1:nr_o
    sum_o = 0;
    for j=1:nr_h
        sum_o = sum_o + W_h_o(i,j) *out_h(j);
    end
    p(i) = 1.0 / (1.0 + exp(-sum_o));
    sum_eQ=sum_eQ+p(i);
   end

%for i=1:nr_o
%   prob_a(i)=p(i)/sum_eQ;
%end

[max_prob index]=max(p);

if (index==1)
      action(1)=1; action(2)=0; action(3)=0; action(4)=0;
elseif (index==2)
      action(1)=0; action(2)=1; action(3)=0; action(4)=0;
elseif (index==3)
      action(1)=0; action(2)=0; action(3)=1; action(4)=0;
elseif (index==4)
      action(1)=0; action(2)=0; action(3)=0; action(4)=1;
end
```

Kërkesa e shtruar zgjidhet plotësisht vetëm pasi të bëhet zgjedhja e saktë e peshave për rrjetin neural (NN) të zgjedhur. Gjetja e vlerave të tyre do të kryhet sipas algoritmit gjenetik (GA).

Programin e ndërtuar në Matlab2013b do e japim në Shtojcën 14.

7.4. Përdorimi i algoritmave gjenetikë (GA) në realizimin e orientimit dhe kontrollit të distancës së iRobot Create prej një objekti në lëvizje.

Përdorimi i algortimit gjenetik (GA), si algoritëm optimizimi, në rastet e dhëna më sipër ka të bëjë me gjetjen e vlerave të peshave për rrjetat neurale (NN).

Do të trajtoja dy raste për gjetjen dhe optimizimin e vlerave të peshave për rrjetat neurale (NN) të optimizuara me algoritmin gjenetik (GA) :

- Optimizimi i peshave sipas një parametri të sistemit.
- Optimizimi i peshave sipas disa parametrave në sistem.

Me optimizim të peshave sipas një parametri do të nënkuptoja gjetjen e peshave për rrjetin neural (NN) për të cilat në sistem plotësohet një kusht i caktuar, si psh përcaktimi i koeficientëve të modelit, në mënyrë të tillë që të minimizojnë *IGK* , *IGA* , *IGAt*, *IGKt* ose *MSE* .
Për rastet në shqyrtim do të shfrytëzonim programet e GA-ve për optimizimin sipas një kriteri të njohura si One-objective GA optimization.

Me optimizim të peshave sipas disa parametrave do të nënkuptoja gjetjen e peshave për rrjetin neural (NN) për të cilat në sistem plotësohen disa kushte të caktuar njëkohësisht. Algoritmi gjenetik (GA) në këto raste njihet me emrin Multi-objective GA optimization.
Për rastin e trajtuar në orientimin e lëvizjes së robotit drejt një zone të paracaktuar me ndihmën e rrjetave neuronike (NN), gjetja dhe optimizimi i peshave realizohet kur pas ekzekutimit të One-objective GA dhe përfundimit të tij, funksioni Fitness() na jep listën e vlerave për `nr_actions`. Eshtë e kuptueshme që prej kësaj liste do të zgjidhet ajo ku parametri `nr_actions` ka vlerën më të vogël, që nënkupton dhe rrugën më të shkurtër për në zonën destinacion. Kësaj vlere i përkon një vektor për vlerat e `vv()`, që është vektori i peshave i optimizuar për rrjetën neurale (NN).

Më poshtë po japim funksionin qëllim që realizon gjetjen e peshave optimale për numrin minimal të hapave që duhet të bëjë roboti për të kryer rrugën më të shkurtër.

```
function nr_actions =Fitnes(vv)
   nr_i=2; % input
   nr_h=4; % hidden
   nr_o=4; % outpute

   % Zona destinacion për robotin
   obj_l=3; % Gjeresia e zones destinacion
   obj_w=3; % Lartësia e zones destinacion
   obj_x=20; % Koordinata x e zones destinacion (cepi poshte majtas ne xy)
   obj_y=20; % Koordinata y e zones destinacion (cepi poshte majtas ne xy)

   agent_l=1; % Gjerësia e Robotit
   agent_w=1; % Lartësia e Robotit
   agent_x=2.5; % Koordinata x e robotit (koord x fillim)
   agent_y=1.5; % Koordinata y e robotit (koord y fillim)

   nr_actions=0;
   while (nr_actions<100)
      if (agent_x>=obj_x-obj_l/2 & _
          agent_x<=obj_x+obj_l/2 & _
          agent_y>=obj_y-obj_w/2 & _
          agent_y<=obj_y+obj_w/2)
```

```
        nr_actions
        break;
    end

    Input_V=[agent_x agent_y];

    action=nn_A(Input_V,vv,nr_i,nr_h,nr_o); % - algoritmi NN

    if action(1)==1
       agent_x = agent_x+1;  % - leviz djathtas
    elseif action(2)==1
       agent_x = agent_x-1;  % - leviz majtas
    elseif action(3)==1
       agent_y = agent_y+1;  % - leviz lart
    elseif action(4)==1
       agent_y = agent_y-1;  % - leviz poshte
    end
    nr_actions=nr_actions+1;
end
```

Në mënyrë të ngjashme më poshtë po japim disa rreshta të funksionit qëllim që realizon gjetjen e peshave optimale për kontrollin e distancës prej një objekti në lëvizje.

Madhësia `R1_sa_dist = abs(gool(1).dist-dist(2))` jep gabimin absolut midis vlerës kërkuar të distancës dhe asaj të vërtetë për çdo një hap. Vlera më e vogël nga shumatoret e të gjitha vlerave të `R1_sa_dist` për çdo llogaritje (një llogaritje përmbledh një simulim që ka 450 hapa), do të jetë zgjidhja më optimale nga GA për vlerat e peshave.

Një pjese e funksionit qëllim për llogaritjen e peshave të optimizuara me One-Objective GA jepet ne Shtojcën 13.

Si përfundim mund të themi se :

- Nëpërmjet këtij kapitulli u tregua se si mund të realizohet teorikisht kontrolli i distancës së robotit prej një objekti në lëvizje duke shfrytëzuar rrjetat neurale (NN) të optimizuara me algoritmat gjenetike (GA).
- Programet e ndërtuara në Matlab2013 tregojnë mënyrën se si mund të kontrollohet pozicioni i robotit kundrejt një objekti të paracaktuar.

KAPITULLI VIII

Në këtë kapitull do të jepen disa rezultate të arritura në simulime dhe në kohe reale për rastet e kontrollit të distancës prej një objekti lëvizës dhe të orientimit drejt një zone destinacion.

8.1. Simulimi i kontrollit të distancës së iRobot Create prej një objekti në lëvizje, në Matlab R2013b.

Në pjesën në vijim po paraqes strukturat gjëndje dhe qëllim në programim Matlab2013, për robotët 1-lider dhe 2-ndjekës. Në të gjitha simulimet distanca e kërkuar për t'u arritur është 5 njësi.

```
% struct e robotit
rob(1)=struct('x',300,'y',300,'th',1.5*pi,'r',17,'trajectory',[],'himg',[],'color','b');
rob(2)=struct('x',300,'y',100,'th',pi/2,'r',17,'trajectory',[],'himg',[],'color','r');

% Destinacioni perfundimtar
gool(1)=struct('dist',5,'ang',5,'himg',[]);
gool(2)=struct('dist',5,'ang',11,'himg',[]);
```

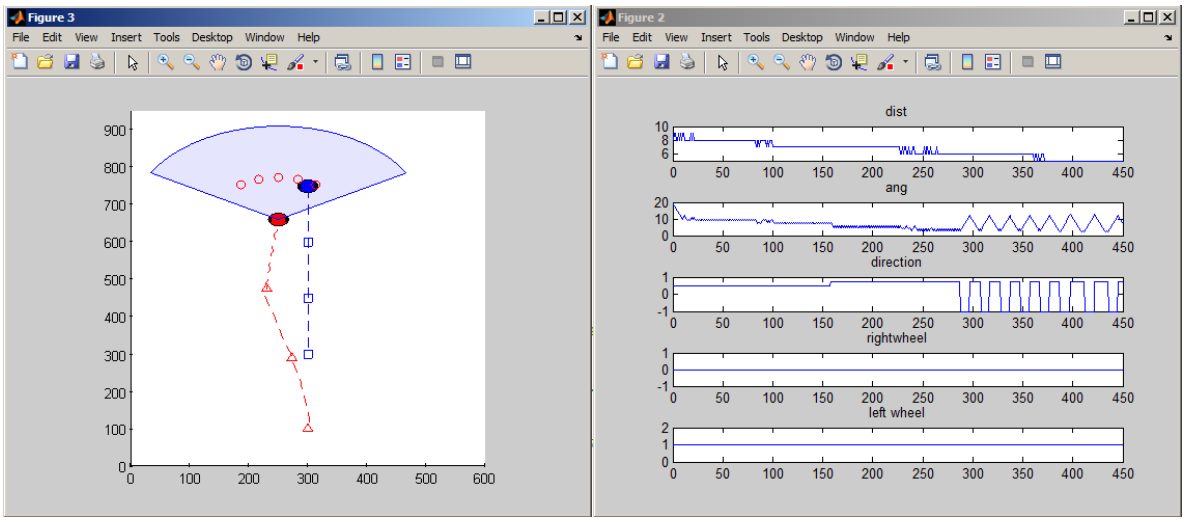


Fig 8.1. *Simulimi 1: Roboti në të majtë të objektit në ndjekje.*

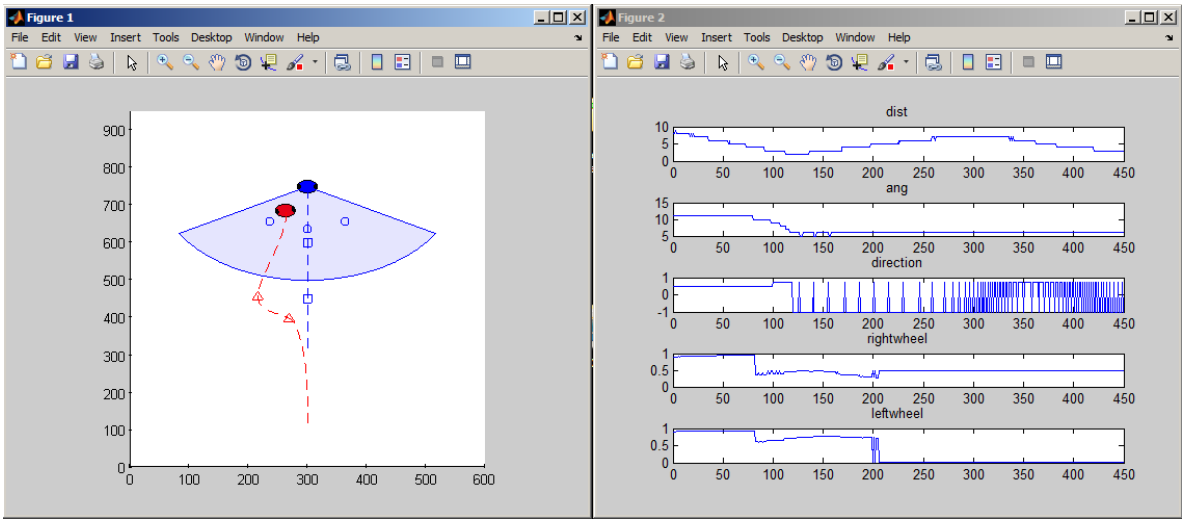


Fig 8.2. *Simulimi 2: Roboti në të majtë të objektit në ndjekje.*

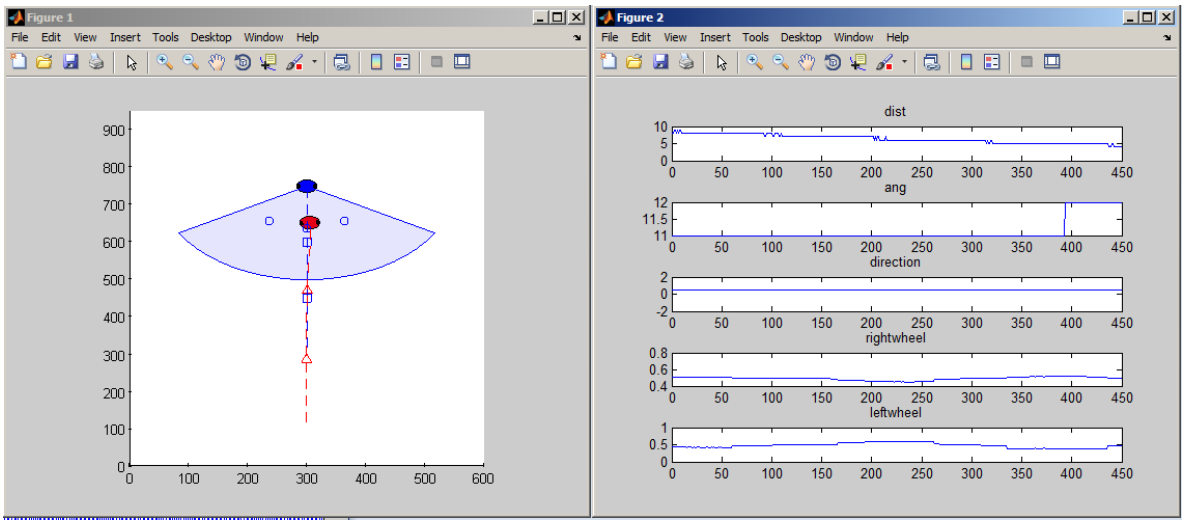


Fig 8.3. *Simulimi 3: Roboti në qendër të objektit në ndjekje.*

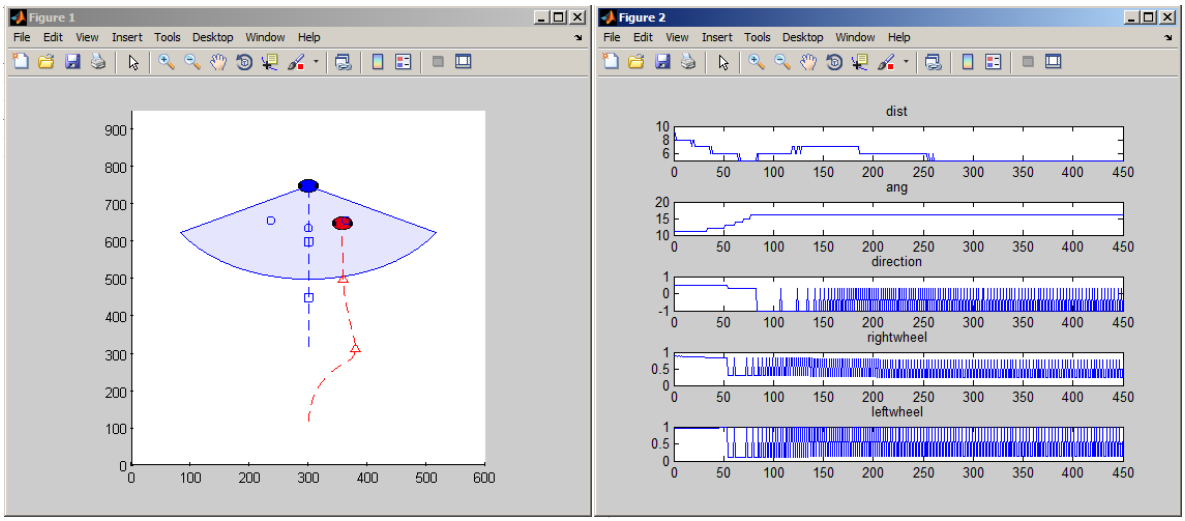


Fig 8.4. *Simulimi 4: Roboti në të djathtë të objektit në ndjekje.*

8.2. Realizimi ekperimental i kontrollit të distancës së iRobot Create prej një objekti qëllim në Matlab R2013b.

Në mënyrë eksperimentale është kryer kontrolli i distancës së iRobot Create (4400) prej një objekti në lëvizje. Objekti në lëvizje është një tjetër robot i targuar me kapak të kuq në pjesën e sipërme. Lëvizja e tij është e komanduar automatikisht prej një kompjuteri (laptop) të vendosur mbi robot, si në Fig. 8.6. Ndjekja e robotit lider (objektit lëvizës) do të kryhet duke ruajtur një distancë prej tij 1200 mm. Në Fig 8.5 jepet grafikisht për një testim në real-time grafiku i distancës së robotit ndjekës prej robotit lider, në funksion të kohës.

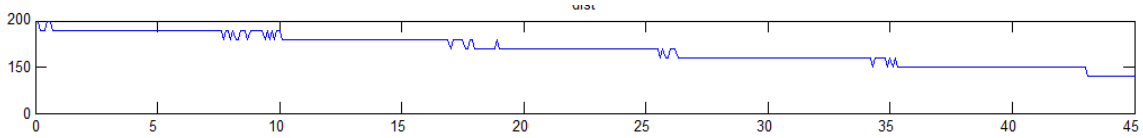


Fig 8.5. *Foto shpjeguese: Grafiku i distancës së robotit ndjekës prej robotit lider.*

Fig 8.6. *Foto shpjeguese: Roboti në ndjekje të objektit në lëvizje.*

Fig 8.7. *Real Time 1: Roboti në ndjekje të objektit në lëvizje.*

Fig 8.8. *Real Time 2: Roboti në ndjekje të objektit në lëvizje.*

8.3. Programet e simulimeve dhe ato në kohë reale Matlab R2013b.

Moduli kryesor i programit në simulim dhe kohë reale, për kontrollin e distancës së robotit prej një objekti lëvizës (roboti 1) .

```
% ------------------------------------------------------------------------
SelectCase=1;  % Zgjedhje për testim 1 - majtas, 2 - qender, 3 - djathtas
% ------------------------------------------------------------------------

if SelectCase==1
    % majtas
     vv=[   -3.9735   -2.1148   -1.0604    3.6589   -4.7129   -3.9536    0.1256    4.9233...
             3.0959    4.6086   -3.2578    2.9948   -4.5460   -1.1166    2.7116];
end

if SelectCase==1
   % qender
   vv=[   -3.6834   -4.7931    4.3168   -4.5963   -4.4180   -4.8110   -3.4138    3.2661...
           -4.8199   -1.1358   -4.5268    0.4614    2.3255    1.9262   -3.5656];
end

if SelectCase==1
```

```
   % djathtas
   vv=[    0.1522   -0.6279   -2.6109    3.8763   -1.4645    4.1903   -2.8077    3.9736...
           0.5842    0.1565    3.2560   -1.9560   -4.4319    2.4831    2.8787];
end

plot_flag=1;

% NN : numri i neuroneve
nr_i=3;
nr_h=3;
nr_o=2;

in1_di=[];
in1_a=[];
in1_d=[];

out1_r=[];
out1_l=[];


% struct e robotit
rob(1)=struct('x',300,'y',300,'th',1.5*pi,'r',17,'trajectory',[],'himg',[],'color','b');
rob(2)=struct('x',300,'y',100,'th',pi/2,'r',17,'trajectory',[],'himg',[],'color','r');


% Destinacioni perfundimtar
gool(1)=struct('dist',5,'ang',5,'himg',[]);
gool(2)=struct('dist',5,'ang',11,'himg',[]);

num_cr=size(rob,2)-1;
field=[0,600,0,800];        % Fusha e pamjes [ xmin, xmax, ymin, ymax ]


% Pershkrime grafike
if plot_flag==1
   hfig=InitFig(field);            % Inicializimi i figures
   hcam=[];                        % Handle of the visual range of the camera image
   rob(1)=DrawRob(hfig,rob(1));    % roboti
   rob(2)=DrawRob(hfig,rob(2));    % qellimi
end

max_step=450;   % Numri maksimal i hapave
dt=0.05;        % koha nga njeri hap tek tjetri ne sek (s)

t_p_x1=[];
t_p_y1=[];
t_p_x2=[];
t_p_y2=[];

step=0;
while(step<max_step)

    step=step+1;

    if step==0
        plot(rob(1).x,rob(1).y,'s','MarkerSize',6,'MarkerEdgeColor','b');
        plot(rob(2).x,rob(2).y,'^','MarkerSize',6,'MarkerEdgeColor','r');
    end

    [visible,ang,dist]=CameraSens(rob);     % (Gjendja : 1 jashte pamjes visible, 0 brenda
zones se pamjes
    direction=RobDirection(rob);            %  Orientimi i robotit (.3 .5 .7 -1)

    in1_di=[in1_di,dist(1)];
    in1_a=[in1_a,ang(1)];
    in1_d=[in1_d,direction(1)];
```

```
    for i=1:num_cr
        if visible(i)==0
            dist(i)=1;
            ang(i)=0;
            direction(i)=-1;
        end

        Input=[dist(i)/10,ang(i)/21,direction(i)];                    % Hyrjet NN
        Output=NN_A(Input,vv,nr_i,nr_h,nr_o);                         % Daljet NN
        rob(i+1)=MoveiRob_t(rob(i+1),Output(1)*50,Output(2)*50,dt);    % Levizja e robotëve
    end

    out1_r=[out1_r,Output(1)];
    out1_l=[out1_l,Output(2)];

    % Levizja e roboti qe perfaqeson objektin levizës
    rob(1) = MoveiRob_r_t(rob(1),-20,-20,dt);

    % Pershkrimi grafik
    if plot_flag==1
       hcam = DrawCamera(hfig,hcam,rob);      % Kamera vision
       gool = DrawGool(hfig,rob,gool);        % Qëllimi

       rob(1) = DrawRob(hfig,rob(1));         % robotet
       rob(2) = DrawRob(hfig,rob(2));

       plot(t_p_x1,t_p_y1,'s','MarkerSize',6,'MarkerEdgeColor','b');
       plot(t_p_x2,t_p_y2,'^','MarkerSize',6,'MarkerEdgeColor','r');

       drawnow
    end

    if mod(step,150)==0
       t_p_x1=[t_p_x1, rob(1).x];
       t_p_y1=[t_p_y1, rob(1).y];
       t_p_x2=[t_p_x2, rob(2).x];
       t_p_y2=[t_p_y2, rob(2).y];
    end
end

figure(2)

subplot(5,1,1)
plot(1:450,in1_di(1:450))
title('dist')
subplot(5,1,2)
plot(1:450,in1_a(1:450))
title('ang')
subplot(5,1,3)
plot(1:450,in1_d(1:450))
title('direction')
subplot(5,1,4)
plot(1:450,out1_r(1:450))
title('rightwheel')
subplot(5,1,5)
plot(1:450,out1_l(1:450))
title('leftwheel')
```

Si përfundim mund të themi se :

- Nëpërmjet këtij kapitulli u treguan rezultatet e kontrollit të distancës së robotit prej një objekti në lëvizje me ndihmën e kameras duke shfrytëzuar rrjetat neurale (NN) të optimizuara me algoritmat gjenetike (GA).

- Programet e ndërtuara në Matlab2013 tregojnë mënyrën se si mund të kontrollohet pozicioni i robotit kundrejt një objekti të paracaktuar.

- Simulimet dhe ekperimentet e kryera tregojnë se algoritmi i ndërtuar funksionon në mënyrë korekte.

9. KONKLUZIONE

Në materialin që paraqitëm :

- Realizuam komandimin e motorëve me hapa në disa mënyra prej kompjuterit, në funksion të portës I/O ku lidhet indirekt paisja motorike.
  - Realizuam komandimin e lëvizjes dhe shpejtësisë së motorit me hapa duke shfrytëzuar portat I/O të kompjuterit, të emërtuara LPT, COM, USB.
  - Realizuam komandimin e lëvizjes dhe shpejtësisë së motorit me hapa prej kompjuterit, pa kabëll (wireless) të motorit me hapa.

  Për të gjitha rastet e trajtuara më sipër u tregua komandimi i shpejtësisë së motorit prej frekuencës së dhënies së komandave dhe konkretisht u realizua :
  - Rrotullim me frekuenca të ndryshme me hap të plotë
  - Rrotullim me frekuenca të ndryshme me gjysëm hapi.
  - Ndalim të menjëhershëm të cdo gjendjeje të tij.
  - Rrotullime në kahun orar ose në atë jo orar.

- Treguam disa mënyra se si kryhet komandimi i motorit me hapa prej kompjuterit, si një prej elementëve kryesorë në paisjet e komanduara dhe robotike, nëpërmjet portave I/O. Realizuam komandimin e motorit me hapa prej portave LPT, COM, USB dhe komandimin valor në distancë me ndihmen e programeve të ndërtuara në disa mjedise programimi.

- Ndërtuam një sistem të thjeshtë ndjekës objektesh nëpërmjet përpunimit të imazheve të sjella nga një kamer. Sistemi ndjekës i përshkruar dhe realizuar mund të shërbejë për ndjekjen e çfarëdolloj objekti të ndritshëm, pra ai përbën një sistem ndjekës universal. Sistemi ndjekës shpreh një mënyrë se si duke ndjekur lëvizjen e diellit, paneli diellor me fotoelementë pozicionohet sipas drejtimit ku sipërfaqja aktive për fluksin e dritës merr vlerën maksimale. Treguam funksionet që duhen përdorur për komandimin e motorit me hapa prej portës USB nëpërmjet :
  - PIC18F4550-I/P & ULN2803
  - Modulit Stepper-Bee.

- Trajtuam dhe ekperimentuam tre metoda për vlerësimin e distancës së një objekti të caktuar prej një sistemi vëzhgimi, të përmbledhura më poshtë :

  - Realizuam një sistem të përbërë prej kompjuterit, një kamere (Monocular Vision) dhe një gjeneruesi lazeri pikësor. Imazhi i marë nga kamerat përpunohen duke përcaktuar nëpërmjet një algoritmi të thjeshtë pozicionin e rezes së dritës (lazer) mbi objektin që

vëzhgojmë. Nëpërmjet këtij pozicionimi përcaktuam distancën e objektit prej sistemit të kontrollit.

- Realizuam një sistem të përbërë prej kompjuterit dhe dy kamerave (Stereo Vision). Llogaritjet për përcaktimin e distancës u mbështetën në parimet e funksionimit të sistemit të kamerave si dhe në përdorimin e koncepteve gjeometrike. Objekti të cilit ju përcaktua distanca është një trup i targuar d.m.th një trup me një shenjë të dallueshme nga mjedisi përreth ose mbi të cilin godet një reze drite lazer duke siguruar shënjimin e objektit.

- Realizuam një sistem të përbërë prej kompjuterit dhe një kamere (Monocular Vision) ku nëpërmjet përpunimit të imazhit dhe analizave gjeometrike duke shfrytëzuar principet e prespektivës, llogaritëm distancen e një objekti prej planit të kamerës. Kjo metodë vlerësimi të distancës së objekteve prej një kamere njihet si metoda e vlerësimit të thellësisë (Depth Estimation).

- Realizuam komandimin e robotit prej portës seriale (lidhja me kabëll ose valor) duke shfrytëzuar librarinë e komandave të ndërtuar për këtë qëllim në mjediset Microsoft Visual Basic 6.0 dhe Microsoft Visual Basic .NET, Android 2.0 dhe Matlab2013. Nëpërmjet programeve të ndërtuara në mjediset e mësipërme tregohet se si mund të thirren komandat për vënien e robotit në lëvizje drejtvizore, rrotulluese në mënyrë të kontrolluar.

- Realizuam kontrollin e distancës së robotit prej një objekti në lëvizje me ndihmën e kameras duke shfrytëzuar metoda klasike të ndjekjes përcaktuar nga shpejtësia e lëvizjes së objektit. Ekperimentet e kryera tregojnë se programi i ndërtuar në Microsoft Visual Studio (Visual Basic) 6.0 funksionon në mënyrë korekte. Probleme shfaqen në rastin e fluktacioneve të dritës në fushën e pamjes së kamerës. Në këto raste humbet pozicioni i objektit në vëzhgim dhe si rezultat programi llogarit vlera të pa sakta për pozicionin. Mënjanimi i vlerave të gabuara në këto raste kryhet nëpërmjet një sistemi krahasimi me vlerat e një hapi përpara. Kur ndryshimet me këto vlera (paraardhëse) janë shumë të thekësuara është e kuptueshme që programi llogaritës ka probleme në gjetjen e objektit në fushën e kamerës. Mënjanimi i vlerave të gabuara në raste kur objekti është më i shpejtë se roboti, kryhet nëpërmjet një sistemi krahasimi me vlerat e hapve më përpara. Kur ndryshimet me këto vlera (paraardhëse) vijnë duke u rritur, është e kuptueshme që programi llogaritës ka probleme në gjetjen e objektit në fushën e kamerës ose nuk ka fuqi motorike për ndjekje.

- Realizuam kontrollin e distancës së robotit prej një objekti në lëvizje me ndihmën e kameras duke shfrytëzuar rrjetat neurale (NN) të optimizuara me algoritmat gjenetike (GA). Programet e ndërtuara në Matlab2013 tregojnë mënyrën se si mund të kontrollohet pozicioni i robotit kundrejt një objekti të paracaktuar. Simulimet dhe ekperimentet e kryera tregojnë se algoritmat e ndërtuar funksionojnë në mënyrë korekte dhe ndjekja e objektit realizohet me lëvizje robotike shumë më afër atyre humane krahasuar me lëvizjet e robotit në rastin e kontrollit të distancës me metodën klasike. Mënjanimi i vlerave të gabuara në raste kur objekti është më i shpejtë se roboti, kryhet nëpërmjet një sistemi krahasimi me vlerat e hapve më përpara. Për të mënjanuar këto raste, eksperimenti i ndjekjes së objektit në lëvizje u krye me dy robotë identikë Rommba 4400, ku njëri luan rolin e objektit në lëvizje që ndiqet prej robotit të dytë.

9.1 Shfrytëzuesit dhe rëndësia e rezultateve

Teknikat e komandimit dhe rezultatet modeste të arritura janë të domosdoshme për të realizuar projekte të tjera që lidhen me komandimin dhe drejtimin e sistemeve të thjeshta robotike. Ajo që është dhënë në këtë material është një punë paraprake që mund t'i shërbejë programuesve si një ndihmesë për zgjidhje të problemeve të komandimit të paisjeve nëpërmjet portave I/O të kompjuterave. Për të realizuar logjikën komanduese, pra pas realizimit të programit që merr informacion nga sensorët (input sensors) dhe bën përpunimin e hyrjeve duke na dhënë derivat komandat që duhen të jepen në motor, ajo që mbetet është ekzekutimi i këtyre komandave ose thënë ndryshe 'dërgimi' i tyre në motor. Pikërisht në këtë moment të fundit hyn në shfrytëzim ajo ç'ka është paraqitur në këtë material modest.
Kontrolli i distancës së robotit prej një objekti në lëvizje, me ndihmën e kameras duke shfrytëzuar metodën klasike të ndjekjes, përcaktuar nga shpejtësia e lëvizjes së objektit dhe metodën moderne të rrjetave neurale (NN) të optimizuara sipas algoritmit gjenetik (GA), mendoj se do të përbëjnë një hap përpara në rrugën e modernizimit të kontrolleve në sistemet robotike. Programet e ndërtuara në mjedise të ndryshme programimi mund të jenë një ndihmesë për studiues të kësaj fushe. Gjithashtu, simulimet dhe ekperimentet e kryera me sqarime në detaje në hapat e kryer do të jenë përsëri një ndihmesë, sado e vogël.

Rëndësia e punimeve të kryera për mendimin tonë qëndron edhe në faktin se pas realizimeve të arritura deri në këtë fazë, fushëpamja për kryerjen e detyrave të tjera bëhet më e pastër dhe idetë janë më të qarta. Problemet e ndjekjes së objekteve janë të ndërlidhura dhe me probleme si orientimi autonom i robotëve në mjedise të panjohura, që njihen si problemet e navigimit të robotëve në mjedise të ndryshme.

9.2 Puna në të ardhmen

Punën për të ardhmen do ta përmblidhja në disa drejtime të cilat po i japim në vazhdim :

- Komandimi i motorit me hapa në LAN dhe në WEB. Trajtimi i këtij problemi mund të jetë shumë i rëndësishëm pasi me të mund të jenë të lidhura një numër i madh aplikimesh në industri dhe mjeksi.
- Optimizimi i kontrollit të distancës dhe pozicionit së robotit prej një objekti në lëvizje, me ndihmën e kameras duke shfrytëzuar metodën klasike të ndjekjes, përcaktuar nga shpejtësia e lëvizjes së objektit.
- Optimizimi i kontrollit të distancës dhe pozicionit së robotit prej një objekti në lëvizje, me ndihmën e kameras duke shfrytëzuar metodën moderne të rrjetave neurale (NN) të optimizuara sipas algoritmit gjenetik (GA) jo sipas një objektivi por sipas optimizimit multi-objectiv.

10. LITERATURA

[1] Principles of Digital Designer; Prentice-Hall; 1997 - Taub, Herbert.
[2] Circuitos Digitais e Microprocessadores; McGraw-Hill; 1984.
[3] Stepping motors: a guide to modern theory and practice Acarnley, P. P. P. Peregrinus on behalf of the IEE, 1984, c1982. LC number: TK2537 .A28 1984
[4] Programing Microsoft Visual Basic 6.0
[5] Programing Microsoft Visual Studio (Visual Basic) 2013
[6] Programing Microsoft Visual Studio (C#) 2013
[7] http://en.wikipedia.org/wiki/Stepper_motor, Aug. 2012.
[8] http://www.rmv.com
[9] Hawkins Electrical Guide, Theo. Audel and Co., 2nd ed. 1917, vol. 1, ch. 21: Brushes and the Brush Page 300-327
[10] http://homepage.cs.uiowa.edu/~jones/step/, Sep. 2012.
[11] http://www.DatasheetCatalog.com, Nov. 2012.
[12] http://www.engineersgarage.com/electronic-components/uln2003-datasheet, Nov. 2012.
[13] http://www.4tronix.co.uk/arduino/Stepper-Motors.php, Oct. 2012.
[14] Matlab R2013b
[15] http://www.kiatronics.com/28byj-48-stepper-motor-5vdc-code-70289.html
[16] http://www.elecrow.com/wiki/index.php?title=ULN2003_Stepper_Motor_Driver
[17] The Art of Assembly Language Programming. Chapter 12. Spring 2008, Yale University
[18] RF Transmitter for 433 MHz SLWS113D−NOVEMBER 2000 − REVISED FEBRUARY 2005
[19] 8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System. Programmable Flash ATmega640/V, ATmega1280/V, ATmega1281/V, ATmega2560/V, ATmega2561/V. 2549P/AVR/10/2012.
[20] Stepper Bee. Stepper motor control from a PC's USB port. Installation and Users Manual. (Including AutoStep Software)
Available exclusively from. PC Control Ltd. www.pc-control.co.uk. 2009 Copyright PC Control Ltd.
[21] A vision-based approach for intelligent robot navigation. G.Capi. Int. J. Intelligent Systems Technologies and Applications, Vol. X, No. X, xxxx
[22] Parallax Laser Range Finder (#28044). v1.0 9/16/2011 Page 24 of 24.
[23] A simple distance measurement system that uses a laser and a webcam.
http://www.ugcs.caltech.edu/~dstaff/video.tar
[24] Obstacle detector using OpticalFlow sensor and laser pointer.
http://www.sparkfun.com/products/10061
[25] Webcam Based DIY Laser Rangefinder.
http://www.codeproject.com/Articles/91470/Computer-Vision-Laser-Range-Finder
[26] Optical methods for distance and displacement measurements. Garry Berkovic and Ehud Shafir. Received April 3, 2012; revised July 16, 2012; accepted July 24, 2012; published September 11, 2012 (Doc. ID 166051)
[27] Kim, J.B.; Jun, H.S. Vision-based location positioning using augmented reality for indoor navigation. IEEE Trans. Consum. Electr. 2008, 54, 954962.
[28] Hills, A. Wireless andrew. IEEE Spectr. 1999, 36, 49-53.
[29] Want, R.; Hopper, A.; Falcao, V.; Gibbons, J. The active badge location system. ACM Trans. Inform. Syst. 1992, 10, 91-102.

[30] Ni, L.M.; Lau, Y.C.; Patil, A.P. LANDMARC: Indoor location sensing using active RFID. Wirel. Netw. 2004, 10, 701-710.
[31] Subramanian, V.; Burks, T.F.; Arroyo, A.A. Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation. Comput. Electr. Agric. 2006, 53, 130-143.
[32] Barawid, O.C., Jr.; Mizushima, A.; Ishii, K. Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application. Biosyst. Eng. 2007, 96, 139-149.
[33] Thrun, S.; Burgard, W.; Fox, D. Probabilistic Robotics; MIT Press: Cambridge, UK, 2005.
[34] Gonzales, Rafael C. Woods, Richard E.Eddins, Steven L 2008. Digital Image Processing using Matlab. Second Edition. United States of America. Gatesmark Pu-blishing.
[35] RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments by Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, Dieter Fox
[36] ActiveComport Serial Port Toolkit Manual Pages 1999-2005 - ActiveXperts Software
[37] Basic4Android Anywhere Software tutorial 2007 – 2013
[38] iRobot Create Open Interface (OI) Specification and Virtual Wall. ©2006 iRobot Corporation.
[39] iRobot Command Module Owner's Manual. ©2007 iRobot Corporation.
[40] Basic4Android Anywhere Software tutorial 2007 – 2013
[41] Microsoft serial port class documentation: http://msdn2.microsoft.com/library/30swa673(en-us,vs.80).aspx
[42] Esposito, Joel M., Barton, Owen, Koehler, Joshua, Lim, David, "Matlab Toolbox for the Create Robot", www.usna.edu/Users/weapsys/esposito/roomba.matlab/, Copyright 2008-2011
[43] Mondada, F. and Floreano, D. (1995) 'Evolution of neural control structures: some experiments on mobile robots', Robotics and Autonomous Systems, Vol. 16, pp.183-195.
[44] Yamada, S. (2005) 'Evolutionary behavior learning for action based environment modeling by a mobile robot', Applied Soft Computing, Vol. 5, pp.245-257.
[45] Çapi, G. (2007) 'Multiobjective evolution of neural controllers and task complexity', IEEE Transactions on Robotics, Vol. 23, No. 6, pp.1225-1234.
[46] Pratihar, D.K. (2003) 'Evolutionary robotics a review', Sadhana, Vol. 28, pp.999-1003.
[47] Goffe, W.L., Ferrier, G.D. and Rogers, J. (1994) 'Global optimization of statistical functions with simulated annealing', Journal of Economics, Vol. 60, pp.65-99.
[48] Koza, J. (1992) Genetic Programming. The MIT Press.
[49] Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization, Machine Learning. MA, USA: Addison-Wesley, Reading.
[50] Howard Demuth, Mark Beale. Neural Netuork Toolbox
[51] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Co., Inc., 1989
[52] C. R. Houck, J. Joines. and M.Kay. A genetic algotithm for function optimisation: A Matlab implementation. ACM Transactions on MathematicalSoftware, 1996, [Online]http://www.eos.ncsu.edu/eos/service/ie/research/kay_res/GAToolBox/gaot
[53] Whitley, *A Genetic Algorithm Tutorial*, Technical Report CS-93-103, Dept. of Computer Science, Colorado State University, 1993
[54] Dr. sc. Ahmet SHALA, RRJETAT FUZZY NEURALE, PRISHTINË, 2006

11. SHTOJCA

Shtojca 1

Programi eshte i perbere prej tre pjesesh qe jane Form1 (GUI) dhe dy module te cilat po i parashtroj ne vijim.

- Form1

```
VERSION 5.00
Begin VB.Form Form1
   Caption         =   "Simulimi i kontrollit te motorit me hapa"
   ClientHeight    =   4155
   ClientLeft      =   60
   ClientTop       =   450
   ClientWidth     =   5565
   ControlBox      =   0   'False
   LinkTopic       =   "Form1"
   ScaleHeight     =   4155
   ScaleWidth      =   5565
   StartUpPosition =   3  'Windows Default
   Begin VB.CommandButton cmdGjysemHap
      Caption         =   "Rrotullim 1/2 hap"
      Height          =   495
      Left            =   3360
      TabIndex        =   9
      Top             =   720
      Width           =   1935
   End
   Begin VB.CommandButton cmdNdaloMenjehere
      Caption         =   "Ndalim menjehere"
      Height          =   495
      Left            =   4320
      TabIndex        =   8
      Top             =   2040
      Width           =   975
   End
   Begin VB.CommandButton cmdDalje
      Caption         =   "Dalje"
      Height          =   495
      Left            =   3960
      TabIndex        =   7
      Top             =   3480
      Width           =   1335
   End
   Begin VB.CommandButton cmdZero
      Caption         =   "Gjendje Zero"
      Height          =   495
      Left            =   240
      TabIndex        =   6
      Top             =   3480
      Width           =   1815
   End
   Begin VB.TextBox TextInterval
      Height          =   285
      Left            =   1680
      TabIndex        =   4
      Top             =   2160
      Width           =   735
   End
   Begin VB.CheckBox Check1
      Caption         =   "Levizja sipas drejtimit tjeter"
      Height          =   255
```

```
    Left           =   1920
    TabIndex       =   3
    Top            =   1440
    Width          =   2295
  End
  Begin VB.CommandButton cmdRrotullimShpejt
    Caption        =   "Rrotullim sipas frekuences"
    Height         =   495
    Left           =   2520
    TabIndex       =   1
    Top            =   2040
    Width          =   1695
  End
  Begin VB.Timer Timer1
    Left           =   240
    Top            =   720
  End
  Begin VB.CommandButton cmdAktivizo
    Caption        =   "Frekuenca e rrotullimit  50 hz"
    Height         =   495
    Left           =   960
    TabIndex       =   0
    Top            =   720
    Width          =   2175
  End
  Begin VB.Label Label2
    Caption        =   "Frekuenca e rrotullimit"
    BeginProperty Font
      Name           =   "MS Sans Serif"
      Size           =   8.25
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    Height         =   255
    Left           =   240
    TabIndex       =   5
    Top            =   1920
    Width          =   2055
  End
  Begin VB.Label Label1
    Caption        =   "Motori me hapa (Stepped motor)"
    BeginProperty Font
      Name           =   "MS Sans Serif"
      Size           =   12
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    Height         =   375
    Left           =   120
    TabIndex       =   2
    Top            =   120
    Width          =   4095
  End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
```

```
Dim boolNdaloMenjehere      As Boolean
Dim portaddress             As Integer
Dim phase                   As Integer

Private GjendjeMotori(3)    As Byte


Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Sub SimpleTurnLeds()
  Dim i%
  Dim ii%

  For ii = 0 To 20
    For i = 0 To 3

      DoEvents

      Out portaddress, GjendjeMotori(i)
      Sleep (1 / CInt(TextInterval.Text)) * 1000

      If boolNdaloMenjehere = True Then
        GoTo NdaloMenjehere
      End If

    Next

    Sleep (1 / CInt(TextInterval.Text)) * 1000
  Next

  Exit Sub

NdaloMenjehere:
    Dim sHex As String
    Timer1.Enabled = False

    sHex = BinToDec("00000000")
    Out portaddress, CInt(sHex)
End Sub

Private Sub SimpleTurnLedsBack()
  Dim i%
  Dim ii%

  For ii = 0 To 20
    For i = 3 To 0 Step -1

      DoEvents

      Out portaddress, GjendjeMotori(i)
      Sleep (1 / CInt(TextInterval.Text)) * 1000

      If boolNdaloMenjehere = True Then
        GoTo NdaloMenjehere
      End If

    Next
    Sleep (1 / CInt(TextInterval.Text)) * 1000
  Next

  Exit Sub

NdaloMenjehere:
    Dim sHex As String
    Timer1.Enabled = False
```

```
    sHex = BinToDec("00000000")
    Out portaddress, CInt(sHex)
End Sub

Private Sub cmdAktivizo_Click()
    Timer1.Interval = 200
    Timer1.Enabled = True
End Sub

Private Sub cmdDalje_Click()
    Dim sHex As String
    Timer1.Enabled = False
    sHex = BinToDec("00000000")
    Out portaddress, CInt(sHex)
    End
End Sub

Private Sub cmdGjysemHap_Click()
  Dim sHex As String
  Timer1.Enabled = False

  TextInterval.Text = 100

  sHex = BinToDec("10000000")
  Out portaddress, CInt(sHex)

  Sleep (1 / CInt(TextInterval.Text)) * 1000

  sHex = BinToDec("11000000")
  Out portaddress, CInt(sHex)

  Sleep (1 / CInt(TextInterval.Text)) * 1000

  sHex = BinToDec("01000000")
  Out portaddress, CInt(sHex)

  Sleep (1 / CInt(TextInterval.Text)) * 1000

  sHex = BinToDec("01100000")
  Out portaddress, CInt(sHex)

  Sleep (1 / CInt(TextInterval.Text)) * 1000

  sHex = BinToDec("00100000")
  Out portaddress, CInt(sHex)

  Sleep (1 / CInt(TextInterval.Text)) * 1000

  sHex = BinToDec("00110000")
  Out portaddress, CInt(sHex)

  Sleep (1 / CInt(TextInterval.Text)) * 1000

  sHex = BinToDec("00010000")
  Out portaddress, CInt(sHex)

  Sleep (1 / CInt(TextInterval.Text)) * 1000

  sHex = BinToDec("10010000")
  Out portaddress, CInt(sHex)



End Sub

Private Sub cmdNdaloMenjehere_Click()
```

```
    boolNdaloMenjehere = True
End Sub

Private Sub cmdRrotullimShpejt_Click()
 boolNdaloMenjehere = False
 Timer1.Enabled = False

 If Check1.value = 1 Then
    SimpleTurnLedsBack
 Else
    SimpleTurnLeds
 End If

End Sub

Private Sub cmdZero_Click()
    Dim sHex As String
    Timer1.Enabled = False

    sHex = BinToDec("00000000")
    Out portaddress, CInt(sHex)
End Sub

Private Sub Form_Load()
  Dim sHex As String

  TextInterval.Text = 100

  sHex = BinToDec("11000000")
  GjendjeMotori(0) = CInt(sHex)

  sHex = BinToDec("01100000")
  GjendjeMotori(1) = CInt(sHex)

  sHex = BinToDec("00110000")
  GjendjeMotori(2) = CInt(sHex)

  sHex = BinToDec("10010000")
  GjendjeMotori(3) = CInt(sHex)

  portaddress = &H378 ' LPT1

End Sub

Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    If Button = 1 Then ' Kliko me te majten

        Check1.value = 0

        Out portaddress, GjendjeMotori(phase)

        If Check1.value = 1 Then
           phase = phase - 1
           If phase = -1 Then phase = 3
        Else
           phase = phase + 1
           If phase = 4 Then phase = 0
        End If

    ElseIf Button = 2 Then ' Kliko me te djathten

       Check1.value = 1

       Out portaddress, GjendjeMotori(phase)

        If Check1.value = 1 Then
```

```
            phase = phase - 1
            If phase = -1 Then phase = 3
        Else
            phase = phase + 1
            If phase = 4 Then phase = 0
        End If
    ElseIf Button = 4 Then ' Kliko ne rrotull

    End If
End Sub

Private Sub TextInterval_LostFocus()
    If IsNumeric(TextInterval.Text) = False Then TextInterval.Text = 50
    If TextInterval.Text = 0 Then TextInterval.Text = 50
End Sub

Private Sub Timer1_Timer()

  Out portaddress, GjendjeMotori(phase)

  If Check1.value = 1 Then
     phase = phase - 1
     If phase = -1 Then phase = 3
 Else
     phase = phase + 1
     If phase = 4 Then phase = 0
  End If
End Sub
```

-Moduli 1 (stepperMotor)

```
'Inp dhe Out deklarimet per porten I/O duke perdorur inpout32.dll.

Public Declare Function Inp Lib "inpout32.dll" _
Alias "Inp32" (ByVal portaddress As Integer) As Integer

Public Declare Sub Out Lib "inpout32.dll" _
Alias "Out32" (ByVal portaddress As Integer, ByVal value As Integer)
```

-Moduli 2 (modHexToBinary)

```
'           Shpjegimet              Funksionet
'          ------------------------------------------------------
'           Binary to Hex           BinToHex(BinNum As String)
'           Binary to Octal         BinToOct(BinNum As String)
'           Binary to Decimal       BinToDec(BinNum As String)
'           Hex to Binary           HexToBin(HexNum As String)
'           Octal to Binary         OctToBin(OctNum As String)
'           Decimal to Binary       DecToBin(DecNum As String)
'
'
Option Explicit

Function BinToHex(BinNum As String) As String
  Dim BinLen As Integer, i As Integer
  Dim HexNum As Variant

  On Error GoTo ErrorHandler

  BinLen = Len(BinNum)
```

```
  For i = BinLen To 1 Step -1
'   Check the string for invalid characters
    If Asc(Mid(BinNum, i, 1)) < 48 Or _
      Asc(Mid(BinNum, i, 1)) > 49 Then
      HexNum = ""
      Err.Raise 1002, "BinToHex", "Invalid Input"
    End If
'   Calculate HEX value of BinNum
    If Mid(BinNum, i, 1) And 1 Then
      HexNum = HexNum + 2 ^ Abs(i - BinLen)
    End If
  Next i
' Return HexNum as String
  BinToHex = Hex(HexNum)
ErrorHandler:
End Function

Function BinToOct(BinNum As String) As String
  Dim BinLen As Integer, i As Integer
  Dim OctNum As Variant

  On Error GoTo ErrorHandler

  BinLen = Len(BinNum)
  For i = BinLen To 1 Step -1
'   Check the string for invalid characters
    If Asc(Mid(BinNum, i, 1)) < 48 Or _
      Asc(Mid(BinNum, i, 1)) > 49 Then
      OctNum = ""
      Err.Raise 1002, "BinToOct", "Invalid Input"
    End If
'   Calculate Octal value of BinNum
    If Mid(BinNum, i, 1) And 1 Then
      OctNum = OctNum + 2 ^ Abs(i - BinLen)
    End If
  Next i
' Return OctNum as String
  BinToOct = Oct(OctNum)
ErrorHandler:
End Function

Public Function BinToDec(BinNum As String) As String
  Dim i As Integer
  Dim DecNum As Long

  On Error GoTo ErrorHandler

' Loop thru BinString
  For i = Len(BinNum) To 1 Step -1
'   Check the string for invalid characters
    If Asc(Mid(BinNum, i, 1)) < 48 Or _
      Asc(Mid(BinNum, i, 1)) > 49 Then
      DecNum = ""
      Err.Raise 1002, "BinToDec", "Invalid Input"
    End If
'   If bit is 1 then raise 2^LoopCount and add it to DecNum
    If Mid(BinNum, i, 1) And 1 Then
      DecNum = DecNum + 2 ^ (Len(BinNum) - i)
    End If
  Next i
' Return DecNum as a String
  BinToDec = DecNum
ErrorHandler:
End Function

Public Function OctToBin(OctNum As String) As String
```

```
  Dim BinNum As String
  Dim lOctNum As Long
  Dim i As Integer

  On Error GoTo ErrorHandler
' Check the string for invalid characters
  For i = 1 To Len(OctNum)
    If (Asc(Mid(OctNum, i, 1)) < 48 Or Asc(Mid(OctNum, i, 1)) > 55) Then
      BinNum = ""
      Err.Raise 1008, "OctToBin", "Invalid Input"
    End If
  Next i

  i = 0
  lOctNum = Val("&O" & OctNum)

  Do
    If lOctNum And 2 ^ i Then
      BinNum = "1" & BinNum
    Else
      BinNum = "0" & BinNum
    End If
    i = i + 1
  Loop Until 2 ^ i > lOctNum
' Return BinNum as a String
  OctToBin = BinNum
ErrorHandler:
End Function

Public Function DecToBin(DecNum As String) As String
  Dim BinNum As String
  Dim lDecNum As Long
  Dim i As Integer

  On Error GoTo ErrorHandler

' Check the string for invalid characters
  For i = 1 To Len(DecNum)
    If Asc(Mid(DecNum, i, 1)) < 48 Or _
      Asc(Mid(DecNum, i, 1)) > 57 Then
      BinNum = ""
      Err.Raise 1010, "DecToBin", "Invalid Input"
    End If
  Next i

  i = 0
  lDecNum = Val(DecNum)

  Do
    If lDecNum And 2 ^ i Then
      BinNum = "1" & BinNum
    Else
      BinNum = "0" & BinNum
    End If
    i = i + 1
  Loop Until 2 ^ i > lDecNum
' Return BinNum as a String
  DecToBin = BinNum
ErrorHandler:
End Function

Public Function HexToBin(HexNum As String) As String
  Dim BinNum As String
  Dim lHexNum As Long
  Dim i As Integer
```

```
  On Error GoTo ErrorHandler

' Check the string for invalid characters
  For i = 1 To Len(HexNum)
    If ((Asc(Mid(HexNum, i, 1)) < 48) Or _
       (Asc(Mid(HexNum, i, 1)) > 57 And _
        Asc(UCase(Mid(HexNum, i, 1))) < 65) Or _
       (Asc(UCase(Mid(HexNum, i, 1))) > 70)) Then
      BinNum = ""
      Err.Raise 1016, "HexToBin", "Invalid Input"
    End If
  Next i

  i = 0
  lHexNum = Val("&h" & HexNum)
  Do
    If lHexNum And 2 ^ i Then
      BinNum = "1" & BinNum
    Else
      BinNum = "0" & BinNum
    End If
    i = i + 1
  Loop Until 2 ^ i > lHexNum
' Return BinNum as a String
  HexToBin = BinNum
ErrorHandler:
End Function
```

Shtojca 2

Komandimi i motorit me hapa nëpërmjet modulit M2 me microcontroller WiZ232-A në gjuhën Qbasic.

```
Option Explicit

' Ky program tregon se si mund te implementohet motori me hapa
' duke perdorur funksionet e ndertuara per microcontroller WiZ232-A.
' Materiali bazohet ne manualin e dhene qe gjendet ne faqen http://www.rmv.com

Const ZeroPos = 8388608 ' used for Shaft Position display
Cls

Start:
Print "Apply power to the M2 (or reset it) and press a key to start"
While INKEY$ = "": Wend
On Error GoTo ResumeError
ResumeError:
Resume Next 'Otherwise, holding a key down might result
'in an I/O error from QBasic
' Let the user to select which port to open, COM1 or COM2
Do
   LOCATE 10, 10
   Print "Select COM port (1) (2): ";
   Do
      comport = Val(INKEY$)
   Loop While comport = 0
Loop While comport <> 1 And comport <> 2
Print comport
' Now open COM Port as selected with a 2048-byte buffer in Tx and Rx
Select Case comport
   Case 1
   Open "COM1:9600,N,8,1,CD0,CS0,DS0,OP0,RS,TB2048,RB2048" For Random As #1
   Case 2
   Open "COM2:9600,N,8,1,CD0,CS0,DS0,OP0,RS,TB2048,RB2048" For Random As #1
End Select
' Set the WiZ232-A in program mode with <C>onfigure <R>esults <A>SCII
' <P>rogram, required for READSERIAL in order to decode incoming
' strings properly.
W$ = "CRAP": GoSub WriteSerial
' Show Main Screen
Cls
Print
LOCATE 5, 5: Print "RMV Electronics Inc.-*- M2 Engine"
LOCATE 6, 5: Print "Application Example: Stepper Motor Controller"
LOCATE 10, 5: Print " L: Move Left 100 steps R: Move Right 100 steps,"
LOCATE 11, 5: Print " l: Move left 1 step r: move rigth 1 step"
LOCATE 12, 5: Print "Esc: Exit"
' Enable timer for automatically showing the shaft position
TIMER ON
On Timer(1) GoSub ShowPosition
Do ' Main Loop begins here
   Do ' Wait for a user keystroke
      K$ = INKEY$
   Loop While K$ = ""

   While WaitMotor = 1: Wend ' or if motor is busy

   Select Case K$ ' process user's requirement
   Case "R"
      PLAY "O3C16F64"
      M$ = "SH+500;800;1;200" ' move 500 steps CW or right
      K$ = ""
```

```
   Case "r"
      M$ = "SH+1;800;1;200" ' move 1 step CW or right
      K$ = ""
   Case "L"
      PLAY "O3F16C64"
      M$ = "SH-500;800;1;200" ' move 500 steps CCW or left
      K$ = ""
   Case "l"
      M$ = "SH-1;800;1;200" ' move 1 step CW or left
      K$ = ""
   Case Chr$(27)
      Close: End  ' End pressing Esc
   End Select
   If K$ = "" Then
   'Get motor status and check motion completed
   'if status OK (=1) send motion command
   'PortBusy acts as semaphore for using the serial port
   W$ = "S?"
   PortBusy = 1: GoSub WriteSerial
   'V returns the status of the motor: V=1 Motor ready to accept commands
   If V = 1 Then W$ = M$: GoSub WriteSerial
      PortBusy = 0
   End If
Loop 'Main Loop ends here

'** SUBROUNTINES **
ShowPosition:
' Displays the shaft's position and motor status once every second
If PortBusy = 0 Then
   W$ = "S?": GoSub WriteSerial
   MotorStatus = V
   W$ = "SN": GoSub WriteSerial
   Select Case V
      Case Is < 0
         LOCATE 16, 10
         Print "Motor Disabled"
      Case Is > ZeroPos
         Position = ZeroPos - V
      Case Is < ZeroPos
         Position = V
   End Select
   If V >= 0 Then
      If MotorStatus = 1 Then WaitMotor = 0 Else WaitMotor = 1
      LOCATE 15, 10: Print "Shaft Position:";: Print USING; "########"; Position
      W$ = "S?": GoSub WriteSerial
      LOCATE 16, 10: Print " Motor Status: ";
      If MotorStatus = 1 Then Print "Idle " Else Print "Moving"
   End If
End If
Return

' Write command to serial port. Returns the value fetched from the
' WiZ232-A in V (in case of error legend is returned in ERRORCODE$)
WriteSerial:
   ' WARNING: When the WiZ232-A is not working in program mode
   ' (command=CRAP[CR]), it returns CR and LF which prevent READSERIAL
   ' below to fetch function values properly. Thus,
   ' ALWAYS do W$="CRAP":GOSUB WRITESERIAL at the onset
   Rem PRINT W$; ' Remove REM to see the commands on the screen
   Print #1, W$ ' Send the string out (PRINT automatically adds a CR)
   GoSub READSERIAL ' and then fetch the response from the WiZ232-A
Return

' Reading serial port. NEVER call this routine directly
READSERIAL:
   S$ = ""
```

```
    If Loc(1) = 0 Then GoTo READSERIAL ' Loop until a character is received
    ' Get received string into S$
Lp1:
    C$ = Input$(1, #1) ' Fetch from the COM buffer one character at the time
    S$ = S$ + C$ ' and add it at the end of string$ S$
    If C$ <> ">" Then GoTo Lp1 ' ">" is always sent last by WiZ232-A
    Rem PRINT S$ ' Remove REM to see the strings returned on the screen
    'Decode string (V$) and value (V)
    VALIDERROR = True
    ERRORCODE$ = ""
    V$ = ""
    For H = 1 To Len(S$)
       If Mid$(S$, H, 1) = Chr$(7) Then VALIDERROR = False
       If Mid$(S$, H, 1) = "?" Then ERRORCODE$ = Mid$(S$, H + 1, 1)
    Next H
    If (VALIDERROR = True And ERRORCODE$ <> "") Then GoSub ERRORSUB: Return
    V$ = ""
    For n = 1 To Len(S$)
       x$ = Mid$(S$, n, 1) ' Get rid of CR and LF if any
       If x$ <> Chr$(13) Then If x$ >= "0" And x$ <= "9" Then V$ = V$ + x$
    Next n
    ' V$ = LEFT$(V$, LEN(V$) - 1) ' Get rid of the '>'
    V = Val(V$) ' and save results
Return

ERRORSUB:
    V = -1 ' this allows the program to handle the error
Return
' FUND
```

Komandimi i motorit me hapa nëpërmjet modulit M2 me microcontroller WiZ232-A në gjuhën Visual Basic 6.0 duke përdorur librarinë ActiveComport Serial.

GUI dhe modulet.

‘ frmMainPanel.frm

```
VERSION 5.00
Object = "{6B7E6392-850A-101B-AFC0-4210102A8DA7}#1.3#0"; "COMCTL32.OCX"
Begin VB.Form frmMainPanel
   Caption        =   "Komandimi i motorit me hapa     E2- MODUL"
   ClientHeight   =   7320
   ClientLeft     =   60
   ClientTop      =   450
   ClientWidth    =   9525
   LinkTopic      =   "Form1"
   ScaleHeight    =   7320
   ScaleWidth     =   9525
   StartUpPosition =   3  'Windows Default
   Begin VB.TextBox txtRead
      Appearance     =   0  'Flat
      BackColor      =   &H80000000&
      BeginProperty Font
         Name           =   "MS Sans Serif"
         Size           =   9.75
         Charset        =   0
         Weight         =   700
         Underline      =   0   'False
         Italic         =   0   'False
         Strikethrough  =   0   'False
      EndProperty
      Height         =   5880
      Left           =   4920
      TabIndex       =   14
```

```
  Top             =   1320
  Width           =   4455
End
Begin VB.TextBox txtHapa
  BeginProperty Font
    Name            =   "MS Sans Serif"
    Size            =   9.75
    Charset         =   0
    Weight          =   700
    Underline       =   0   'False
    Italic          =   0   'False
    Strikethrough   =   0   'False
  EndProperty
  Height          =   360
  Left            =   3600
  TabIndex        =   12
  Top             =   3720
  Width           =   735
End
Begin VB.CommandButton cmdStatusMotori
  Caption         =   "Statusi i motorit"
  BeginProperty Font
    Name            =   "Verdana"
    Size            =   8.25
    Charset         =   0
    Weight          =   400
    Underline       =   0   'False
    Italic          =   0   'False
    Strikethrough   =   0   'False
  EndProperty
  Height          =   495
  Left            =   480
  TabIndex        =   10
  Top             =   6600
  Width           =   1815
End
Begin VB.TextBox txtShpejtesia
  Alignment       =   2  'Center
  Appearance      =   0  'Flat
  BackColor       =   &H8000000F&
  BeginProperty Font
    Name            =   "Verdana"
    Size            =   9.75
    Charset         =   0
    Weight          =   400
    Underline       =   0   'False
    Italic          =   0   'False
    Strikethrough   =   0   'False
  EndProperty
  Height          =   375
  Left            =   3600
  TabIndex        =   8
  Top             =   5340
  Width           =   735
End
Begin ComctlLib.Slider SliderShpejtesia
  Height          =   675
  Left            =   360
  TabIndex        =   7
  Top             =   5880
  Width           =   4215
  _ExtentX        =   7435
  _ExtentY        =   1111
  _Version        =   327682
  Max             =   50
End
```

```
Begin VB.ComboBox cmbPorta
  BeginProperty Font
    Name            =   "Verdana"
    Size            =   9
    Charset         =   0
    Weight          =   400
    Underline       =   0   'False
    Italic          =   0   'False
    Strikethrough   =   0   'False
  EndProperty
  Height          =   330
  Left            =   3720
  Style           =   2  'Dropdown List
  TabIndex        =   6
  Top             =   240
  Width           =   975
End
Begin VB.CommandButton buttonOpen
  Caption         =   "Hapje porte"
  BeginProperty Font
    Name            =   "Verdana"
    Size            =   8.25
    Charset         =   0
    Weight          =   400
    Underline       =   0   'False
    Italic          =   0   'False
    Strikethrough   =   0   'False
  EndProperty
  Height          =   495
  Left            =   2880
  TabIndex        =   4
  Top             =   720
  Width           =   1815
End
Begin VB.CommandButton buttonClose
  Caption         =   "Mbyllje porte"
  BeginProperty Font
    Name            =   "Verdana"
    Size            =   8.25
    Charset         =   0
    Weight          =   400
    Underline       =   0   'False
    Italic          =   0   'False
    Strikethrough   =   0   'False
  EndProperty
  Height          =   495
  Left            =   2640
  TabIndex        =   3
  Top             =   6600
  Width           =   1815
End
Begin VB.CommandButton btStop
  Height          =   855
  Left            =   1800
  Picture         =   "frmMainPanel.frx":0000
  Style           =   1  'Graphical
  TabIndex        =   2
  Top             =   4320
  Width           =   1215
End
Begin VB.CommandButton btRrotullimdjathtas
  Height          =   855
  Left            =   3120
  Picture         =   "frmMainPanel.frx":0631
  Style           =   1  'Graphical
  TabIndex        =   1
```

```
  Top           =   4320
  Width         =   1215
End
Begin VB.CommandButton btRrotullimmajtas
  Height        =   855
  Left          =   480
  Picture       =   "frmMainPanel.frx":0C6E
  Style         =   1  'Graphical
  TabIndex      =   0
  Top           =   4320
  Width         =   1215
End
Begin VB.Label Label5
  Alignment     =   2  'Center
  Caption       =   "Pergjigjet e komunikimit me modulin E2"
  BeginProperty Font
    Name          =   "Verdana"
    Size          =   11.25
    Charset       =   0
    Weight        =   700
    Underline     =   0   'False
    Italic        =   0   'False
    Strikethrough =   0   'False
  EndProperty
  Height        =   735
  Left          =   4920
  TabIndex      =   15
  Top           =   600
  Width         =   4335
End
Begin VB.Label Label4
  Caption       =   "Koha e hapit prej hapit"
  BeginProperty Font
    Name          =   "Verdana"
    Size          =   11.25
    Charset       =   0
    Weight        =   700
    Underline     =   0   'False
    Italic        =   0   'False
    Strikethrough =   0   'False
  EndProperty
  Height        =   375
  Left          =   360
  TabIndex      =   13
  Top           =   5400
  Width         =   3135
End
Begin VB.Image Image1
  Height        =   2130
  Left          =   240
  Picture       =   "frmMainPanel.frx":12B1
  Top           =   1320
  Width         =   4500
End
Begin VB.Label Label3
  Caption       =   "Hapa qe duhen te kryhen"
  BeginProperty Font
    Name          =   "Verdana"
    Size          =   11.25
    Charset       =   0
    Weight        =   700
    Underline     =   0   'False
    Italic        =   0   'False
    Strikethrough =   0   'False
  EndProperty
  Height        =   375
```

```
    Left           =   480
    TabIndex       =   11
    Top            =   3720
    Width          =   3135
  End
  Begin VB.Label Label2
    Caption        =   "ms"
    BeginProperty Font
      Name           =   "Verdana"
      Size           =   11.25
      Charset        =   0
      Weight         =   400
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    Height         =   255
    Left           =   4440
    TabIndex       =   9
    Top            =   5400
    Width          =   375
  End
  Begin VB.Label Label1
    Caption        =   "Porta e komunikimit"
    BeginProperty Font
      Name           =   "Verdana"
      Size           =   11.25
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    Height         =   255
    Left           =   240
    TabIndex       =   5
    Top            =   240
    Width          =   2775
  End
End
Attribute VB_Name = "frmMainPanel"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub btRrotullimdjathtas_Click()
   If objComport.IsOpened = True Then
    ' Rrotullim djathtas
     WriteSerial "SH+" & CInt(txtHapa.Text) & ";800;1;200"
   Else
     MsgBox "Porta jo e hapur !!! " & Err.Description, vbCritical, App.Title
   End If
End Sub

Private Sub btRrotullimmajtas_Click()
   If objComport.IsOpened = True Then
   ' Rrotullim majtas
     WriteSerial "SH-" & CInt(txtHapa.Text) & ";800;1;200"
   Else
     MsgBox "Porta jo e hapur !!! " & Err.Description, vbCritical, App.Title
   End If
End Sub

Private Sub btStop_Click()
```

```
   If objComport.IsOpened = True Then
    ' stop the robot

   Else
     MsgBox "Porta jo e hapur !!! " & Err.Description, vbCritical, App.Title
   End If
End Sub

Private Sub buttonClose_Click()
   If objComport.IsOpened Then
     objComport.Close
   End If
   End
End Sub

Private Sub buttonOpen_Click()
   If OpenPortCommunication(Mid(cmbPorta.Text, 4)) = True Then
      ' Set the WiZ232-A in program mode with <C>onfigure <R>esults <A>SCII
      ' <P>rogram, required for READSERIAL in order to decode incoming
      ' strings properly.
      WriteSerial "CRAP" & Chr(13)
   Else
     MsgBox "Porta jo e hapur !!! " & Err.Description, vbCritical, App.Title
   End If
End Sub

Public Function WriteSerial(Wd As String) As Boolean
   ' Dim i   As Integer

   ' Dergimi i te dhenave duke shtuar dhe nje CR (karakterin 13))

   objComport.WriteString Wd & Chr(13)

   " Dergim i informacionit byte pas byte
   ' For i = 1 To Len(Wd)
   '    x$ = Mid$(Wd, i, 1)
   '    objComport.WriteByte Asc(x$)
   ' Next
   ' objComport.WriteByte 13

   Call ReadSerial ' and then fetch the response from the WiZ232-A
End Function

Public Function ReadSerial() As String
   Dim ReadByte As Integer
   Dim h, n As Integer
   Dim C$, s$, ERRORCODE$, V$, x$

   Dim VALIDERROR  As Boolean
   VALIDERROR = False

Lp1:
   ReadByte = objComport.ReadByte ' Merr nga COM buffer nje karakter ne cdo kohe
   C$ = Chr(ReadByte)
   s$ = s$ + C$ ' and add it at the end of string$ S$
   If C$ <> ">" Then GoTo Lp1 ' ">" is always sent last by WiZ232-A
   Rem PRINT S$ ' Remove REM to see the strings returned on the screen
   'Decode string (V$) and value (V)
   VALIDERROR = True
   ERRORCODE$ = ""
   V$ = ""
   For h = 1 To Len(s$)
      If Mid$(s$, h, 1) = Chr$(7) Then VALIDERROR = False
      If Mid$(s$, h, 1) = "?" Then ERRORCODE$ = Mid$(s$, h + 1, 1)
   Next h
   If (VALIDERROR = True And ERRORCODE$ <> "") Then GoSub ERRORSUB: Return
```

```
    V$ = ""
    For n = 1 To Len(s$)
        x$ = Mid$(s$, n, 1) ' Get rid of CR and LF if any
        If x$ <> Chr$(13) Then If x$ >= "0" And x$ <= "9" Then V$ = V$ + x$
    Next n
    ' V$ = LEFT$(V$, LEN(V$) - 1) ' Get rid of the '>'
    V = Val(V$) ' and save results

    txtRead.Text = txtRead.Text & V$

    ReadSerial = V$
    Exit Function

ERRORSUB:
    V = -1 ' Kontrollimi i gabimit

    txtRead.Text = txtRead.Text & V$

    ReadSerial = "-1"
End Function

Private Sub cmdStatusMotori_Click()
    ' Kerkon statusin e motorit
    ' 1 - levizje , 0 - ndaluar
    WriteSerial "S?"
End Sub

Private Sub Form_Load()
    cmbPorta.Clear
    cmbPorta.AddItem "COM1"
    cmbPorta.AddItem "COM2"
    cmbPorta.AddItem "COM3"
    cmbPorta.AddItem "COM4"
    cmbPorta.AddItem "COM5"
    cmbPorta.AddItem "COM6"
    cmbPorta.AddItem "COM7"
    cmbPorta.AddItem "COM8"
    cmbPorta.AddItem "COM9"
    cmbPorta.AddItem "COM10"
    cmbPorta.ListIndex = 0

    txtHapa.Text = "100"

    SliderShpejtesia.Value = SliderShpejtesia.Max
    txtShpejtesia.Text = SliderShpejtesia.Value * 10

    Set objComport = CreateObject("ActiveXperts.ComPort")
    objComport.ComTimeout = 200

End Sub

Private Sub SliderShpejtesia_Click()
    txtShpejtesia.Text = SliderShpejtesia.Value * 10
End Sub

Private Sub SliderShpejtesia_Scroll()
    txtShpejtesia.Text = SliderShpejtesia.Value * 10
End Sub
```

‘ modOpenPort.mod

```
Option Explicit

' Krijuar nga Genti Progri
'                   Date : 01/12/2013
```

```
' Funksioni hapes i portes se komunikimit me nepermjet E2-MODUL

Global objComport As ACOMPORTLib.ComPort

Public Function OpenPortCommunication(my_COM As String) As Boolean
    Dim Key             As Integer
    Dim s               As String
    Dim comm            As String
    Dim confirmation    As String

    ' Output buffer must be big enough to take largest message size
    If IsNumeric(my_COM) = False Then
       MsgBox "Porte komunikimi e pa percaktuar."
       OpenPortCommunication = False
    End If

    ' Establishing connection to Roomba...
    comm = "COM" & my_COM


    objComport.Device = comm

    objComport.ComTimeout = 1000

    objComport.LogFile = App.Path & "\Errors.Log"

    ' defaults at 9600, can change
    objComport.BaudRate = 9600

    objComport.HardwareFlowControl = 0
    objComport.SoftwareFlowControl = 0

    objComport.DataBits = objComport.asDATABITS_8
    objComport.StopBits = objComport.asSTOPBITS_1
    objComport.Parity = objComport.asPARITY_NONE

   ' objComport.RaiseDTR (0)
   ' objComport.RaiseRTS (0)

    objComport.Open

    confirmation = GetResult


    If objComport.IsOpened = True And objComport.LastError = 0 Then
       OpenPortCommunication = True
    Else
       MsgBox "Pamundesi ne hapjen e portes se komunikimit."
       OpenPortCommunication = False
    End If
End Function

Public Function GetResult() As String
    If objComport.LastError = 0 Then
       GetResult = "SUCCESS"
    Else
       GetResult = "ERROR " & objComport.LastError & " ( " & objComport.GetErrorDescription(objComport.LastError) & " )"
    End If
End Function
```

Shtojca 3

```
'=============================================================================================================
' Programi i kontrollit te dy motoreve nepermjet portes USB ne Visual Basic duke perdorur modulin
'                                        Stepper-Bee
'=============================================================================================================
'
Public Class Form1
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents M1Steps As System.Windows.Forms.TextBox
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents M1Interval As System.Windows.Forms.TextBox
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Run1 As System.Windows.Forms.Button
    Friend WithEvents GroupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents M1Reverse As System.Windows.Forms.CheckBox
    Friend WithEvents M1Output1 As System.Windows.Forms.CheckBox
    Friend WithEvents M1Output2 As System.Windows.Forms.CheckBox
    Friend WithEvents M1Output3 As System.Windows.Forms.CheckBox
    Friend WithEvents Stop1 As System.Windows.Forms.Button
    Friend WithEvents Label3 As System.Windows.Forms.Label
    Friend WithEvents M2Interval As System.Windows.Forms.TextBox
    Friend WithEvents Label4 As System.Windows.Forms.Label
    Friend WithEvents M2Steps As System.Windows.Forms.TextBox
    Friend WithEvents GroupBox2 As System.Windows.Forms.GroupBox
    Friend WithEvents Stop2 As System.Windows.Forms.Button
    Friend WithEvents M2Output3 As System.Windows.Forms.CheckBox
    Friend WithEvents M2Output2 As System.Windows.Forms.CheckBox
    Friend WithEvents M2Output1 As System.Windows.Forms.CheckBox
    Friend WithEvents M2Reverse As System.Windows.Forms.CheckBox
    Friend WithEvents Run2 As System.Windows.Forms.Button
    Friend WithEvents M1StepMode As System.Windows.Forms.ComboBox
    Friend WithEvents M2StepMode As System.Windows.Forms.ComboBox
    Friend WithEvents Label5 As System.Windows.Forms.Label
```

```
    Friend WithEvents Label6 As System.Windows.Forms.Label
    Friend WithEvents SetMode As System.Windows.Forms.Button
    Friend WithEvents GroupBox3 As System.Windows.Forms.GroupBox
    Friend WithEvents M1StepsLeft As System.Windows.Forms.TextBox
    Friend WithEvents M2StepsLeft As System.Windows.Forms.TextBox
    Friend WithEvents M1Active As System.Windows.Forms.CheckBox
    Friend WithEvents M2Active As System.Windows.Forms.CheckBox
    Friend WithEvents Label7 As System.Windows.Forms.Label
    Friend WithEvents Label8 As System.Windows.Forms.Label
    Friend WithEvents GroupBox4 As System.Windows.Forms.GroupBox
    Friend WithEvents GroupBox5 As System.Windows.Forms.GroupBox
    Friend WithEvents Input1 As System.Windows.Forms.CheckBox
    Friend WithEvents Input2 As System.Windows.Forms.CheckBox
    Friend WithEvents Input3 As System.Windows.Forms.CheckBox
    Friend WithEvents Input4 As System.Windows.Forms.CheckBox
    Friend WithEvents Input5 As System.Windows.Forms.CheckBox
    Friend WithEvents Label9 As System.Windows.Forms.Label
    Friend WithEvents Label10 As System.Windows.Forms.Label
    Friend WithEvents Label11 As System.Windows.Forms.Label
    Friend WithEvents Label12 As System.Windows.Forms.Label
    Friend WithEvents Label13 As System.Windows.Forms.Label
    Friend WithEvents Label14 As System.Windows.Forms.Label
    Friend WithEvents GroupBox6 As System.Windows.Forms.GroupBox
    Friend WithEvents GroupBox7 As System.Windows.Forms.GroupBox
    Friend WithEvents GetMotorStatus As System.Windows.Forms.Button
    Friend WithEvents Label15 As System.Windows.Forms.Label
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.Button1 = New System.Windows.Forms.Button()
        Me.M1Steps = New System.Windows.Forms.TextBox()
        Me.Label1 = New System.Windows.Forms.Label()
        Me.M1Interval = New System.Windows.Forms.TextBox()
        Me.Label2 = New System.Windows.Forms.Label()
        Me.Run1 = New System.Windows.Forms.Button()
        Me.GroupBox1 = New System.Windows.Forms.GroupBox()
        Me.Stop1 = New System.Windows.Forms.Button()
        Me.M1Output3 = New System.Windows.Forms.CheckBox()
        Me.M1Output2 = New System.Windows.Forms.CheckBox()
        Me.M1Output1 = New System.Windows.Forms.CheckBox()
        Me.M1Reverse = New System.Windows.Forms.CheckBox()
        Me.Label3 = New System.Windows.Forms.Label()
        Me.M2Interval = New System.Windows.Forms.TextBox()
        Me.Label4 = New System.Windows.Forms.Label()
        Me.M2Steps = New System.Windows.Forms.TextBox()
        Me.GroupBox2 = New System.Windows.Forms.GroupBox()
        Me.Stop2 = New System.Windows.Forms.Button()
        Me.M2Output3 = New System.Windows.Forms.CheckBox()
        Me.M2Output2 = New System.Windows.Forms.CheckBox()
        Me.M2Output1 = New System.Windows.Forms.CheckBox()
        Me.M2Reverse = New System.Windows.Forms.CheckBox()
        Me.Run2 = New System.Windows.Forms.Button()
        Me.M1StepMode = New System.Windows.Forms.ComboBox()
        Me.M2StepMode = New System.Windows.Forms.ComboBox()
        Me.Label5 = New System.Windows.Forms.Label()
        Me.Label6 = New System.Windows.Forms.Label()
        Me.SetMode = New System.Windows.Forms.Button()
        Me.GroupBox3 = New System.Windows.Forms.GroupBox()
        Me.M1StepsLeft = New System.Windows.Forms.TextBox()
        Me.M2StepsLeft = New System.Windows.Forms.TextBox()
        Me.M1Active = New System.Windows.Forms.CheckBox()
        Me.M2Active = New System.Windows.Forms.CheckBox()
        Me.Label7 = New System.Windows.Forms.Label()
        Me.Label8 = New System.Windows.Forms.Label()
        Me.GroupBox4 = New System.Windows.Forms.GroupBox()
        Me.GroupBox5 = New System.Windows.Forms.GroupBox()
        Me.Input1 = New System.Windows.Forms.CheckBox()
```

```
Me.Input2 = New System.Windows.Forms.CheckBox()
Me.Input3 = New System.Windows.Forms.CheckBox()
Me.Input4 = New System.Windows.Forms.CheckBox()
Me.Input5 = New System.Windows.Forms.CheckBox()
Me.Label9 = New System.Windows.Forms.Label()
Me.Label10 = New System.Windows.Forms.Label()
Me.Label11 = New System.Windows.Forms.Label()
Me.Label12 = New System.Windows.Forms.Label()
Me.Label13 = New System.Windows.Forms.Label()
Me.Label14 = New System.Windows.Forms.Label()
Me.GroupBox6 = New System.Windows.Forms.GroupBox()
Me.GetMotorStatus = New System.Windows.Forms.Button()
Me.GroupBox7 = New System.Windows.Forms.GroupBox()
Me.Label15 = New System.Windows.Forms.Label()
Me.GroupBox1.SuspendLayout()
Me.GroupBox2.SuspendLayout()
Me.GroupBox3.SuspendLayout()
Me.GroupBox6.SuspendLayout()
Me.SuspendLayout()
'
'Button1
'
Me.Button1.Location = New System.Drawing.Point(376, 4)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(96, 23)
Me.Button1.TabIndex = 0
Me.Button1.Text = "Inicializimi"
'
'M1Steps
'
Me.M1Steps.Location = New System.Drawing.Point(216, 59)
Me.M1Steps.Name = "M1Steps"
Me.M1Steps.Size = New System.Drawing.Size(64, 20)
Me.M1Steps.TabIndex = 1
Me.M1Steps.Text = "1"
'
'Label1
'
Me.Label1.Location = New System.Drawing.Point(51, 62)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(157, 13)
Me.Label1.TabIndex = 2
Me.Label1.Text = "Numri i hapave qe do te behen"
Me.Label1.TextAlign = System.Drawing.ContentAlignment.TopRight
'
'M1Interval
'
Me.M1Interval.Location = New System.Drawing.Point(216, 91)
Me.M1Interval.Name = "M1Interval"
Me.M1Interval.Size = New System.Drawing.Size(64, 20)
Me.M1Interval.TabIndex = 3
Me.M1Interval.Text = "1"
'
'Label2
'
Me.Label2.Location = New System.Drawing.Point(32, 91)
Me.Label2.Name = "Label2"
Me.Label2.Size = New System.Drawing.Size(176, 16)
Me.Label2.TabIndex = 4
Me.Label2.Text = "Koha ndermejt hapave (ms)"
Me.Label2.TextAlign = System.Drawing.ContentAlignment.TopRight
'
'Run1
'
Me.Run1.Location = New System.Drawing.Point(48, 120)
```

```
Me.Run1.Name = "Run1"
Me.Run1.Size = New System.Drawing.Size(75, 23)
Me.Run1.TabIndex = 9
Me.Run1.Text = "Fillo"
'
'GroupBox1
'
Me.GroupBox1.Controls.Add(Me.Stop1)
Me.GroupBox1.Controls.Add(Me.M1Output3)
Me.GroupBox1.Controls.Add(Me.M1Output2)
Me.GroupBox1.Controls.Add(Me.M1Output1)
Me.GroupBox1.Controls.Add(Me.M1Reverse)
Me.GroupBox1.Controls.Add(Me.Run1)
Me.GroupBox1.Location = New System.Drawing.Point(16, 35)
Me.GroupBox1.Name = "GroupBox1"
Me.GroupBox1.Size = New System.Drawing.Size(304, 200)
Me.GroupBox1.TabIndex = 10
Me.GroupBox1.TabStop = False
Me.GroupBox1.Text = "Motor 1"
'
'Stop1
'
Me.Stop1.Location = New System.Drawing.Point(48, 160)
Me.Stop1.Name = "Stop1"
Me.Stop1.Size = New System.Drawing.Size(75, 23)
Me.Stop1.TabIndex = 13
Me.Stop1.Text = "Ndalo"
'
'M1Output3
'
Me.M1Output3.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M1Output3.Location = New System.Drawing.Point(184, 168)
Me.M1Output3.Name = "M1Output3"
Me.M1Output3.Size = New System.Drawing.Size(72, 24)
Me.M1Output3.TabIndex = 12
Me.M1Output3.Text = "Output 3"
'
'M1Output2
'
Me.M1Output2.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M1Output2.Location = New System.Drawing.Point(184, 144)
Me.M1Output2.Name = "M1Output2"
Me.M1Output2.Size = New System.Drawing.Size(72, 24)
Me.M1Output2.TabIndex = 11
Me.M1Output2.Text = "Output 2"
'
'M1Output1
'
Me.M1Output1.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M1Output1.Location = New System.Drawing.Point(184, 120)
Me.M1Output1.Name = "M1Output1"
Me.M1Output1.Size = New System.Drawing.Size(72, 24)
Me.M1Output1.TabIndex = 10
Me.M1Output1.Text = "Output 1"
'
'M1Reverse
'
Me.M1Reverse.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M1Reverse.Location = New System.Drawing.Point(98, 88)
Me.M1Reverse.Name = "M1Reverse"
Me.M1Reverse.Size = New System.Drawing.Size(118, 24)
Me.M1Reverse.TabIndex = 0
Me.M1Reverse.Text = "Rootullim mbrapsh"
'
'Label3
```

```
'
Me.Label3.Location = New System.Drawing.Point(32, 307)
Me.Label3.Name = "Label3"
Me.Label3.Size = New System.Drawing.Size(176, 16)
Me.Label3.TabIndex = 14
Me.Label3.Text = "Koha ndermejt hapave (ms)"
Me.Label3.TextAlign = System.Drawing.ContentAlignment.TopRight
'
'M2Interval
'
Me.M2Interval.Location = New System.Drawing.Point(216, 307)
Me.M2Interval.Name = "M2Interval"
Me.M2Interval.Size = New System.Drawing.Size(64, 20)
Me.M2Interval.TabIndex = 13
Me.M2Interval.Text = "1"
'
'Label4
'
Me.Label4.Location = New System.Drawing.Point(32, 27)
Me.Label4.Name = "Label4"
Me.Label4.Size = New System.Drawing.Size(160, 17)
Me.Label4.TabIndex = 12
Me.Label4.Text = "Numri i hapave qe do te behen"
Me.Label4.TextAlign = System.Drawing.ContentAlignment.TopRight
'
'M2Steps
'
Me.M2Steps.Location = New System.Drawing.Point(216, 275)
Me.M2Steps.Name = "M2Steps"
Me.M2Steps.Size = New System.Drawing.Size(64, 20)
Me.M2Steps.TabIndex = 11
Me.M2Steps.Text = "1"
'
'GroupBox2
'
Me.GroupBox2.Controls.Add(Me.Stop2)
Me.GroupBox2.Controls.Add(Me.M2Output3)
Me.GroupBox2.Controls.Add(Me.M2Output2)
Me.GroupBox2.Controls.Add(Me.M2Output1)
Me.GroupBox2.Controls.Add(Me.M2Reverse)
Me.GroupBox2.Controls.Add(Me.Run2)
Me.GroupBox2.Controls.Add(Me.Label4)
Me.GroupBox2.Location = New System.Drawing.Point(16, 251)
Me.GroupBox2.Name = "GroupBox2"
Me.GroupBox2.Size = New System.Drawing.Size(304, 200)
Me.GroupBox2.TabIndex = 15
Me.GroupBox2.TabStop = False
Me.GroupBox2.Text = "Motor 2"
'
'Stop2
'
Me.Stop2.Location = New System.Drawing.Point(48, 160)
Me.Stop2.Name = "Stop2"
Me.Stop2.Size = New System.Drawing.Size(75, 23)
Me.Stop2.TabIndex = 13
Me.Stop2.Text = "Ndalo"
'
'M2Output3
'
Me.M2Output3.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M2Output3.Location = New System.Drawing.Point(184, 168)
Me.M2Output3.Name = "M2Output3"
Me.M2Output3.Size = New System.Drawing.Size(72, 24)
Me.M2Output3.TabIndex = 12
Me.M2Output3.Text = "Output 3"
```

```
'
'M2Output2
'
Me.M2Output2.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M2Output2.Location = New System.Drawing.Point(184, 144)
Me.M2Output2.Name = "M2Output2"
Me.M2Output2.Size = New System.Drawing.Size(72, 24)
Me.M2Output2.TabIndex = 11
Me.M2Output2.Text = "Output 2"
'
'M2Output1
'
Me.M2Output1.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M2Output1.Location = New System.Drawing.Point(184, 120)
Me.M2Output1.Name = "M2Output1"
Me.M2Output1.Size = New System.Drawing.Size(72, 24)
Me.M2Output1.TabIndex = 10
Me.M2Output1.Text = "Output 1"
'
'M2Reverse
'
Me.M2Reverse.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M2Reverse.Location = New System.Drawing.Point(98, 88)
Me.M2Reverse.Name = "M2Reverse"
Me.M2Reverse.Size = New System.Drawing.Size(118, 24)
Me.M2Reverse.TabIndex = 0
Me.M2Reverse.Text = "Rootullim mbrapsh"
'
'Run2
'
Me.Run2.Location = New System.Drawing.Point(48, 120)
Me.Run2.Name = "Run2"
Me.Run2.Size = New System.Drawing.Size(75, 23)
Me.Run2.TabIndex = 9
Me.Run2.Text = "Fillo"
'
'M1StepMode
'
Me.M1StepMode.AllowDrop = True
Me.M1StepMode.Items.AddRange(New Object() {"Wave Step", "Full Step"})
Me.M1StepMode.Location = New System.Drawing.Point(440, 75)
Me.M1StepMode.Name = "M1StepMode"
Me.M1StepMode.Size = New System.Drawing.Size(80, 21)
Me.M1StepMode.TabIndex = 16
Me.M1StepMode.Tag = ""
'
'M2StepMode
'
Me.M2StepMode.AllowDrop = True
Me.M2StepMode.Items.AddRange(New Object() {"Wave Step", "Full Step"})
Me.M2StepMode.Location = New System.Drawing.Point(440, 107)
Me.M2StepMode.Name = "M2StepMode"
Me.M2StepMode.Size = New System.Drawing.Size(80, 21)
Me.M2StepMode.TabIndex = 18
Me.M2StepMode.Tag = ""
'
'Label5
'
Me.Label5.Location = New System.Drawing.Point(24, 35)
Me.Label5.Name = "Label5"
Me.Label5.Size = New System.Drawing.Size(48, 16)
Me.Label5.TabIndex = 19
Me.Label5.Text = "Motor 1"
Me.Label5.TextAlign = System.Drawing.ContentAlignment.TopRight
'
```

```
'Label6
'
Me.Label6.Location = New System.Drawing.Point(18, 67)
Me.Label6.Name = "Label6"
Me.Label6.Size = New System.Drawing.Size(56, 16)
Me.Label6.TabIndex = 20
Me.Label6.Text = "Motor 2"
Me.Label6.TextAlign = System.Drawing.ContentAlignment.TopRight
'
'SetMode
'
Me.SetMode.Location = New System.Drawing.Point(549, 86)
Me.SetMode.Name = "SetMode"
Me.SetMode.Size = New System.Drawing.Size(99, 23)
Me.SetMode.TabIndex = 14
Me.SetMode.Text = "Fikeso menyren"
'
'GroupBox3
'
Me.GroupBox3.Controls.Add(Me.Label5)
Me.GroupBox3.Controls.Add(Me.Label6)
Me.GroupBox3.Location = New System.Drawing.Point(360, 43)
Me.GroupBox3.Name = "GroupBox3"
Me.GroupBox3.Size = New System.Drawing.Size(304, 112)
Me.GroupBox3.TabIndex = 21
Me.GroupBox3.TabStop = False
Me.GroupBox3.Text = "Menyra e punimit te motoreve  bazuar ne HAPI"
'
'M1StepsLeft
'
Me.M1StepsLeft.Location = New System.Drawing.Point(552, 219)
Me.M1StepsLeft.Name = "M1StepsLeft"
Me.M1StepsLeft.Size = New System.Drawing.Size(64, 20)
Me.M1StepsLeft.TabIndex = 23
Me.M1StepsLeft.Text = "1"
'
'M2StepsLeft
'
Me.M2StepsLeft.Location = New System.Drawing.Point(552, 283)
Me.M2StepsLeft.Name = "M2StepsLeft"
Me.M2StepsLeft.Size = New System.Drawing.Size(64, 20)
Me.M2StepsLeft.TabIndex = 24
Me.M2StepsLeft.Text = "1"
'
'M1Active
'
Me.M1Active.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M1Active.Location = New System.Drawing.Point(400, 219)
Me.M1Active.Name = "M1Active"
Me.M1Active.Size = New System.Drawing.Size(56, 24)
Me.M1Active.TabIndex = 14
Me.M1Active.Text = "Active"
'
'M2Active
'
Me.M2Active.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.M2Active.Location = New System.Drawing.Point(400, 283)
Me.M2Active.Name = "M2Active"
Me.M2Active.Size = New System.Drawing.Size(56, 24)
Me.M2Active.TabIndex = 25
Me.M2Active.Text = "Active"
'
'Label7
'
Me.Label7.Location = New System.Drawing.Point(488, 227)
```

```
Me.Label7.Name = "Label7"
Me.Label7.Size = New System.Drawing.Size(56, 16)
Me.Label7.TabIndex = 26
Me.Label7.Text = "Steps Left"
Me.Label7.TextAlign = System.Drawing.ContentAlignment.TopRight
'
'Label8
'
Me.Label8.Location = New System.Drawing.Point(488, 291)
Me.Label8.Name = "Label8"
Me.Label8.Size = New System.Drawing.Size(56, 16)
Me.Label8.TabIndex = 27
Me.Label8.Text = "Steps Left"
Me.Label8.TextAlign = System.Drawing.ContentAlignment.TopRight
'
'GroupBox4
'
Me.GroupBox4.Location = New System.Drawing.Point(384, 203)
Me.GroupBox4.Name = "GroupBox4"
Me.GroupBox4.Size = New System.Drawing.Size(264, 48)
Me.GroupBox4.TabIndex = 28
Me.GroupBox4.TabStop = False
Me.GroupBox4.Text = "Motor 1 Status"
'
'GroupBox5
'
Me.GroupBox5.Location = New System.Drawing.Point(384, 267)
Me.GroupBox5.Name = "GroupBox5"
Me.GroupBox5.Size = New System.Drawing.Size(264, 48)
Me.GroupBox5.TabIndex = 29
Me.GroupBox5.TabStop = False
Me.GroupBox5.Text = "Motor 2 Status"
'
'Input1
'
Me.Input1.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.Input1.Location = New System.Drawing.Point(464, 363)
Me.Input1.Name = "Input1"
Me.Input1.Size = New System.Drawing.Size(16, 24)
Me.Input1.TabIndex = 30
'
'Input2
'
Me.Input2.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.Input2.Location = New System.Drawing.Point(496, 363)
Me.Input2.Name = "Input2"
Me.Input2.Size = New System.Drawing.Size(16, 24)
Me.Input2.TabIndex = 31
'
'Input3
'
Me.Input3.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.Input3.Location = New System.Drawing.Point(528, 363)
Me.Input3.Name = "Input3"
Me.Input3.Size = New System.Drawing.Size(16, 24)
Me.Input3.TabIndex = 32
'
'Input4
'
Me.Input4.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.Input4.Location = New System.Drawing.Point(560, 363)
Me.Input4.Name = "Input4"
Me.Input4.Size = New System.Drawing.Size(16, 24)
Me.Input4.TabIndex = 33
'
```

```
'Input5
'
Me.Input5.CheckAlign = System.Drawing.ContentAlignment.MiddleRight
Me.Input5.Location = New System.Drawing.Point(592, 363)
Me.Input5.Name = "Input5"
Me.Input5.Size = New System.Drawing.Size(16, 24)
Me.Input5.TabIndex = 34
'
'Label9
'
Me.Label9.Location = New System.Drawing.Point(464, 347)
Me.Label9.Name = "Label9"
Me.Label9.Size = New System.Drawing.Size(8, 16)
Me.Label9.TabIndex = 35
Me.Label9.Text = "1"
'
'Label10
'
Me.Label10.Location = New System.Drawing.Point(496, 347)
Me.Label10.Name = "Label10"
Me.Label10.Size = New System.Drawing.Size(8, 16)
Me.Label10.TabIndex = 36
Me.Label10.Text = "2"
'
'Label11
'
Me.Label11.Location = New System.Drawing.Point(528, 347)
Me.Label11.Name = "Label11"
Me.Label11.Size = New System.Drawing.Size(8, 16)
Me.Label11.TabIndex = 37
Me.Label11.Text = "3"
'
'Label12
'
Me.Label12.Location = New System.Drawing.Point(560, 347)
Me.Label12.Name = "Label12"
Me.Label12.Size = New System.Drawing.Size(8, 16)
Me.Label12.TabIndex = 38
Me.Label12.Text = "4"
'
'Label13
'
Me.Label13.Location = New System.Drawing.Point(592, 347)
Me.Label13.Name = "Label13"
Me.Label13.Size = New System.Drawing.Size(8, 16)
Me.Label13.TabIndex = 39
Me.Label13.Text = "5"
'
'Label14
'
Me.Label14.Location = New System.Drawing.Point(32, 37)
Me.Label14.Name = "Label14"
Me.Label14.Size = New System.Drawing.Size(40, 16)
Me.Label14.TabIndex = 40
Me.Label14.Text = "Inputs"
'
'GroupBox6
'
Me.GroupBox6.Controls.Add(Me.Label14)
Me.GroupBox6.Location = New System.Drawing.Point(384, 331)
Me.GroupBox6.Name = "GroupBox6"
Me.GroupBox6.Size = New System.Drawing.Size(264, 64)
Me.GroupBox6.TabIndex = 30
Me.GroupBox6.TabStop = False
Me.GroupBox6.Text = "Input Status"
```

```
'
'GetMotorStatus
'
Me.GetMotorStatus.Location = New System.Drawing.Point(400, 411)
Me.GetMotorStatus.Name = "GetMotorStatus"
Me.GetMotorStatus.Size = New System.Drawing.Size(235, 23)
Me.GetMotorStatus.TabIndex = 41
Me.GetMotorStatus.Text = "Shfaq gjendjet e motorit"
'
'GroupBox7
'
Me.GroupBox7.Location = New System.Drawing.Point(360, 171)
Me.GroupBox7.Name = "GroupBox7"
Me.GroupBox7.Size = New System.Drawing.Size(304, 280)
Me.GroupBox7.TabIndex = 42
Me.GroupBox7.TabStop = False
Me.GroupBox7.Text = "Status"
'
'Label15
'
Me.Label15.Location = New System.Drawing.Point(13, 9)
Me.Label15.Name = "Label15"
Me.Label15.Size = New System.Drawing.Size(336, 23)
Me.Label15.TabIndex = 43
Me.Label15.Text = "Inicializimi kryhet perpara se te kryhen veprime me butonat ne  GUI"
'
'Form1
'
Me.AutoScaleBaseSize = New System.Drawing.Size(5, 13)
Me.ClientSize = New System.Drawing.Size(671, 461)
Me.Controls.Add(Me.Label15)
Me.Controls.Add(Me.GetMotorStatus)
Me.Controls.Add(Me.Label13)
Me.Controls.Add(Me.Label12)
Me.Controls.Add(Me.Label11)
Me.Controls.Add(Me.Label10)
Me.Controls.Add(Me.Label9)
Me.Controls.Add(Me.Input5)
Me.Controls.Add(Me.Input4)
Me.Controls.Add(Me.Input3)
Me.Controls.Add(Me.Input2)
Me.Controls.Add(Me.Input1)
Me.Controls.Add(Me.Label8)
Me.Controls.Add(Me.Label7)
Me.Controls.Add(Me.M2Active)
Me.Controls.Add(Me.M2StepsLeft)
Me.Controls.Add(Me.M1StepsLeft)
Me.Controls.Add(Me.M2StepMode)
Me.Controls.Add(Me.M1StepMode)
Me.Controls.Add(Me.Label3)
Me.Controls.Add(Me.M2Interval)
Me.Controls.Add(Me.M2Steps)
Me.Controls.Add(Me.GroupBox2)
Me.Controls.Add(Me.Label2)
Me.Controls.Add(Me.M1Interval)
Me.Controls.Add(Me.Label1)
Me.Controls.Add(Me.M1Steps)
Me.Controls.Add(Me.Button1)
Me.Controls.Add(Me.GroupBox1)
Me.Controls.Add(Me.SetMode)
Me.Controls.Add(Me.GroupBox3)
Me.Controls.Add(Me.M1Active)
Me.Controls.Add(Me.GroupBox4)
Me.Controls.Add(Me.GroupBox5)
Me.Controls.Add(Me.GroupBox6)
```

```
        Me.Controls.Add(Me.GroupBox7)
        Me.Name = "Form1"
        Me.Text = "Kontrolli i motorave nepermjet Stepper Bee"
        Me.GroupBox1.ResumeLayout(False)
        Me.GroupBox2.ResumeLayout(False)
        Me.GroupBox3.ResumeLayout(False)
        Me.GroupBox6.ResumeLayout(False)
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub

#End Region

    Declare Function InitStp Lib "stp.dll" () As Integer
    Declare Function RunMotor1 Lib "stp.dll" (ByVal steps As Integer, ByVal interval As Integer,
ByVal direction As Integer, ByVal outputs As Integer) As Boolean
    Declare Function RunMotor2 Lib "stp.dll" (ByVal steps As Integer, ByVal interval As Integer,
ByVal direction As Integer, ByVal outputs As Integer) As Boolean
    Declare Function StopMotor1 Lib "stp.dll" (ByVal outputs As Integer) As Boolean
    Declare Function StopMotor2 Lib "stp.dll" (ByVal outputs As Integer) As Boolean
    Declare Function SetStepMode Lib "stp.dll" (ByVal M1Mode As Integer, ByVal M2Mode As Integer) As
Boolean
    Declare Function GetCurrentStatus Lib "stp.dll" (ByRef M1Active As Integer, ByRef M2Active As
Integer, ByRef M1Steps As Integer, ByRef M2Steps As Integer, ByRef Inputs As Integer) As Boolean

    'inicializimi i StepperBee. Ky modul thirret ne fillim.
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        InitStp()
    End Sub

    ' Fillim i levizjes - Motor 1
    Private Sub Run1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Run1.Click
        Dim steps As Integer
        Dim interval As Integer
        Dim direction As Integer
        Dim outputs As Integer

        steps = M1Steps.Text
        interval = M1Interval.Text

        ' Levizja e motorit para ose mbrapa
        If M1Reverse.Checked Then
            direction = 1
        Else
            direction = 0
        End If

        outputs = 0
        If M1Output1.Checked Then
            outputs = outputs Or 1
        End If

        If M1Output2.Checked Then
            outputs = outputs Or 2
        End If

        If M1Output3.Checked Then
            outputs = outputs Or 4
        End If

        ' Startim i motorit 1
        RunMotor1(steps, interval, direction, outputs)
```

```
    End Sub

    ' stopim motori 1 dhe percaktim daljesh njekohesisht
    Private Sub Stop1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Stop1.Click
        Dim outputs As Integer

        outputs = 0
        If M1Output1.Checked Then
            outputs = outputs Or 1
        End If

        If M1Output2.Checked Then
            outputs = outputs Or 2
        End If

        If M1Output3.Checked Then
            outputs = outputs Or 4
        End If

        StopMotor1(outputs)

    End Sub

    ' Fillim i levizjes - Motor 2
    Private Sub Run2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Run2.Click
        Dim steps As Integer
        Dim interval As Integer
        Dim direction As Integer
        Dim outputs As Integer

        steps = M2Steps.Text
        interval = M2Interval.Text

        ' Levizja e motorit para ose mbrapa
        If M2Reverse.Checked Then
            direction = 1
        Else
            direction = 0
        End If

        outputs = 0
        If M2Output1.Checked Then
            outputs = outputs Or 1
        End If

        If M2Output2.Checked Then
            outputs = outputs Or 2
        End If

        If M2Output3.Checked Then
            outputs = outputs Or 4
        End If

        ' Startim i motorit 2
        RunMotor2(steps, interval, direction, outputs)

    End Sub

    ' stopim motori 2 dhe percaktim daljesh njekohesisht
    Private Sub Stop2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Stop2.Click
        Dim outputs As Integer
```

```
        ' the outputs are specified based on the tick boxes as in "Run2_Click" above
        outputs = 0
        If M2Output1.Checked Then
            outputs = outputs Or 1
        End If

        If M2Output2.Checked Then
            outputs = outputs Or 2
        End If

        If M2Output3.Checked Then
            outputs = outputs Or 4
        End If

        StopMotor2(outputs)
    End Sub

    ' Specifikime per hapin e pune se motorave
    Private Sub SetMode_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
SetMode.Click
        Dim M1Mode, M2Mode As Integer

        ' Hapi motori 1
        If (M1StepMode.SelectedIndex = 1) Then
            M1Mode = 1                              ' hap i plote i zgjedhur
        Else
            M1Mode = 0                              ' hap jo i plote i zgjedhur
        End If

        ' Hapi motori 2
        If (M2StepMode.SelectedIndex = 1) Then
            M2Mode = 1                              ' hap i plote i zgjedhur
        Else
            M2Mode = 0                              ' hap jo i plote i zgjedhur
        End If

        ' Fikeso moden e punes per motorat lidhur me hapin e tyre
        SetStepMode(M1Mode, M2Mode)
    End Sub

    ' Moduli merr informacion mbi gjendjen e motorave
    Private Sub GetMotorStatus_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles GetMotorStatus.Click
        Dim Motor1Active, Motor2Active, Motor1StepsLeft, Motor2StepsLeft, DigitalInputs As Integer

        ' merr gjendjen e motorave nepermejt funksionit GetCurrentStatus
        GetCurrentStatus(Motor1Active, Motor2Active, Motor1StepsLeft, Motor2StepsLeft,
DigitalInputs)

        ' Gjendja e motorit 1
        If Motor1Active = 1 Then
            M1Active.Checked = True
        Else
            M1Active.Checked = False
        End If

        ' Gjendja e motorit 2
        If Motor2Active = 1 Then
            M2Active.Checked = True
        Else
            M2Active.Checked = False
        End If

        M1StepsLeft.Text = Motor1StepsLeft
```

```
        M2StepsLeft.Text = Motor2StepsLeft

        ' bits 0 to 4 of the DigitalInputs parameter correspond to the current
        ' state of the digital inputs, so check the appropriate bit for each
        ' input and tick it's screen box if on (1 = on, logic high, +5v)
        If (DigitalInputs And 1) Then
            Input1.Checked = True
        Else
            Input1.Checked = False
        End If

        If (DigitalInputs And 2) Then
            Input2.Checked = True
        Else
            Input2.Checked = False
        End If

        If (DigitalInputs And 4) Then
            Input3.Checked = True
        Else
            Input3.Checked = False
        End If

        If (DigitalInputs And 8) Then
            Input4.Checked = True
        Else
            Input4.Checked = False
        End If

        If (DigitalInputs And 16) Then
            Input5.Checked = True
        Else
            Input5.Checked = False
        End If

    End Sub


End Class
```

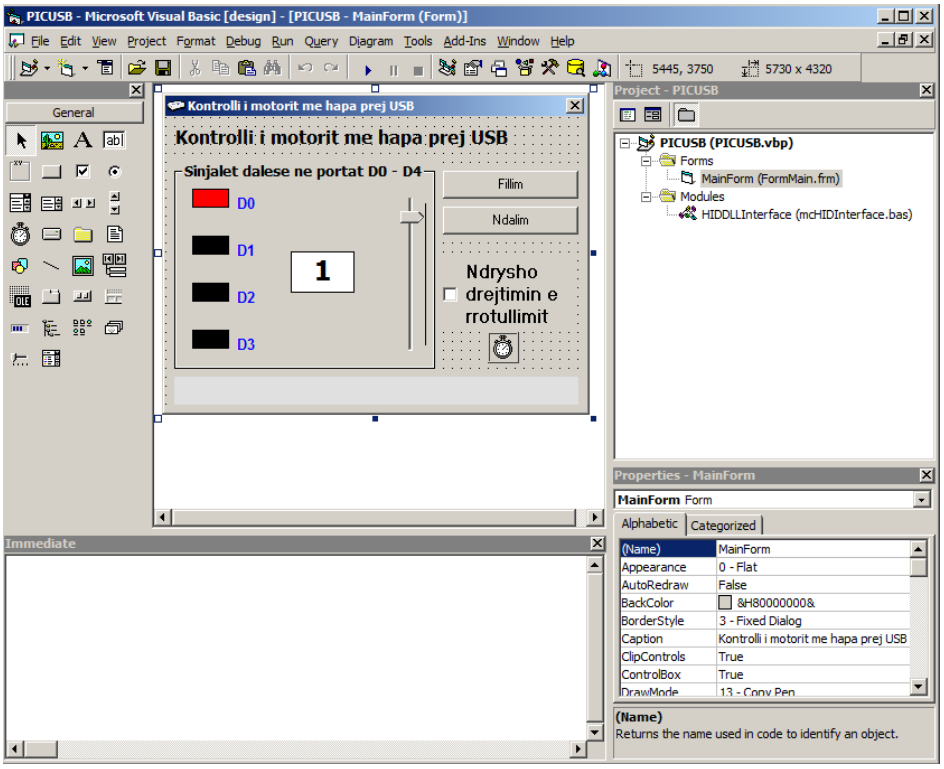
Në vazhdim po japim programin e ndërtuar për PIC18F4550-I/P në komandimit e lëvizjes së motorit me hapa në MicroCode Studio – PicBasic Pro V 3.0.7.0.

```
'*****************************************************************
'*  Name    : STEPUSB.BAS                                        *
'*  Author  : G.Progri                                           *
'*  Notice  : Copyright (c) 2015                                 *
'*  Date    : 15.11.2015                                         *
'*****************************************************************
' select MCU and clock speed
Device = 18F4550
XTAL = 48
Declare PORTB_PULLUPS 1
' descriptor file, located in \inc\usb_18 - a copy
' is located in the same folder as this file
USB_DESCRIPTOR = "STEPUSBDESC.inc"

' USB Buffer...
Symbol USBBufferSizeMax = 8
Symbol USBBufferSizeTX  = 8
Symbol USBBufferSizeRX  = 8
```

```
Dim    USBBuffer[USBBufferSizeMax] As Byte

' some useful flags...
Dim PP0 As Byte SYSTEM        ' USBPOLL status return
Symbol CARRY_FLAG = STATUS.0  ' high if microcontroller does not have control over the buffer
Symbol ATTACHED_STATE = 6     ' is USB attached

' ************************************************************
' * main program loop - remember, you must keep the USB      *
' * connection alive with a call to USBPoll, USBIn or USBOut *
' * every couple of milliseconds or so                       *
' ************************************************************
TRISD = %00000000

PORTD = %00000001
GoSub MakeSleep
PORTD = %00000011
GoSub MakeSleep
PORTD = %00000111
GoSub MakeSleep
PORTD = %00001111
GoSub MakeSleep
PORTD = %00000111
GoSub MakeSleep
PORTD = %00000011
GoSub MakeSleep
PORTD = %00000001
GoSub MakeSleep


GoSub AttachToUSB
ProgramLoop:
   USBIn 1, USBBuffer, USBBufferSizeRX, ProgramLoop
   PORTD = USBBuffer[0]
   GoTo ProgramLoop

' ************************************************************
' * receive data from the USB bus                            *
' ************************************************************
DoUSBIn:
   USBIn 1, USBBuffer, USBBufferSizeRX, DoUSBIn
   Return

' ************************************************************
' * transmit data                                            *
' ************************************************************
DoUSBOut:
   USBOut 1, USBBuffer, USBBufferSizeTX, DoUSBOut
   Return

' ************************************************************
' * wait for USB interface to attach                         *
' ************************************************************
AttachToUSB:
   Repeat
      USBPoll
   Until PP0 = ATTACHED_STATE
   Return


' ************************************************************
' * wait after command                                       *
' ************************************************************
MakeSleep:
 For i=0 To 255 Step 1
```

```
    For j=0 To 255 Step 1
        For k=0 To 10 Step 1
        Next
    Next
 Next
Return
```

Aplikacionin GUI i ndërtuar në Microsoft Visual Basic 6.0 prej të cilit dërgohen komandat për rrotullimin e motorit me hapa para dhe prapa.



MainForm.frm

```
' vendor and product IDs
Private Const VendorID = 6030
Private Const ProductID = 2009

' read and write buffers
Private Const BufferInSize = 8
Private Const BufferOutSize = 8

Dim BufferIn(0 To BufferInSize)         As Byte
Dim BufferOut(0 To BufferOutSize)       As Byte

Dim say                 As Integer
Dim tur                 As Integer
Dim boolAutoRun         As Boolean
Dim Intgen              As Integer
Dim hiz                 As Integer
```

```
Private Sub Command1_Click()
    BufferOut(1) = Val(Text1.Text)
    hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub

Private Sub Command2_Click()
    Intgen = 1
    boolAutoRun = False
    TimerSpeed.Enabled = True
End Sub

Private Sub Command3_Click()
    TimerSpeed.Enabled = False
End Sub

' ****************************************************************
' when the form loads, connect to the HID controller - pass
' the form window handle so that you can receive notification
' events...
'****************************************************************
Private Sub Form_Load()
    App.Title = Me.Caption

   ' do not remove!
   ConnectToHID (Me.hwnd)
End Sub

'****************************************************************
' disconnect from the HID controller...
'****************************************************************
Private Sub Form_Unload(Cancel As Integer)
    BufferOut(1) = 0
    hidWriteEx VendorID, ProductID, BufferOut(0)
    DisconnectFromHID
End Sub

'****************************************************************
' a HID device has been plugged in...
'****************************************************************
Public Sub OnPlugged(ByVal pHandle As Long)
   If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) = ProductID Then
      ' ** YOUR CODE HERE **
      Label2.Caption = "Komunikimi me PIC18F4550 USB..."
      Label2.ForeColor = &HC00000

      'Image1.Visible = True
      Frame1.Enabled = True
      'Frame2.Enabled = True
      'Frame4.Enabled = True

   End If
End Sub

'****************************************************************
' a HID device has been unplugged...
'****************************************************************
Public Sub OnUnplugged(ByVal pHandle As Long)
   If hidGetVendorID(pHandle) = VendorID And hidGetProductID(pHandle) = ProductID Then
      ' ** YOUR CODE HERE **
      Label2.Caption = "Nderpreje lidhjeje me PIC18F4550 USB..."
      Label2.ForeColor = &HFF&
      'Image1.Visible = False
      Frame1.Enabled = False
      'Frame2.Enabled = False
      'Frame4.Enabled = False
```

```
      TimerSpeed.Enabled = False

   End If
End Sub

'*********************************************************************
' controller changed notification - called
' after ALL HID devices are plugged or unplugged
'*********************************************************************
Public Sub OnChanged()
   Dim DeviceHandle As Long

   ' get the handle of the device we are interested in, then set
   ' its read notify flag to true - this ensures you get a read
   ' notification message when there is some data to read...
   DeviceHandle = hidGetHandle(VendorID, ProductID)
   hidSetReadNotify DeviceHandle, True
End Sub

'*********************************************************************
' on read event...
'*********************************************************************
Public Sub OnRead(ByVal pHandle As Long)

   ' read the data (don't forget, pass the whole array)...
   If hidRead(pHandle, BufferIn(0)) Then
      ' ** YOUR CODE HERE **
      ' first byte is the report ID, e.g. BufferIn(0)
      ' the other bytes are the data from the microcontrolller...
      'Label3.Caption = BufferIn(1)
      'Label4.Caption = BufferIn(2)
      'ProgressBar1.Value = BufferIn(1) + 1
      'ProgressBar2.Value = BufferIn(2) + 1
      'Label5.Caption = BufferIn(3)
   End If
End Sub

'*********************************************************************
' this is how you write some data...
'*********************************************************************
Public Sub WriteSomeData()
   BufferOut(0) = 0   ' first by is always the report ID
   BufferOut(1) = 10  ' first data item, etc etc

   ' write the data (don't forget, pass the whole array)...
   hidWriteEx VendorID, ProductID, BufferOut(0)
End Sub

Private Sub SliderTimerInterval_Change()
   TimerSpeed.Interval = SliderTimerInterval.Value
End Sub

Private Sub Text1_Change()
    If Val(Text1.Text) > 255 Then Text1.Text = ""
    Shape1(1).FillColor = &H0&
    Shape1(2).FillColor = &H0&
    Shape1(4).FillColor = &H0&
    Shape1(8).FillColor = &H0&
    Shape1(Val(Text1.Text)).FillColor = &HFF&
End Sub

Private Sub Text1_GotFocus()
    SeciliTxBox
End Sub
```

```
Sub SeciliTxBox()
    Screen.ActiveControl.SelStart = 0
    Screen.ActiveControl.SelLength = Len(Screen.ActiveControl.Text)
End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then Exit Sub
    If IsNumeric(Chr(KeyAscii)) = False Then KeyAscii = 0
End Sub

Private Sub TimerSpeed_Timer()

    On Error Resume Next

    If (cSignTurn.Value = 1) Then
        Intgen = Intgen * 2
        If (Intgen = 16) Then
            Intgen = 1
        End If
    End If

    If (cSignTurn.Value = 0) Then
        Intgen = Intgen / 2
        If (Intgen = 0) Then
            Intgen = 8
        End If
    End If

    If (boolAutoRun = True) Then
    say = say + 1
        If (say = tur) Then
            TimerSpeed.Enabled = False
            say = 0
        End If
    End If

    Text1.Text = Str(Intgen)

    BufferOut(1) = Val(Text1.Text)

    hidWriteEx VendorID, ProductID, BufferOut(0)

    Err.Clear
End Sub
```

Moduli HIDDLLInterface.bas

```
' this is the interface to the HID controller DLL - you should not
' normally need to change anything in this file.
'
' WinProc() calls your main form 'event' procedures - these are currently
' set to..
'
' MainForm.OnPlugged(ByVal pHandle as long)
' MainForm.OnUnplugged(ByVal pHandle as long)
' MainForm.OnChanged()
' MainForm.OnRead(ByVal pHandle as long)

Option Explicit

' HID interface API declarations...
Declare Function hidConnect Lib "mcHID.dll" Alias "Connect" (ByVal pHostWin As Long) As Boolean
```

```
Declare Function hidDisconnect Lib "mcHID.dll" Alias "Disconnect" () As Boolean
Declare Function hidGetItem Lib "mcHID.dll" Alias "GetItem" (ByVal pIndex As Long) As Long
Declare Function hidGetItemCount Lib "mcHID.dll" Alias "GetItemCount" () As Long
Declare Function hidRead Lib "mcHID.dll" Alias "Read" (ByVal pHandle As Long, ByRef pData As Byte)
As Boolean
Declare Function hidWrite Lib "mcHID.dll" Alias "Write" (ByVal pHandle As Long, ByRef pData As Byte)
As Boolean
Declare Function hidReadEx Lib "mcHID.dll" Alias "ReadEx" (ByVal pVendorID As Long, ByVal pProductID
As Long, ByRef pData As Byte) As Boolean
Declare Function hidWriteEx Lib "mcHID.dll" Alias "WriteEx" (ByVal pVendorID As Long, ByVal
pProductID As Long, ByRef pData As Byte) As Boolean
Declare Function hidGetHandle Lib "mcHID.dll" Alias "GetHandle" (ByVal pVendoID As Long, ByVal
pProductID As Long) As Long
Declare Function hidGetVendorID Lib "mcHID.dll" Alias "GetVendorID" (ByVal pHandle As Long) As Long
Declare Function hidGetProductID Lib "mcHID.dll" Alias "GetProductID" (ByVal pHandle As Long) As
Long
Declare Function hidGetVersion Lib "mcHID.dll" Alias "GetVersion" (ByVal pHandle As Long) As Long
Declare Function hidGetVendorName Lib "mcHID.dll" Alias "GetVendorName" (ByVal pHandle As Long,
ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetProductName Lib "mcHID.dll" Alias "GetProductName" (ByVal pHandle As Long,
ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetSerialNumber Lib "mcHID.dll" Alias "GetSerialNumber" (ByVal pHandle As Long,
ByVal pText As String, ByVal pLen As Long) As Long
Declare Function hidGetInputReportLength Lib "mcHID.dll" Alias "GetInputReportLength" (ByVal pHandle
As Long) As Long
Declare Function hidGetOutputReportLength Lib "mcHID.dll" Alias "GetOutputReportLength" (ByVal
pHandle As Long) As Long
Declare Sub hidSetReadNotify Lib "mcHID.dll" Alias "SetReadNotify" (ByVal pHandle As Long, ByVal
pValue As Boolean)
Declare Function hidIsReadNotifyEnabled Lib "mcHID.dll" Alias "IsReadNotifyEnabled" (ByVal pHandle
As Long) As Boolean
Declare Function hidIsAvailable Lib "mcHID.dll" Alias "IsAvailable" (ByVal pVendorID As Long, ByVal
pProductID As Long) As Boolean

' windows API declarations - used to set up messaging...
Private Declare Function CallWindowProc Lib "user32" Alias "CallWindowProcA" (ByVal lpPrevWndFunc As
Long, ByVal hwnd As Long, ByVal Msg As Long, ByVal wParam As Long, ByVal lParam As Long) As Long
Private Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (ByVal hwnd As Long,
ByVal nIndex As Long, ByVal dwNewLong As Long) As Long

' windows API Constants
Private Const WM_APP = 32768
Private Const GWL_WNDPROC = -4

' HID message constants
Private Const WM_HID_EVENT = WM_APP + 200
Private Const NOTIFY_PLUGGED = 1
Private Const NOTIFY_UNPLUGGED = 2
Private Const NOTIFY_CHANGED = 3
Private Const NOTIFY_READ = 4

' local variables
Private FPrevWinProc As Long    ' Handle to previous window procedure
Private FWinHandle As Long      ' Handle to message window

' Set up a windows hook to receive notification
' messages from the HID controller DLL - then connect
' to the controller
Public Function ConnectToHID(ByVal pHostWin As Long) As Boolean
   FWinHandle = pHostWin
   ConnectToHID = hidConnect(FWinHandle)
   FPrevWinProc = SetWindowLong(FWinHandle, GWL_WNDPROC, AddressOf WinProc)
End Function

' Unhook from the HID controller and disconnect...
```

```
Public Function DisconnectFromHID() As Boolean
   DisconnectFromHID = hidDisconnect
   SetWindowLong FWinHandle, GWL_WNDPROC, FPrevWinProc
End Function

' This is the procedure that intercepts the HID controller messages...
Private Function WinProc(ByVal pHWnd As Long, ByVal pMsg As Long, ByVal wParam As Long, ByVal lParam
As Long) As Long
   If pMsg = WM_HID_EVENT Then
       Select Case wParam

        ' HID device has been plugged message...
        Case Is = NOTIFY_PLUGGED
           MainForm.OnPlugged (lParam)

        ' HID device has been unplugged
        Case Is = NOTIFY_UNPLUGGED
           MainForm.OnUnplugged (lParam)

        ' controller has changed...
        Case Is = NOTIFY_CHANGED
           MainForm.OnChanged

        ' read event...
        Case Is = NOTIFY_READ
           MainForm.OnRead (lParam)
        End Select

   End If

   ' next...
   WinProc = CallWindowProc(FPrevWinProc, pHWnd, pMsg, wParam, lParam)

End Function
```

Shtojca 4

Programi (GUI) në kompjuter në gjuhën Visual Basic 6.0 për dërgimin e komandave (për rrotullimin e motorit dhe ndryshimin e shpejtesise se tij) në microcontroller.

```
VERSION 5.00
Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"
Begin VB.Form frmMotorWireless
   Caption         =   "Komandimi i shpejtesise dhe drejtimi i levizjes"
   ClientHeight    =   6660
   ClientLeft      =   60
   ClientTop       =   450
   ClientWidth     =   9060
   LinkTopic       =   "Form1"
   ScaleHeight     =   6660
   ScaleWidth      =   9060
   StartUpPosition =   2  'CenterScreen
   Begin VB.CommandButton cmdReset
      Caption         =   "Reset ALL"
      BeginProperty Font
         Name            =   "Arial Narrow"
         Size            =   14.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   615
      Left            =   120
      MaskColor       =   &H00FFFFFF&
      Style           =   1  'Graphical
      TabIndex        =   10
      Top             =   5880
      Width           =   2175
   End
   Begin VB.Frame Frame3
      Caption         =   "Komandimi i shpejtesise"
      BeginProperty Font
         Name            =   "Arial Narrow"
         Size            =   14.25
         Charset         =   0
         Weight          =   700
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      ForeColor       =   &H00000080&
      Height          =   1575
      Left            =   120
      TabIndex        =   8
      Top             =   2280
      Width           =   8775
      Begin VB.CommandButton cmdSpeedPlus
         Caption         =   "Shpejtesia +"
         BeginProperty Font
            Name            =   "Arial Narrow"
            Size            =   14.25
            Charset         =   0
            Weight          =   700
            Underline       =   0   'False
            Italic          =   0   'False
            Strikethrough   =   0   'False
         EndProperty
```

```
    Height          =   615
    Left            =   840
    Style           =   1  'Graphical
    TabIndex        =   12
    Top             =   600
    Width           =   2655
  End
  Begin VB.CommandButton cmdSpeedMinus
    Caption         =   "Shpejtesia -"
    BeginProperty Font
      Name            =   "Arial Narrow"
      Size            =   14.25
      Charset         =   0
      Weight          =   700
      Underline       =   0   'False
      Italic          =   0   'False
      Strikethrough   =   0   'False
    EndProperty
    Height          =   615
    Left            =   5160
    Style           =   1  'Graphical
    TabIndex        =   11
    Top             =   600
    Width           =   2655
  End
End
Begin VB.CommandButton cmdExit
  Caption         =   "DALJE"
  BeginProperty Font
    Name            =   "Arial Narrow"
    Size            =   14.25
    Charset         =   0
    Weight          =   700
    Underline       =   0   'False
    Italic          =   0   'False
    Strikethrough   =   0   'False
  EndProperty
  Height          =   615
  Left            =   6720
  MaskColor       =   &H000080FF&
  Style           =   1  'Graphical
  TabIndex        =   7
  Top             =   5880
  UseMaskColor    =   -1  'True
  Width           =   2175
End
Begin VB.Frame Frame2
  Caption         =   "Zgjedhja e portes dhe inicializime"
  BeginProperty Font
    Name            =   "Arial Narrow"
    Size            =   14.25
    Charset         =   0
    Weight          =   700
    Underline       =   0   'False
    Italic          =   0   'False
    Strikethrough   =   0   'False
  EndProperty
  ForeColor       =   &H00000080&
  Height          =   1455
  Left            =   120
  TabIndex        =   0
  Top             =   720
  Width           =   8775
  Begin VB.CommandButton cmdConnect
    Caption         =   "Connect"
    BeginProperty Font
```

```
      Name           =   "Arial Narrow"
      Size           =   14.25
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    Height         =   615
    Left           =   6840
    TabIndex       =   6
    Top            =   600
    Width          =   1695
  End
  Begin VB.ComboBox cmbBoundRate
    BeginProperty Font
      Name           =   "Arial Narrow"
      Size           =   12
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    Height         =   420
    Left           =   4920
    Style          =   2  'Dropdown List
    TabIndex       =   5
    Top            =   720
    Width          =   1815
  End
  Begin VB.ComboBox cmbPort
    BeginProperty Font
      Name           =   "Arial Narrow"
      Size           =   12
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    Height         =   420
    Left           =   1800
    Style          =   2  'Dropdown List
    TabIndex       =   4
    Top            =   720
    Width          =   1095
  End
  Begin VB.Label Label5
    AutoSize       =   -1  'True
    BackStyle      =   0  'Transparent
    Caption        =   "Baund rate"
    BeginProperty Font
      Name           =   "Arial Narrow"
      Size           =   13.5
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    ForeColor      =   &H80000008&
    Height         =   330
    Left           =   3600
    TabIndex       =   2
    Top            =   750
```

```
    Width          =   1155
  End
  Begin VB.Label Label4
    AutoSize       =   -1  'True
    BackStyle      =   0  'Transparent
    Caption        =   "Porta seriale"
    BeginProperty Font
      Name           =   "Arial Narrow"
      Size           =   13.5
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    ForeColor      =   &H80000008&
    Height         =   330
    Left           =   240
    TabIndex       =   1
    Top            =   750
    Width          =   1320
  End
End
Begin VB.Frame Frame4
  Caption        =   "Komandimi i drejtimit te levizjes"
  BeginProperty Font
    Name           =   "Arial Narrow"
    Size           =   14.25
    Charset        =   0
    Weight         =   700
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  ForeColor      =   &H00000080&
  Height         =   1695
  Left           =   120
  TabIndex       =   9
  Top            =   3960
  Width          =   8775
  Begin VB.CommandButton cmdAntiClockWise
    Caption        =   "Drejtimi anti orar"
    BeginProperty Font
      Name           =   "Arial Narrow"
      Size           =   14.25
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
    EndProperty
    Height         =   615
    Left           =   5160
    Style          =   1  'Graphical
    TabIndex       =   14
    Top            =   720
    Width          =   2655
  End
  Begin VB.CommandButton cmdClockWise
    Caption        =   "Drejtimi orar"
    BeginProperty Font
      Name           =   "Arial Narrow"
      Size           =   14.25
      Charset        =   0
      Weight         =   700
      Underline      =   0   'False
```

```
          Italic         =   0   'False
          Strikethrough  =   0   'False
        EndProperty
        Height         =   615
        Left           =   840
        Style          =   1  'Graphical
        TabIndex       =   13
        Top            =   720
        Width          =   2655
      End
    End
    Begin MSCommLib.MSComm CommObject
      Left           =   7560
      Top            =   120
      _ExtentX       =   1005
      _ExtentY       =   1005
      _Version       =   393216
      DTREnable      =   -1  'True
    End
    Begin VB.Label Label1
      Caption        =   "Kontrolli i levizjes se motorit me hapa "
      BeginProperty Font
        Name           =   "Impact"
        Size           =   18
        Charset        =   0
        Weight         =   400
        Underline      =   0   'False
        Italic         =   0   'False
        Strikethrough  =   0   'False
      EndProperty
      ForeColor      =   &H00000080&
      Height         =   615
      Left           =   120
      TabIndex       =   3
      Top            =   120
      Width          =   6855
    End
End
Attribute VB_Name = "frmMotorWireless"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private Sub cmdConnect_Click()
    On Error Resume Next
    If cmdConnect.Caption = "Connect" Then
        CommObject.CommPort = VBA.Right(cmbPort.Text, 1)
        CommObject.Settings = cmbBoundRate.Text

        CommObject.PortOpen = True

        If CommObject.PortOpen = True Then
          ' Lidhja e kryer
          DoEnable True
        Else
          ' Lidhja jo e kryer (problem ne hapjen e portes)
          MsgBox "Porta nuk mund te hapet !!!. Error : (" & Err.Number & ")" & Err.Description, vbCritical, App.Title
          DoEnable False
        End If
    Else
        CommObject.PortOpen = False
        DoEnable False
    End If
End Sub
```

```
Private Function DoEnable(sbool As Boolean)
   If sbool = False Then
      cmdConnect.Caption = "Connect"
   Else
      cmdConnect.Caption = "Disconnect"
   End If
   cmdSpeedPlus.Enabled = sbool
   cmdSpeedMinus.Enabled = sbool
   cmdClockWise.Enabled = sbool
   cmdAntiClockWise.Enabled = sbool
End Function

Private Sub cmdSpeedPlus_Click()
   ' Dergim informacioni ne porte
   ' (+) - karakteri per rritje shpejtesie rrotullimi per motorin
   CommObject.Output = "+"
   'CommObject.Output = vbCrLf
End Sub

Private Sub cmdSpeedMinus_Click()
   ' Dergim informacioni ne porte
   ' (-) - karakteri per ulje shpejtesie rrotullimi per motorin
   CommObject.Output = "-"
   'CommObject.Output = vbCrLf
End Sub

Private Sub cmdClockWise_Click()
   ' Dergim informacioni ne porte
   ' (R) - karakteri per rrotullim orar
   CommObject.Output = "R"
   'CommObject.Output = vbCrLf
End Sub

Private Sub cmdExit_Click()
   Unload frmMotorWireless
End Sub

Private Sub cmdAntiClockWise_Click()
   ' Dergim informacioni ne porte
   ' (L) - karakteri per rrotullim anti-orar
   CommObject.Output = "L"
   'CommObject.Output = vbCrLf
End Sub

Private Sub cmdReset_Click()
   ' Dergim informacioni ne porte
   ' (0) - karakteri per resetim gjendjesh
   CommObject.Output = "0"
   'CommObject.Output = vbCrLf
End Sub

Private Sub Form_Load()

   ' Ngarkim informacioni per portat seriale

   cmbPort.Clear
   cmbPort.AddItem "COM1"
   cmbPort.AddItem "COM2"
   cmbPort.AddItem "COM3"
   cmbPort.AddItem "COM4"
   cmbPort.AddItem "COM5"
   cmbPort.AddItem "COM6"
   cmbPort.AddItem "COM7"
   cmbPort.AddItem "COM8"
   cmbPort.AddItem "COM9"
```

```
    cmbPort.ListIndex = 0

    cmbBoundRate.Clear
    cmbBoundRate.AddItem "1200,N,8,1"
    cmbBoundRate.AddItem "2400,N,8,1"
    cmbBoundRate.AddItem "9600,N,8,1"
    cmbBoundRate.ListIndex = 1

    DoEnable False
End Sub
```

Kodet për microprocessor në anën e transmetimit të komandave drejt RF.


```
// Kodet për microprocessor ne anen e transmetimit te komandave
#define TRANSMITTER
#define NETWORK_SIG_SIZE 3
#define VAL_SIZE         2
#define NET_ADDR 5
```

```
// Few bytes used to initiate a transfer
const byte g_network_sig[NETWORK_SIG_SIZE] = {0x8F, 0xAA, NET_ADDR};

// Pini 13 eshte LED ne Arduino board
int led = 13;

void setup() {
// Inicializimi i portes seriale qe komunikon me kompjuterin ne bound rate 9600 bite/s
Serial.begin(9600);

// Hardware supports up to 2400, but 1200 gives longer range
Serial1.begin(600);
}
void loop() {
// lexoj nga porta seriale 0, dergoj data ne porten seriale 1:
if (Serial.available()) {
char inByte = Serial.read();

// Ndertoj protokollin e komunikimit per dergimin e komandave
if(inByte=='l' || inByte=='L') {                      // drejtimi antiorar
        writeUInt(100);
        blink(4);
        }
    else if(inByte=='r' || inByte=='R') {             // drejtimi orar
        writeUInt(101);
        blink(4);
        }
    else if(inByte=='+') {                          // rritje e shpejtesise se rrotullimit
        writeUInt(102);
        blink(6);
        }
    else if(inByte=='-') {                          // ulja e shpejtesise se rrotullimit
        writeUInt(103);
        blink(6);
        }
    else if(inByte=='0'){              // Reset gjendjet
        writeUInt(104);
        blink(8);
        }
    }
}

// Blink the LED Pin 13
void blink(int howManyTimes) {
    int i;
    for (i=0; i< howManyTimes; i++) {
        digitalWrite(led, HIGH);
        delay(300);
        digitalWrite(led, LOW);
        delay(300);
    }
}


// dergimi i clerave në sistemin RF sender
void writeUInt(unsigned int val)
{
 byte checksum = (val/256) ^ (val&0xFF);
 Serial1.write(0xF0);         // This gets reciever in sync with transmitter
 Serial1.write(g_network_sig, NETWORK_SIG_SIZE);
 Serial1.write((byte*)&val, VAL_SIZE);
 Serial1.write(checksum);     //CHECKSUM_SIZE
}
```

Kodet për microprocessor në anën e marrjes së komandave nga RF.

```
// Kodet për microprocessor ne anen e marrjes se komandave nga Receiver
#define RECEIVER
int freq=500;                                   // shpejtesia e percaktuar nga frekuenca
int motorPins[] = {8, 9, 10, 11};               // stepper motor control pins
int count = 0;
int count2 = 0;
int speedDefaind=32;                            // speed fillesatare
unsigned int rfDataCommand;                     // datat komanda te mara nga RF

// variabli qe percakton drejtimin e rrotullimit te motorit true=drejtimi orar, false=drejtimi anti-
orar
boolean stepperDirection = false;
// whether the string is complete
boolean stringComplete = false;
// Pin 13 eshte LED ne Arduino board
int led = 13;
void setup() {
    // Inicializimi i portes seriale:
    Serial.begin(600);

    // Inicializimi i pineve te motorit si pine dalese
    pinMode(led, OUTPUT);

    // Inizilizimi i pineve te kontrollit te motorit
    for (count = 0; count < 4; count++) {
        pinMode(motorPins[count], OUTPUT);
        }

    //set timer 1 interuppt
    cli();                                      // stop all interuppts
    TCCR1A=0;                                   // set entire register to 0
    TCCR1B=0;                                   // set entire register to 0
    TCNT1=0;                                    // initialize counter value to zero
    OCR1A=(156624/speedDefaind);                // ((1000000)/(1024*freq))-1; // initial value is
15624> for 1Hz
    TCCR1B|=(1<<WGM12);                         // turn on ctc mode
    TCCR1B|=(1<<CS12)|(1<<CS10);                // set CS12 and CS10 to 1 to set prescaller 1024
    TIMSK1|=(1<<OCIE1A);                        // enable timer compare interuppt
    sei();                                      // enable global interrupt flag
}

void loop() {
// Leximi I te dhenave te sjella ne serial nga recever-i
rfDataCommand=readUInt(true);                       // funksioni i cili pret te lexoje te dhenat e
ardhura nga RF

//kontroll i shpejtesise
if(rfDataCommand==100){
    blink(2);                                       // pulso Led dy here
    stepperDirection=true;                          // drejtimi anti-orar
    }
else if(rfDataCommand==101){
    blink(3);                                       // Pulso Led tre here
    stepperDirection=false;                         // drejtimi orar
    }
else if(rfDataCommand==102){
    blink(4);                                       // Pulso Led kater here
    if(speedDefaind<2048) speedDefaind=speedDefaind*2;
    stringComplete = true;
    }
else if(rfDataCommand==103){
```

```
    blink(5);                                          // Pulso Led pese here
    if(speedDefaind>4) speedDefaind=speedDefaind/2;
    stringComplete = true;
    }
else if(rfDataCommand==104){
    blink(6);                                          // Pulso Led gjashte here
    speedDefaind=32;
    stepperDirection = false;
    stringComplete = true;
    }

if(stringComplete){
    OCR1A=(156624/speedDefaind);
    stringComplete = false;
    }
}
// timer 1 interuppt service routine
ISR(TIMER1_COMPA_vect){
    if ((count2 == 0) || (count2 == 1)) {
    count2 = 16;
    }
count2>>=1;

if(stepperDirection==false) {
        for (count = 3; count >= 0; count--) {
        digitalWrite(motorPins[3 - count], count2>>count&0x01);
        }
    }
else if(stepperDirection) {
        for (count = 3; count >= 0; count--) {
        digitalWrite(motorPins[count], count2>>count&0x01);
        }
    }
}

// Blink the LED Pin 13
void blink(int howManyTimes) {
int i;
for (i=0; i< howManyTimes; i++) {
    digitalWrite(led, HIGH);
    delay(300);
    digitalWrite(led, LOW);
    delay(300);
    }
}



// lexon vlerat e sjella nga sistemi RF receiver
unsigned int readUInt(bool wait)
{
 int pos = 0;          // Position in the network signature
 unsigned int val;     // Value of the unsigned int
 byte c = 0;           // Current byte

 if ((Serial.available() < PACKET_SIZE) && (wait == false))
 {
   return 0xFFFF;
 }

 while (pos < NETWORK_SIG_SIZE)
 {
   while (Serial.available() == 0);  // Wait until something is avalible
   c = Serial.read();
```

```
   if (c == g_network_sig[pos])
   {
     if (pos == NETWORK_SIG_SIZE-1)
     {
       byte checksum;

       while(Serial.available() < VAL_SIZE + CHECKSUM_SIZE);   // Wait until something is avalible
       val      =  Serial.read();
       val      += ((unsigned int)Serial.read())*256;
       checksum =  Serial.read();

       if (checksum != ((val/256) ^ (val&0xFF)))
       {
         // Checksum failed
         pos = -1;
       }
     }
     ++pos;
   }
   else if (c == g_network_sig[0])
   {
     pos = 1;
   }
   else
   {
     pos = 0;
     if (!wait)
     {
       return 0xFFFF;
     }
   }
 }
 return val;
}
```

Shtojca 5

Programi për llogaritjen e distancës së sistemit kamera & lazer dhe sistemit të dy kamerave prej një objekti të caktuar.

Form1.frm

```
VERSION 5.00
Begin VB.Form Form1
   BorderStyle     =   1  'Fixed Single
   Caption         =   "DirectShow Webcam Kontroll levizjeje dhe percaktim distance"
   ClientHeight    =   11700
   ClientLeft      =   45
   ClientTop       =   645
   ClientWidth     =   19095
   BeginProperty Font
      Name           =   "Tahoma"
      Size           =   9
      Charset        =   0
      Weight         =   400
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
   EndProperty
   Icon           =   "Form1.frx":0000
   LinkTopic      =   "Form1"
   MaxButton      =   0   'False
   ScaleHeight    =   11700
   ScaleWidth     =   19095
   StartUpPosition =   2  'CenterScreen
   Begin VB.CommandButton cmdKalibrim
      Caption        =   "Kalibrimi"
      Height         =   615
      Left           =   15120
      TabIndex       =   17
      Top            =   360
      Width          =   1215
   End
   Begin VB.TextBox txtDist
      Height         =   375
      Left           =   13680
      TabIndex       =   16
      Top            =   480
      Width          =   1335
   End
   Begin VB.Timer Timer1
      Interval       =   100
      Left           =   4680
      Top            =   4440
   End
   Begin VB.PictureBox Image2
      BorderStyle    =   0  'None
      Height         =   7575
      Left           =   9960
      ScaleHeight    =   7575
      ScaleWidth     =   8895
      TabIndex       =   15
      Top            =   1080
      Width          =   8895
   End
   Begin VB.PictureBox Image1
      Height         =   3855
      Left           =   0
      ScaleHeight    =   3795
```

```
   ScaleWidth     =   4755
   TabIndex       =   2
   Top            =   6360
   Width          =   4815
End
Begin Kontrol_Levizje_Distance.DSPreview dspPreview
   Height         =   3960
   Index          =   0
   Left           =   60
   Top            =   360
   Width          =   4830
   _ExtentX       =   8520
   _ExtentY       =   6985
End
Begin Kontrol_Levizje_Distance.DSPreview dspPreview
   Height         =   3960
   Index          =   1
   Left           =   5040
   Top            =   360
   Width          =   4830
   _ExtentX       =   8520
   _ExtentY       =   6985
End
Begin VB.Label lRreshti
   Caption        =   "Rreshti:"
   BeginProperty Font
      Name           =   "Arial"
      Size           =   13.5
      Charset        =   161
      Weight         =   400
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
   EndProperty
   Height         =   375
   Index          =   1
   Left           =   5640
   TabIndex       =   14
   Top            =   4440
   Width          =   1215
End
Begin VB.Label lKolona
   Caption        =   "Kolona:"
   BeginProperty Font
      Name           =   "Arial"
      Size           =   13.5
      Charset        =   161
      Weight         =   400
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
   EndProperty
   Height         =   375
   Index          =   1
   Left           =   5640
   TabIndex       =   13
   Top            =   4920
   Width          =   1215
End
Begin VB.Label lDistanca
   Caption        =   "Distanca  kamera 2 ne cm: "
   BeginProperty Font
      Name           =   "Arial"
      Size           =   13.5
      Charset        =   161
      Weight         =   400
```

```
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   1
  Left           =   5640
  TabIndex       =   12
  Top            =   5400
  Width          =   3375
End
Begin VB.Label valRreshti
  BeginProperty Font
    Name           =   "Arial"
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   1
  Left           =   7200
  TabIndex       =   11
  Top            =   4440
  Width          =   1695
End
Begin VB.Label valKolona
  BeginProperty Font
    Name           =   "Arial"
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   1
  Left           =   7200
  TabIndex       =   10
  Top            =   4920
  Width          =   1695
End
Begin VB.Label valDistanca
  BeginProperty Font
    Name           =   "Arial"
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   1
  Left           =   7200
  TabIndex       =   9
  Top            =   5880
  Width          =   1935
End
Begin VB.Label lRreshti
  Caption        =   "Rreshti:"
  BeginProperty Font
    Name           =   "Arial"
```

```
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   0
  Left           =   480
  TabIndex       =   8
  Top            =   4440
  Width          =   1215
End
Begin VB.Label lKolona
  Caption        =   "Kolona:"
  BeginProperty Font
    Name           =   "Arial"
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   0
  Left           =   480
  TabIndex       =   7
  Top            =   4920
  Width          =   1215
End
Begin VB.Label lDistanca
  Caption        =   "Distanca  kamera 1 ne cm: "
  BeginProperty Font
    Name           =   "Arial"
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   0
  Left           =   480
  TabIndex       =   6
  Top            =   5400
  Width          =   3375
End
Begin VB.Label valRreshti
  BeginProperty Font
    Name           =   "Arial"
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   0
  Left           =   2040
  TabIndex       =   5
  Top            =   4440
  Width          =   1695
```

```
End
Begin VB.Label valKolona
  BeginProperty Font
    Name           =   "Arial"
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   0
  Left           =   2040
  TabIndex       =   4
  Top            =   4920
  Width          =   1695
End
Begin VB.Label valDistanca
  BeginProperty Font
    Name           =   "Arial"
    Size           =   13.5
    Charset        =   161
    Weight         =   400
    Underline      =   0   'False
    Italic         =   0   'False
    Strikethrough  =   0   'False
  EndProperty
  Height         =   375
  Index          =   0
  Left           =   2040
  TabIndex       =   3
  Top            =   5880
  Width          =   1935
End
Begin VB.Label Label2
  Alignment      =   2  'Center
  Caption        =   "Kamera 2"
  Height         =   255
  Left           =   5040
  TabIndex       =   1
  Top            =   60
  Width          =   4830
End
Begin VB.Label Label1
  Alignment      =   2  'Center
  Caption        =   "Kamera 1"
  Height         =   255
  Left           =   60
  TabIndex       =   0
  Top            =   60
  Width          =   4830
End
Begin VB.Menu mnuCameras
  Caption        =   "Kamera"
  Begin VB.Menu mnuCamerasAddNew
    Caption        =   "Shto nje kamera..."
  End
  Begin VB.Menu mnuCamerasRemove
    Caption        =   "Fshi kameren e zgjedhur"
    Enabled        =   0   'False
  End
End
Begin VB.Menu mnuPreview0
  Caption        =   "Shfaq kameran 1"
  Begin VB.Menu mnuPreview0Deselect
```

```
      Caption         =   "Deselect"
      Enabled         =   0   'False
    End
    Begin VB.Menu mnuPreview0Div1
      Caption         =   "-"
    End
    Begin VB.Menu mnuPreview0Choice
      Caption         =   "none"
      Enabled         =   0   'False
      Index           =   0
    End
  End
  Begin VB.Menu mnuPreview1
    Caption         =   "Shfaq kameran 2"
    Begin VB.Menu mnuPreview1Deselect
      Caption         =   "Deselect"
      Enabled         =   0   'False
    End
    Begin VB.Menu mnuPreview1Div1
      Caption         =   "-"
    End
    Begin VB.Menu mnuPreview1Choice
      Caption         =   "none"
      Enabled         =   0   'False
      Index           =   0
    End
  End
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Private SelectedCamera(1)              As Integer

Private Koeficienti_Sys_Lazer_Kamera        As Double

Private Sub DeselectFailedCamera(ByVal Index As Integer, ByVal Error As Long)
   Dim CameraName As String

   Select Case Index
      Case 0
         With mnuPreview0Choice(SelectedCamera(Index))
            .Checked = False
            CameraName = .Caption
         End With
      Case 1
         With mnuPreview1Choice(SelectedCamera(Index))
            .Checked = False
            CameraName = .Caption
         End With
   End Select
   SelectedCamera(Index) = -1
   SaveSettings
   MsgBox "Selected camera for Preview " & CStr(Index) _
      & " failed, may not be connected:" & vbNewLine _
      & vbNewLine _
      & CameraName & vbNewLine _
      & vbNewLine _
      & "BuildGraph error " & CStr(Error), _
        vbOKOnly Or vbInformation
End Sub

Private Function IsCameraInMenu(ByVal CameraName As String) As Boolean
```

```
    Dim c As Integer

    With mnuPreview0Choice
        For c = 0 To .UBound
            If CameraName = .Item(c).Caption And .Item(c).Enabled Then
                IsCameraInMenu = True
                Exit For
            End If
        Next
    End With
End Function

Private Sub LoadSettings()
    Dim F As Integer
    Dim c As Integer
    Dim CameraName As String

    SelectedCamera(0) = -1 'None.
    SelectedCamera(1) = -1 'None.
    On Error Resume Next
    GetAttr "Settings.txt"
    If Err.Number = 0 Then
        On Error GoTo 0
        F = FreeFile(0)
        Open "Settings.txt" For Input As #F
        Input #F, SelectedCamera(0)
        mnuPreview0Deselect.Enabled = SelectedCamera(0) >= 0
        Input #F, SelectedCamera(1)
        mnuPreview1Deselect.Enabled = SelectedCamera(1) >= 0
        Do Until EOF(F)
            Input #F, CameraName
            If c > 0 Then
                Load mnuPreview0Choice(c)
                Load mnuPreview1Choice(c)
            End If
            With mnuPreview0Choice(c)
                .Enabled = c <> SelectedCamera(1)
                .Caption = CameraName
                .Checked = c = SelectedCamera(0)
            End With
            With mnuPreview1Choice(c)
                .Enabled = c <> SelectedCamera(0)
                .Caption = CameraName
                .Checked = c = SelectedCamera(1)
            End With
            c = c + 1
        Loop
        Close #F
        mnuCamerasRemove.Enabled = True
    End If
End Sub

Private Sub SaveSettings()
    Dim F As Integer
    Dim c As Integer

    F = FreeFile(0)
    Open "Settings.txt" For Output As #F
    Write #F, SelectedCamera(0)
    Write #F, SelectedCamera(1)
    For c = 0 To mnuPreview0Choice.UBound
        Write #F, mnuPreview0Choice(c).Caption
    Next
    Close #F
End Sub
```

```
Private Sub cmdKalibrim_Click()
   Dim DistPikeLazerNgaQendra  As Integer

   Set Image2.Picture = dspPreview(1).Snap
   DistPikeLazerNgaQendra = DistanceKalibrimiPrejQendres(Image2)

   Koeficienti_Sys_Lazer_Kamera = DistPikeLazerNgaQendra * CDbl(txtDist.Text)

End Sub

Private Sub Form_Load()
   Dim Index As Integer
   Dim StartResult As Boolean

   LoadSettings
   For Index = 0 To 1
      If SelectedCamera(Index) >= 0 Then
         Select Case Index
            Case 0
               StartResult = _
                  dspPreview(Index).StartCamera( _
                     mnuPreview0Choice(SelectedCamera(Index)).Caption)
            Case 1
               StartResult = _
                  dspPreview(Index).StartCamera( _
                     mnuPreview1Choice(SelectedCamera(Index)).Caption)
         End Select
         If Not StartResult Then DeselectFailedCamera Index, dspPreview(Index).Error
      End If
   Next
End Sub

Private Sub Form_Unload(Cancel As Integer)
   'Make sure we break the link, don't want to have a memory
   'leak or hang the camera or something:
   dspPreview(0).StopCamera
   dspPreview(1).StopCamera

   Unload Form2
End Sub

Private Sub mnuPreview0Choice_Click(Index As Integer)
   Dim OldIndex As Integer

   OldIndex = SelectedCamera(0)
   If Index <> OldIndex Then
      If OldIndex >= 0 Then
         dspPreview(0).StopCamera
         mnuPreview0Choice(OldIndex).Checked = False
      End If
      SelectedCamera(0) = Index
      mnuPreview0Choice(SelectedCamera(0)).Checked = True
      If dspPreview(0).StartCamera(mnuPreview0Choice(SelectedCamera(0)).Caption) Then
         mnuPreview0Deselect.Enabled = True
         If OldIndex >= 0 Then mnuPreview1Choice(OldIndex).Enabled = True
         mnuPreview1Choice(SelectedCamera(0)).Enabled = False
         SaveSettings
      Else
         DeselectFailedCamera 0, dspPreview(0).Error
      End If
   End If
End Sub

Private Sub mnuPreview0Deselect_Click()
   dspPreview(0).StopCamera
   mnuPreview0Choice(SelectedCamera(0)).Checked = False
```

```
    mnuPreview0Deselect.Enabled = False
    mnuPreview1Choice(SelectedCamera(0)).Enabled = True
    SelectedCamera(0) = -1
    SaveSettings
End Sub

Private Sub mnuPreview1Choice_Click(Index As Integer)
    Dim OldIndex As Integer
    Dim StartResult As Integer

    OldIndex = SelectedCamera(1)
    If Index <> OldIndex Then
        If OldIndex >= 0 Then
            dspPreview(1).StopCamera
            mnuPreview1Choice(OldIndex).Checked = False
        End If
        SelectedCamera(1) = Index
        mnuPreview1Choice(SelectedCamera(1)).Checked = True
        If dspPreview(1).StartCamera(mnuPreview0Choice(SelectedCamera(1)).Caption) Then
            mnuPreview1Deselect.Enabled = True
            If OldIndex >= 0 Then mnuPreview0Choice(OldIndex).Enabled = True
            mnuPreview0Choice(SelectedCamera(1)).Enabled = False
            SaveSettings
        Else
            DeselectFailedCamera 1, dspPreview(1).Error
        End If
    End If
End Sub

Private Sub mnuPreview1Deselect_Click()
    dspPreview(1).StopCamera
    mnuPreview1Choice(SelectedCamera(1)).Checked = False
    mnuPreview1Deselect.Enabled = False
    mnuPreview0Choice(SelectedCamera(1)).Enabled = True
    SelectedCamera(1) = -1
    SaveSettings
End Sub

Private Sub mnuCamerasAddNew_Click()
    Dim Index As Integer

    Form2.Show vbModal, Me
    If Form2.Oked Then
        If IsCameraInMenu(Form2.CameraName) Then
            MsgBox "Camera:" & vbNewLine _
                & vbNewLine _
                & Form2.CameraName & vbNewLine _
                & vbNewLine _
                & "Is already in the menus."
        Else
            Index = mnuPreview0Choice.UBound
            If mnuPreview0Choice(Index).Caption <> "none" Then
                Index = Index + 1
                Load mnuPreview0Choice(Index)
                Load mnuPreview1Choice(Index)
            End If
            With mnuPreview0Choice(Index)
                .Caption = Form2.CameraName
                .Enabled = True
                .Checked = False
            End With
            With mnuPreview1Choice(Index)
                .Caption = Form2.CameraName
                .Enabled = True
                .Checked = False
            End With
```

```
            SaveSettings
            mnuCamerasRemove.Enabled = True
        End If
    End If
End Sub

Private Sub mnuCamerasRemove_Click()
    Dim Index As Integer
    Dim c As Integer

    For Index = 0 To 1
        dspPreview(Index).StopCamera
        SelectedCamera(Index) = -1
        With mnuPreview0Choice(0)
            .Caption = "none"
            .Enabled = False
        End With
        For c = mnuPreview0Choice.UBound To 1 Step -1
            Unload mnuPreview0Choice(c)
        Next
        With mnuPreview1Choice(0)
            .Caption = "none"
            .Enabled = False
        End With
        For c = mnuPreview1Choice.UBound To 1 Step -1
            Unload mnuPreview1Choice(c)
        Next
    Next
    mnuCamerasRemove.Enabled = False
    On Error Resume Next
    Kill "Settings.txt"
End Sub

Private Function GjetjeDistancePrejObjektitModifikuar(ByVal picColor As PictureBox) As LAZER_DISTANCE

    Dim PozicionLazerit As LAZER_DISTANCE

    Const pixR As Integer = 3
    Const pixG As Integer = 2
    Const pixB As Integer = 1


    Dim bitmap_info             As BITMAPINFO
    Dim pixels()                As Byte
    Dim bytes_per_scanLine      As Long
    Dim pad_per_scanLine        As Integer
    Dim X                       As Integer
    Dim Y                       As Integer

    Dim lartesia, gjeresia      As Integer
    Dim rreshti                 As Integer
    Dim kolona                  As Integer
    Dim max_rreshti             As Integer
    Dim max_kolona              As Integer
    Dim max_rreshtiR            As Integer

    Dim koeficient_radian_pixel     As Double
    Dim koeficient_kompesimi_gabimi As Double

    Dim distanca_lazer_kamer    As Double

    Dim pixels_prej_qendres     As Double

    Dim distanca                As Double

    ' Prepare the bitmap description.
```

```
    With bitmap_info.bmiHeader
        .biSize = 40
        .biWidth = picColor.ScaleWidth
        ' Use negative height to scan top-down.
        .biHeight = -picColor.ScaleHeight
        .biPlanes = 1
        .biBitCount = 32
        .biCompression = BI_RGB
        bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \ 32) * 4)
        pad_per_scanLine = bytes_per_scanLine - (((.biWidth * .biBitCount) + 7) \ 8)
        .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
    End With


    ' Calibrated parameter for pixel to distance conversion
    koeficient_radian_pixel = 0.0565
    koeficient_kompesimi_gabimi = 0.002426
    distanca_lazer_kamer = 1.842

    max_rreshtiR = 0

    ' Load the bitmap's data.
    ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To picColor.ScaleHeight)
    GetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info, DIB_RGB_COLORS

    lartesia = picColor.Height / Screen.TwipsPerPixelY
    gjeresia = picColor.Width / Screen.TwipsPerPixelX

    ' Image processing code

    ' The laser dot should not be seen above the middle row (with a little pad)
    For rreshti = 1 To lartesia - 1 Step 10

        ' Our physical setup is roughly calibrated to make the laser
        ' dot in the middle columns...dont bother lookng too far away
        For kolona = 1 To gjeresia - 1 Step 10



            ' Look for the largest red pixel value in the scene (red laser)
            ' If CInt(pixels(pixR, kolona, rreshti)) + CInt(pixels(pixG, rreshti, rreshti)) + CInt(pixels(pixB, rreshti, rreshti)) > max_rreshtiR
Then

            If pixels(pixR, kolona, rreshti) > max_rreshtiR Then
                max_rreshtiR = pixels(pixR, kolona, rreshti)
                max_rreshti = rreshti
                max_kolona = kolona
            End If


        Next

    Next

    ' Calculate the distance for the laser dot from middle of frame
    Dim x_gender, y_gender

    x_gender = CInt(gjeresia / 2)
    y_gender = CInt(lartesia / 2)

    pixels_prej_qendres = Sqr((y_gender - max_rreshti) ^ 2 + (x_gender - max_kolona) ^ 2)

    ' Calculate distanca in cm based on calibrated parameters
    distanca = distanca_lazer_kamer / Tan(pixels_prej_qendres * koeficient_radian_pixel + koeficient_kompesimi_gabimi)

    ' Print laser dot position row and column to screen
```

```
    PozicionLazerit.NrRreshtitLazerit = max_rreshti
    PozicionLazerit.NrKolonesLazerit = max_kolona

    ' Print distanca to laser illuminated object to screen
    PozicionLazerit.distanca = distanca

    If max_kolona > 0 And rreshti > 0 Then
       ' Draw a red vertical line to intersect target
       If max_rreshti > 50 Then
          For rreshti = max_rreshti - 50 To max_rreshti
             pixels(pixR, max_kolona, rreshti) = 255
          Next
       End If

       For rreshti = lartesia / 2 - 50 To lartesia / 2 + 50
          pixels(pixR, x_gender, rreshti) = 127
       Next

       ' Draw a red horizontal line to intersect target
       If max_kolona > 50 Then
          For kolona = max_kolona - 50 To max_kolona - 1
             pixels(pixR, kolona, max_rreshti) = 255
          Next
       End If

       For kolona = gjeresia / 2 - 50 To gjeresia / 2 + 50
          pixels(pixR, kolona, y_gender) = 127
       Next

       ' Display the result.
       SetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info, DIB_RGB_COLORS

       picColor.Picture = picColor.Image

    End If

    GjetjeDistancePrejObjektitModifikuar = PozicionLazerit

End Function


Private Function GjetjeDistancePrejObjektit(ByVal picColor As PictureBox) As LAZER_DISTANCE
    Dim PozicionLazerit As LAZER_DISTANCE
    Const pixR As Integer = 3
    Const pixG As Integer = 2
    Const pixB As Integer = 1
    Dim bitmap_info             As BITMAPINFO
    Dim pixels()                As Byte
    Dim bytes_per_scanLine      As Long
    Dim pad_per_scanLine        As Integer
    Dim X                       As Integer
    Dim Y                       As Integer

    Dim lartesia, gjeresia      As Integer
    Dim rreshti                 As Integer
    Dim kolona                  As Integer
    Dim max_rreshti             As Integer
    Dim max_kolona              As Integer
    Dim max_rreshtiR            As Integer

    Dim koeficient_radian_pixel     As Double
    Dim koeficient_kompesimi_gabimi As Double

    Dim pixels_prej_qendres     As Double

    Dim distanca                As Double
```

```
' Prepare the bitmap description.
With bitmap_info.bmiHeader
   .biSize = 40
   .biWidth = picColor.ScaleWidth
   ' Use negative height to scan top-down.
   .biHeight = -picColor.ScaleHeight
   .biPlanes = 1
   .biBitCount = 32
   .biCompression = BI_RGB
   bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \ 32) * 4)
   pad_per_scanLine = bytes_per_scanLine - (((.biWidth * .biBitCount) + 7) \ 8)
   .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
End With

max_rreshtiR = 0

' Load the bitmap's data.
ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To picColor.ScaleHeight)
GetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info, DIB_RGB_COLORS

lartesia = picColor.Height / Screen.TwipsPerPixelY
gjeresia = picColor.Width / Screen.TwipsPerPixelX

' Kodet per procesimin e imazhit
For rreshti = 1 To lartesia - 1 Step 10
   For kolona = 1 To gjeresia - 1 Step 10
      If pixels(pixR, kolona, rreshti) > max_rreshtiR Then
         max_rreshtiR = pixels(pixR, kolona, rreshti)
         max_rreshti = rreshti
         max_kolona = kolona
      End If
   Next
Next

max_rreshti = max_rreshti + 5
max_kolona = max_kolona + 5

' llogaritja e distaces prej pikes lazer deri ne qendra e faqes se imazhit
Dim x_gender, y_gender

x_gender = CInt(gjeresia / 2)
y_gender = CInt(lartesia / 2)

pixels_prej_qendres = Sqr((y_gender - max_rreshti) ^ 2 + (x_gender - max_kolona) ^ 2)

' Llogaritja e distances ne baze te madhesive 'pixels_prej_qendres' dhe kalibrimit 'Koeficienti_Sys_Lazer_Kamera'
distanca = Koeficienti_Sys_Lazer_Kamera / pixels_prej_qendres

' Percakto pozicionin e rreshtit dhe kolones se pikes lazer.
PozicionLazerit.NrRreshtitLazerit = max_rreshti
PozicionLazerit.NrKolonesLazerit = max_kolona

' Percakto distancen e bojektit nga kamera
PozicionLazerit.distanca = distanca

If max_kolona > 0 And rreshti > 0 Then

   ' Vizato nje vije te kuqe vertikale qe kalon ne piken lazer
   If max_rreshti > 50 Then
      For rreshti = max_rreshti - 50 To max_rreshti
         pixels(pixR, max_kolona, rreshti) = 255
      Next
   End If

   For rreshti = lartesia / 2 - 50 To lartesia / 2 + 50
```

```
            pixels(pixR, x_gender, rreshti) = 127
        Next

        ' Vizato nje vije te kuqe horizontale qe kalon ne piken lazer
        If max_kolona > 50 Then
            For kolona = max_kolona - 50 To max_kolona - 1
                pixels(pixR, kolona, max_rreshti) = 255
            Next
        End If

        For kolona = gjeresia / 2 - 50 To gjeresia / 2 + 50
            pixels(pixR, kolona, y_gender) = 127
        Next

        ' Shfaq imazhin e targuar.
        SetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info, DIB_RGB_COLORS
        picColor.Picture = picColor.Image
    End If
    GjetjeDistancePrejObjektit = PozicionLazerit
End Function


Private Function DistanceKalibrimiPrejQendres(ByVal picColor As PictureBox) As Integer
    Dim PozicionLazerit As LAZER_DISTANCE
    Const pixR As Integer = 3
    Const pixG As Integer = 2
    Const pixB As Integer = 1
    Dim bitmap_info          As BITMAPINFO
    Dim pixels()             As Byte
    Dim bytes_per_scanLine   As Long
    Dim pad_per_scanLine     As Integer
    Dim X                    As Integer
    Dim Y                    As Integer
    Dim lartesia, gjeresia   As Integer
    Dim rreshti              As Integer
    Dim kolona               As Integer
    Dim max_rreshti          As Integer
    Dim max_kolona           As Integer
    Dim max_rreshtiR         As Integer

    Dim koeficient_radian_pixel     As Double
    Dim koeficient_kompesimi_gabimi As Double


    ' Inicializim per bitmap_info bitmap .
    With bitmap_info.bmiHeader
        .biSize = 40
        .biWidth = picColor.ScaleWidth
        .biHeight = -picColor.ScaleHeight
        .biPlanes = 1
        .biBitCount = 32
        .biCompression = BI_RGB
        bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \ 32) * 4)
        pad_per_scanLine = bytes_per_scanLine - (((.biWidth * .biBitCount) + 7) \ 8)
        .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
    End With

    max_rreshtiR = 0

    ' Ngarkim bitmap datash
    ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To picColor.ScaleHeight)
    GetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info, DIB_RGB_COLORS

    lartesia = picColor.Height / Screen.TwipsPerPixelY
    gjeresia = picColor.Width / Screen.TwipsPerPixelX
```

```
    ' Kodet per procesimin e imazhit
    For rreshti = 1 To lartesia - 1 Step 10
        For kolona = 1 To gjeresia - 1 Step 10
            If pixels(pixR, kolona, rreshti) > max_rreshtiR Then
                max_rreshtiR = pixels(pixR, kolona, rreshti)
                max_rreshti = rreshti
                max_kolona = kolona
            End If
        Next
    Next

    ' Llogarit distancen prej pikes lazer deri ne qender te pamjes
    Dim x_gender, y_gender

    x_gender = CInt(gjeresia / 2)
    y_gender = CInt(lartesia / 2)

    DistanceKalibrimiPrejQendres = Sqr((y_gender - max_rreshti) ^ 2 + (x_gender - max_kolona) ^ 2)

End Function

Private Sub Timer1_Timer()
    Dim DistLazer As LAZER_DISTANCE

   ' Set Image1.Picture = dspPreview(0).Snap
   ' DistLazer = GjetjeDistancePrejObjektit(Image1)

  '  valRreshti(0).Caption = DistLazer.NrRreshtitLazerit
  '  valKolona(0).Caption = DistLazer.NrKolonesLazerit
  '  valDistanca(0).Caption = DistLazer.distanca



    Set Image2.Picture = dspPreview(1).Snap
    DistLazer = GjetjeDistancePrejObjektit(Image2)

    valRreshti(1).Caption = DistLazer.NrRreshtitLazerit
    valKolona(1).Caption = DistLazer.NrKolonesLazerit
    valDistanca(1).Caption = DistLazer.distanca

End Sub
```

Form2.frm

```
VERSION 5.00
Begin VB.Form Form2
   BorderStyle     =   1  'Fixed Single
   Caption         =   "Add new camera"
   ClientHeight    =   5130
   ClientLeft      =   45
   ClientTop       =   345
   ClientWidth     =   4275
   BeginProperty Font
      Name           =   "Tahoma"
      Size           =   9
      Charset        =   0
      Weight         =   400
      Underline      =   0   'False
      Italic         =   0   'False
      Strikethrough  =   0   'False
   EndProperty
   LinkTopic       =   "Form2"
   MaxButton       =   0   'False
   ScaleHeight     =   5130
   ScaleWidth      =   4275
```

```
   StartUpPosition =   1  'CenterOwner
   Begin VB.CommandButton cmdCancel
      Cancel          =   -1  'True
      Caption         =   "Cancel"
      Height          =   375
      Left            =   3300
      TabIndex        =   2
      Top             =   4680
      Width           =   855
   End
   Begin VB.CommandButton cmdOk
      Caption         =   "Ok"
      Default         =   -1  'True
      Enabled         =   0   'False
      Height          =   375
      Left            =   2220
      TabIndex        =   1
      Top             =   4680
      Width           =   855
   End
   Begin VB.ListBox lstFilters
      Height          =   4470
      Left            =   0
      Sorted          =   -1  'True
      TabIndex        =   0
      Top             =   0
      Width           =   4275
   End
End
Attribute VB_Name = "Form2"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit

Public Oked As Boolean
Public CameraName As String

Private Sub cmdCancel_Click()
    Oked = False
    lstFilters.SetFocus
    Hide
End Sub

Private Sub cmdOk_Click()
    Oked = True
    With lstFilters
        CameraName = .List(.ListIndex)
    End With
    lstFilters.SetFocus
    Hide
End Sub

Private Sub Form_Activate()
    Dim rfiEach As QuartzTypeLib.IRegFilterInfo

    lstFilters.Clear
    With New QuartzTypeLib.FilgraphManager
        For Each rfiEach In .RegFilterCollection
            lstFilters.AddItem rfiEach.Name
        Next
    End With
End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
```

```
    If UnloadMode = vbFormControlMenu Then
        Cancel = True
        Oked = False
        Hide
    End If
End Sub

Private Sub Form_Resize()
    If WindowState <> vbMinimized Then
        lstFilters.Width = ScaleWidth
    End If
End Sub

Private Sub lstFilters_Click()
    cmdOk.Enabled = lstFilters.ListIndex > -1
End Sub
```

modGlobal.mod

```
Attribute VB_Name = "modGlobal"
Option Explicit

Declare Function GetObject Lib "gdi32" Alias "GetObjectA" (ByVal hObject As Long, ByVal nCount As Long, lpObject As Any) As
Long
Declare Function GetDIBits Lib "gdi32" (ByVal aHDC As Long, ByVal hBitmap As Long, ByVal nStartScan As Long, ByVal
nNumScans As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
Declare Function SetDIBits Lib "gdi32" (ByVal hdc As Long, ByVal hBitmap As Long, ByVal nStartScan As Long, ByVal nNumScans
As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
Type BITMAP '14 bytes
    bmType As Long
    bmWidth As Long
    bmHeight As Long
    bmWidthBytes As Long
    bmPlanes As Integer
    bmBitsPixel As Integer
    bmBits As Long
End Type
Type BITMAPINFOHEADER '40 bytes
    biSize As Long
    biWidth As Long
    biHeight As Long
    biPlanes As Integer
    biBitCount As Integer
    biCompression As Long
    biSizeImage As Long
    biXPelsPerMeter As Long
    biYPelsPerMeter As Long
    biClrUsed As Long
    biClrImportant As Long
End Type
Type RGBQUAD
    rgbBlue As Byte
    rgbGreen As Byte
    rgbRed As Byte
    rgbReserved As Byte
End Type
Type BITMAPINFO
    bmiHeader As BITMAPINFOHEADER
    bmiColors As RGBQUAD
End Type
Public Const DIB_RGB_COLORS = 0&
Public Const BI_RGB = 0&

Type LAZER_DISTANCE
    NrRreshtitLazerit   As Integer
```

```
    NrKolonesLazerit    As Integer
    distanca            As Double
End Type
```

DSDib2Pic.mod

```
Attribute VB_Name = "DSDib2Pic"
Option Explicit

Private Const API_FALSE As Long = 0
Private Const API_TRUE As Long = 1
Private Const API_NULL As Long = 0

Private Const CBM_INIT As Long = &H4&

Private Const DIB_RGB_COLORS As Long = 0
Private Const DIB_PAL_COLORS As Long = 1

Private Enum BiCompressionValues
    BI_RGB = 0
    BI_BITFIELDS = 3
    BI_FOURCC_YUY2 = &H32595559
    BI_FOURCC_UYVY = &H59565955
End Enum

Private Type BITMAPINFOHEADER
    biSize As Long
    biWidth As Long
    biHeight As Long
    biPlanes As Integer
    biBitCount As Integer
    biCompression As BiCompressionValues
    biSizeImage As Long
    biXPelsPerMeter As Long
    biYPelsPerMeter As Long
    biClrUsed As Long
    biClrImportant As Long
End Type

Private Type BITMAPINFO
    bmiHeader As BITMAPINFOHEADER
    bmiColors As Long 'RGBQUAD
End Type

Private Type PictDescBmp
    cbSizeOfStruct As Long
    picType As PictureTypeConstants
    hBitmap As Long
    hPal As Long
End Type

Private Type GUID
    Data1 As Long
    Data2 As Integer
    Data3 As Integer
    Data4(7) As Byte
End Type

Private Declare Function CreateCompatibleDC Lib "gdi32" (ByVal hDC As Long) As Long

Private Declare Function CreateDIBSection Lib "gdi32" ( _
    ByVal hDC As Long, _
    ByVal pbmi As Long, _
    ByVal iUsage As Long, _
    ByRef ppvBits As Long, _
```

```
    ByVal hSection As Long, _
    ByVal dwOffset As Long) As Long

Private Declare Function DeleteDC Lib "gdi32" (ByVal hDC As Long) As Long

Public Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long

Private Declare Function GdiFlush Lib "gdi32" () As Long

Private Declare Sub MoveMemory Lib "kernel32" Alias "RtlMoveMemory" ( _
    ByRef Destination As Any, _
    ByRef pSource As Any, _
    ByVal Length As Long)

Private Declare Function OleCreatePictureIndirect Lib "olepro32" ( _
    ByRef pPicDesc As PictDescBmp, _
    ByRef RefIID As GUID, _
    ByVal fOwn As Long, _
    ByRef IPic As IPicture) As Long

Private Declare Function SelectObject Lib "gdi32" ( _
    ByVal hDC As Long, _
    ByVal hObject As Long) As Long

Private hMemDC As Long
Private hPrevObject As Long

Private Function LongDIB2HBitmap(ByRef LongDIB() As Long) As Long
    'Returns non-0 hBitmap on success.
    Dim bmiHeader As BITMAPINFOHEADER 'Data copied into here for access.
    Dim ColorOffset As Long
    Dim BitsOffset As Long
    Dim Usage As Long
    Dim pBits As Long
    Dim SizeImage As Long

    hMemDC = CreateCompatibleDC(API_NULL)
    If hMemDC Then
        MoveMemory bmiHeader, LongDIB(0), Len(bmiHeader)
        With bmiHeader
            ColorOffset = .biSize \ 4
            'We have a "packed DIB" so biClrUsed is either 0 or
            'the actual color table size!
            BitsOffset = ColorOffset + .biClrUsed
            If .biClrUsed Then Usage = DIB_PAL_COLORS
            Select Case .biCompression
                Case BI_RGB, BI_BITFIELDS
                    'Pad scan line to full width, multiply by height.
                    SizeImage = ((((.biWidth * .biBitCount) + &H1F) And Not &H1F&) \ &H8) _
                            * Abs(.biHeight)
                Case Else
                    SizeImage = .biSizeImage
            End Select
            LongDIB2HBitmap = CreateDIBSection(hMemDC, _
                                        VarPtr(LongDIB(0)), _
                                        Usage, _
                                        pBits, _
                                        API_NULL, _
                                        0)
            If LongDIB2HBitmap <> 0 Then
                GdiFlush
                MoveMemory ByVal pBits, LongDIB(BitsOffset), SizeImage
                hPrevObject = SelectObject(hMemDC, LongDIB2HBitmap)
            End If
        End With
    End If
```

```
End Function

Private Function HBitmap2Picture(ByVal hBitmap As Long, ByVal hPal As Long) As StdPicture
   'Returns Nothing on failure.
   Dim Result As Long
   Dim IPic As IPicture
   Dim IID_IDispatch As GUID
   Dim BmpDesc As PictDescBmp

   With IID_IDispatch
      .Data1 = &H20400
      .Data4(0) = &HC0
      .Data4(7) = &H46
   End With

   With BmpDesc
      .cbSizeOfStruct = Len(BmpDesc)
      .picType = vbPicTypeBitmap
      .hBitmap = hBitmap
      .hPal = hPal
   End With

   Result = OleCreatePictureIndirect(BmpDesc, IID_IDispatch, API_TRUE, IPic)
   If Result = 0 Then
      Set HBitmap2Picture = IPic
   End If
End Function

Public Function LongDIB2Picture(ByRef LongDIB() As Long) As StdPicture
   Dim hBitmap As Long

   hBitmap = LongDIB2HBitmap(LongDIB)
   If hBitmap Then
      Set LongDIB2Picture = HBitmap2Picture(hBitmap, 0)
      SelectObject hMemDC, hPrevObject
      hPrevObject = 0
   End If
   If hMemDC Then
      DeleteDC hMemDC
   End If
   hMemDC = 0
End Function
```

DSPreview.ctl

```
VERSION 5.00
Begin VB.UserControl DSPreview
   Appearance      =   0  'Flat
   CanGetFocus     =   0   'False
   ClientHeight    =   3600
   ClientLeft      =   0
   ClientTop       =   0
   ClientWidth     =   4800
   ClipBehavior    =   0  'None
   ClipControls    =   0   'False
   HitBehavior     =   0  'None
   PaletteMode     =   4  'None
   ScaleHeight     =   3600
   ScaleWidth      =   4800
   ToolboxBitmap   =   "DSPreview.ctx":0000
   Begin VB.Shape shpBorder
      BorderColor     =   &H80000006&
      Height          =   3375
      Left            =   120
      Top             =   120
```

```
    Width          =   4515
  End
End
Attribute VB_Name = "DSPreview"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = False
Attribute VB_Exposed = False
Option Explicit
'
'DSPreview control
'
'Requires a reference to:
'
'   ActiveMovie control type library (quartz.dll).
'

'=== Public Enums ====================================================

Public Enum BorderStyles
   bsNone
   bsFixedSingle
End Enum
#If False Then
Dim bsNone, bsFixedSingle
#End If

'FILTER_STATE values, should have been defined in Quartz.dll,
'but another item Microsoft left out.
Public Enum FILTER_STATE
   State_NotStarted = -1 'No FilgraphManager object, i.e. preview not started.
   State_Stopped = 0
   State_Paused = 1
   State_Running = 2
End Enum
#If False Then
Dim State_Error, State_Stopped, State_Paused, State_Running
#End If

'=== Private Declarations ============================================

Private Const WS_BORDER = &H800000
Private Const WS_DLGFRAME = &H400000
Private Const WS_SYSMENU = &H80000
Private Const WS_THICKFRAME = &H40000
Private Const MASKBORDERLESS = Not (WS_BORDER Or WS_DLGFRAME Or WS_SYSMENU Or WS_THICKFRAME)
Private Const MASKBORDERMIN = Not (WS_DLGFRAME Or WS_SYSMENU Or WS_THICKFRAME)

Private Const E_FAIL As Long = &H80004005

'These are "scripts" followed by BuildGraph() below to create a
'DirectShow FilterGraph for webcam viewing.
'
'FILTERLIST is incomplete, and must be prepended with the name
'of your webcam's Video Capture Source filter.  Since there may
'be multiples, FILTERLIST begins with "~Capture" which is used
'when BuildGraph() interprets this script to select one having
'a pin named "Capture".
Private Const FILTERLIST As String = _
       "~Capture|" _
     & "AVI Decompressor|" _
     & "Color Space Converter|" _
     & "Video Renderer"
Private Const CONNECTIONLIST As String = _
       "Capture~XForm In|" _
     & "XForm Out~Input|" _
```

```
    & "XForm Out~VMR Input0"

Private fgmVidCap As QuartzTypeLib.FilgraphManager 'Not "Is Nothing" means camera is previewing.
Private bv2VidCap As QuartzTypeLib.IBasicVideo2
Private vwVidCap As QuartzTypeLib.IVideoWindow
Private SelectedCamera As Integer '-1 means none selected.
Private InsideWidth As Double 'UserControl's ScaleMode units.
Private BorderOffset As Long 'Pixels.
Private DesignHeight As Single 'UserControl's initial height.

Private AspectRatio As Double

'=== Private Property Caches ========================================

Private mBorderStyle As BorderStyles

'=== Public Properties ==============================================

'Error values:
'
'         0: No error.
'     1 - 9: Failure processing FILTERLIST script at item "Error."
'   101 - 109: Failure processing CONNECTIONLIST script at item "Error."
'       200: Not running preview.
'       300: Failed to enter paused state.
'       400: Failed to resume into running state.
'       500: StartCamera() called but already started.
'       600: Snap() failed to create StdPicture snapshot.
Public Error As Integer

Public Property Get BackColor() As OLE_COLOR
   BackColor = UserControl.BackColor
End Property

Public Property Let BackColor(ByVal RHS As OLE_COLOR)
   UserControl.BackColor = RHS
   PropertyChanged "BackColor"
End Property

Public Property Get BorderStyle() As BorderStyles
   BorderStyle = mBorderStyle
End Property

Public Property Let BorderStyle(ByVal RHS As BorderStyles)
   If bsNone > RHS Or RHS > bsFixedSingle Then Err.Raise 5, TypeName(Me)
   mBorderStyle = RHS
   shpBorder.Visible = mBorderStyle = bsFixedSingle
   PropertyChanged "BorderStyle"
End Property

Public Property Get State() As FILTER_STATE
   If fgmVidCap Is Nothing Then
      State = State_NotStarted
   Else
      fgmVidCap.GetState 0, State
   End If
End Property

'=== Public Methods =================================================

Public Function PauseCamera() As Boolean
   'Returns True on success, error in Error.
   Const PauseWaitMs As Long = 16
   Dim State As FILTER_STATE

   If fgmVidCap Is Nothing Then
```

```
      Error = 200
      Exit Function
   End If

   With fgmVidCap
      .Pause
      Do
         .GetState PauseWaitMs, State
      Loop Until State = State_Paused Or Err.Number = E_FAIL
      If Err.Number = E_FAIL Then
         Error = 300
         Exit Function
      End If
   End With

   PauseCamera = True
End Function

Public Function ResumeCamera() As Boolean
   'Returns True on success, error in Error.
   Const ResumeWaitMs As Long = 16
   Dim State As FILTER_STATE

   If fgmVidCap Is Nothing Then
      Error = 200
      Exit Function
   End If

   With fgmVidCap
      .Run
      Do
         .GetState ResumeWaitMs, State
         Error = Err.Number
      Loop Until State = State_Running Or Error = E_FAIL
      If Error = E_FAIL Then
         Error = 400
         Exit Function
      Else
      Error = 0
      End If
   End With

   ResumeCamera = True
End Function

Public Function Snap() As StdPicture
   'Returns Nothing on failure.
   Const PauseWaitMs As Long = 16
   Const biSize = 40 'BITMAPINFOHEADER and not BITMAPV4HEADER, etc. but we don't get those.
   Dim State As FILTER_STATE
   Dim Size As Long
   Dim DIB() As Long

   If fgmVidCap Is Nothing Then
      Error = 200
      Exit Function
   End If

   With fgmVidCap
      .Pause
      Do
         .GetState PauseWaitMs, State
         Error = Err.Number
      Loop Until State = State_Paused Or Error = E_FAIL
      If Error = E_FAIL Then
         Error = 300
```

```
            Exit Function
        Else
            Error = 0
        End If

        With bv2VidCap
            'Estimate size.  Correct for 32-bit RGB and generous
            'for anything with fewer bits per pixel, compressed,
            'or palette-ized (we hope).
            Size = biSize + .VideoWidth * .VideoHeight
            ReDim DIB(Size - 1)
            Size = Size * 4 'To bytes.
            .GetCurrentImage Size, DIB(0)
        End With

        .Run
    End With

    Set Snap = LongDIB2Picture(DIB)
    If Snap Is Nothing Then
        Error = 600
    End If
End Function

Public Function StartCamera(ByVal CameraName As String) As Boolean
    'Returns True on success, error in Error.

    If Not fgmVidCap Is Nothing Then
        Error = 500
        Exit Function
    End If

    Set fgmVidCap = New QuartzTypeLib.FilgraphManager
    'Tack camera name onto FILTERLIST and try to start it.  Add 1 for error reporting.
    Error = BuildGraph(fgmVidCap, CameraName & FILTERLIST, CONNECTIONLIST) + 1
    If Error >= 1 Then
        Set fgmVidCap = Nothing
        Exit Function
    End If

    Set bv2VidCap = fgmVidCap
    With bv2VidCap
        AspectRatio = CDbl(.VideoHeight) / CDbl(.VideoWidth)
    End With
    shpBorder.Visible = False
    Height = Width * AspectRatio

    Set vwVidCap = fgmVidCap
    With vwVidCap
        .FullScreenMode = False
        .Left = 0
        .Top = 0
        .Width = ScaleX(ScaleWidth, ScaleMode, vbPixels)
        .Height = ScaleY(ScaleWidth * AspectRatio, ScaleMode, vbPixels)
        If mBorderStyle = bsFixedSingle Then
            .WindowStyle = .WindowStyle And MASKBORDERMIN
        Else
            .WindowStyle = .WindowStyle And MASKBORDERLESS
        End If
        .Owner = hWnd
        .Visible = True
    End With

    StartCamera = True
    fgmVidCap.Run
End Function
```

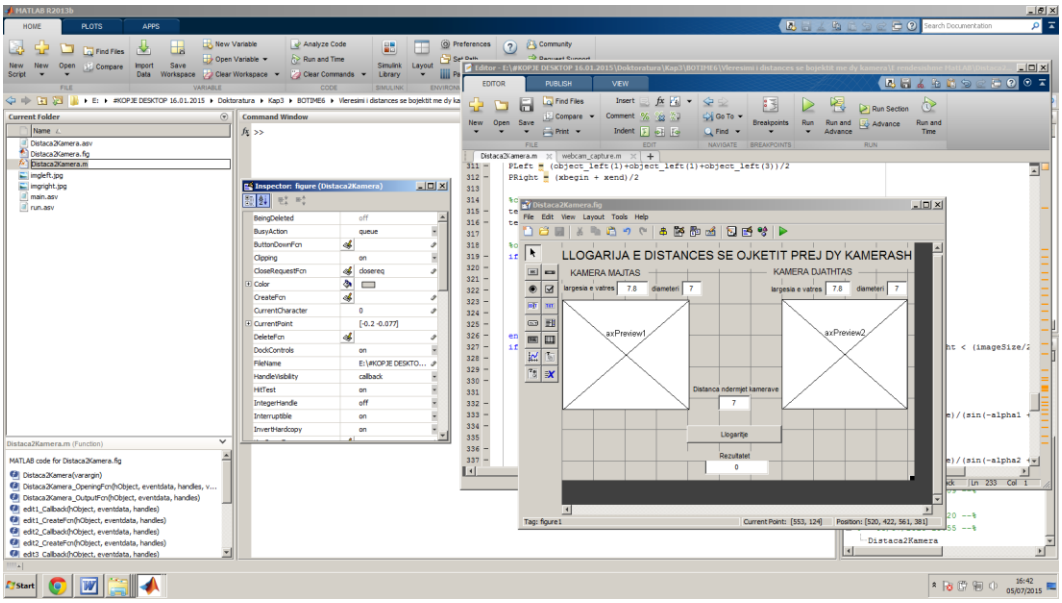
```
Public Sub StopCamera()
   Const StopWaitMs As Long = 40
   Dim State As FILTER_STATE

   If Not fgmVidCap Is Nothing Then
      With fgmVidCap
         .Stop
         Do
            .GetState StopWaitMs, State
            Error = Err.Number
         Loop Until State = State_Stopped Or Error = E_FAIL
      End With
      Error = 0

      With vwVidCap
         .Visible = False
         .Owner = 0
      End With
      Set vwVidCap = Nothing
      Set bv2VidCap = Nothing
      Set fgmVidCap = Nothing

      Height = DesignHeight
      shpBorder.Visible = mBorderStyle = bsFixedSingle
   End If
End Sub

'=== Private Methods ============================================

Private Function BuildGraph( _
   ByVal FGM As QuartzTypeLib.FilgraphManager, _
   ByVal Filters As String, _
   ByVal Connections As String) As Integer
   'Returns -1 on success, or FilterIndex when not found, or
   'ConnIndex + 100 when a pin of the connection not found.
   '
   'Filters:
   '
   '   A string with Filter Name values separated by "|" delimiters
   '   and optionally each of these can be followed by one required
   '   Pin Name value separated by a "~" delimiter for use as a tie
   '   breaker when there might be multiple filters with the same
   '   Name value.
   '
   'Connections:
   '
   '   A string with a list of output pins to be connected to
   '   input pins.  Each pin-pair is separated by "|" delimiters
   '   and each pair has out and in pins separated by a "~"
   '   delimiter.  The pin-pairs should be one less than the number
   '   of filters.
   Dim FilterNames() As String
   Dim FilterIndex As Integer
   Dim FilterParts() As String
   Dim FoundFilter As Boolean
   Dim rfiEach As QuartzTypeLib.IRegFilterInfo
   Dim fiFilters() As QuartzTypeLib.IFilterInfo
   Dim Conns() As String
   Dim ConnIndex As Integer
   Dim ConnParts() As String
   Dim piEach As QuartzTypeLib.IPinInfo
   Dim piOut As QuartzTypeLib.IPinInfo
   Dim piIn As QuartzTypeLib.IPinInfo

   'Setup for filter script processing.
```

```
    FilterNames = Split(UCase$(Filters), "|")
    ReDim fiFilters(UBound(FilterNames))

    'Find and add filters.
    For FilterIndex = 0 To UBound(FilterNames)
        FilterParts = Split(FilterNames(FilterIndex), "~")
        For Each rfiEach In FGM.RegFilterCollection
            If UCase$(rfiEach.Name) = FilterParts(0) Then
                rfiEach.Filter fiFilters(FilterIndex)
                If UBound(FilterParts) > 0 Then
                    For Each piEach In fiFilters(FilterIndex).Pins
                        If UCase$(piEach.Name) = FilterParts(1) Then
                            FoundFilter = True
                            Exit For
                        End If
                    Next
                Else
                    FoundFilter = True
                    Exit For
                End If
            End If
        Next
        If FoundFilter Then
            FoundFilter = False
        Else
            BuildGraph = FilterIndex
            Exit Function 'Error result will be 0, 1, etc.
        End If
    Next
    BuildGraph = -1

    'Setup for connection script processing.
    Conns = Split(UCase$(Connections), "|")
    FilterIndex = 0

    'Find and connect pins.
    For ConnIndex = 0 To UBound(Conns)
        ConnParts = Split(Conns(ConnIndex), "~")
        For Each piEach In fiFilters(FilterIndex).Pins
            If UCase$(piEach.Name) = ConnParts(0) Then
                Set piOut = piEach
                Exit For
            End If
        Next
        For Each piEach In fiFilters(FilterIndex + 1).Pins
            If UCase$(piEach.Name) = ConnParts(1) Then
                Set piIn = piEach
                Exit For
            End If
        Next
        If piOut Is Nothing Or piIn Is Nothing Then
            'Error, missing a pin.
            BuildGraph = ConnIndex + 100 'Error result will be 100, 101, etc.
            Exit Function
        End If
        piOut.ConnectDirect piIn
        FilterIndex = FilterIndex + 1
    Next
End Function

'=== Event Handlers ================================================

Private Sub UserControl_Initialize()
End Sub

Private Sub UserControl_InitProperties()
```

```
    BorderStyle = bsFixedSingle
End Sub

Private Sub UserControl_ReadProperties(PropBag As PropertyBag)
    With PropBag
        UserControl.BackColor = .ReadProperty("BackColor", vbButtonFace)
        BorderStyle = .ReadProperty("BorderStyle", bsFixedSingle)
    End With
    DesignHeight = Height
End Sub

Private Sub UserControl_Resize()
    shpBorder.Move 0, 0, ScaleWidth, ScaleHeight
End Sub

Private Sub UserControl_Terminate()
    StopCamera
End Sub

Private Sub UserControl_WriteProperties(PropBag As PropertyBag)
    With PropBag
        .WriteProperty "BackColor", UserControl.BackColor, vbButtonFace
        .WriteProperty "BorderStyle", mBorderStyle, bsFixedSingle
    End With
End Sub
```

Shtojca 6

Programi për llogaritjen e distancës së sistemit prej dy kamerash prej një objekti të caktuar në Matlab R2013b.



Distaca2Kamera.m

```
function varargout = Distaca2Kamera(varargin)
% DISTACA2KAMERA MATLAB code for Distaca2Kamera.fig
%      DISTACA2KAMERA, by itself, creates a new DISTACA2KAMERA or raises the existing
%      singleton*.
%
%      H = DISTACA2KAMERA returns the handle to a new DISTACA2KAMERA or the handle to
%      the existing singleton*.
%
%      DISTACA2KAMERA('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in DISTACA2KAMERA.M with the given input arguments.
%
%      DISTACA2KAMERA('Property','Value',...) creates a new DISTACA2KAMERA or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Distaca2Kamera_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Distaca2Kamera_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Distaca2Kamera

% Last Modified by GUIDE v2.5 05-Jul-2015 16:05:36

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Distaca2Kamera_OpeningFcn, ...
                   'gui_OutputFcn',  @Distaca2Kamera_OutputFcn, ...
```

```
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before Distaca2Kamera is made visible.
function Distaca2Kamera_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Distaca2Kamera (see VARARGIN)

% Choose default command line output for Distaca2Kamera
handles.output = hObject;


vid1 = videoinput('winvideo', 1);
% only capture one frame per trigger, we are not recording a video
vid1.FramesPerTrigger = 1;
% output would image in RGB color space
vid1.ReturnedColorspace = 'rgb';
% tell matlab to start the webcam on user request, not automatically
triggerconfig(vid1, 'manual');
% we need this to know the image height and width
vidRes = get(vid1, 'VideoResolution');
% image width
imWidth = vidRes(1);
% image height
imHeight = vidRes(2);
% number of bands of our image (should be 3 because it's RGB)
nBands = get(vid1, 'NumberOfBands');
% create an empty image container and show it on axPreview
hImageL = image(zeros(imHeight, imWidth, nBands), 'parent', handles.axPreview1);
% begin the webcam preview
preview(vid1, hImageL);



vid2 = videoinput('winvideo', 2);
% only capture one frame per trigger, we are not recording a video
vid2.FramesPerTrigger = 1;
% output would image in RGB color space
vid2.ReturnedColorspace = 'rgb';
% tell matlab to start the webcam on user request, not automatically
triggerconfig(vid2, 'manual');
% we need this to know the image height and width
vidRes = get(vid2, 'VideoResolution');
% image width
imWidth = vidRes(1);
% image height
imHeight = vidRes(2);
% number of bands of our image (should be 3 because it's RGB)
nBands = get(vid2, 'NumberOfBands');
% create an empty image container and show it on axPreview
hImageR = image(zeros(imHeight, imWidth, nBands), 'parent', handles.axPreview2);
% begin the webcam preview
preview(vid2, hImageR);
```

```
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Distaca2Kamera wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Distaca2Kamera_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

left_image_file ='imgleft.jpg';
FLeft = getframe(handles.axPreview1);
% image(FLeft.cdata);
ImageL = frame2im(FLeft);
% imwrite(FLeft.cdata, left_image_file);
imwrite(ImageL, left_image_file);
focal_left = str2double(get(handles.edit1,'String'));


right_image_file ='imgright.jpg';
FRight = getframe(handles.axPreview2);
% image(FRight.cdata);
ImageR = frame2im(FLeft);
% imwrite(FRight.cdata, right_image_file);
imwrite(ImageR, right_image_file);
focal_right = str2double(get(handles.edit3,'String'));
```

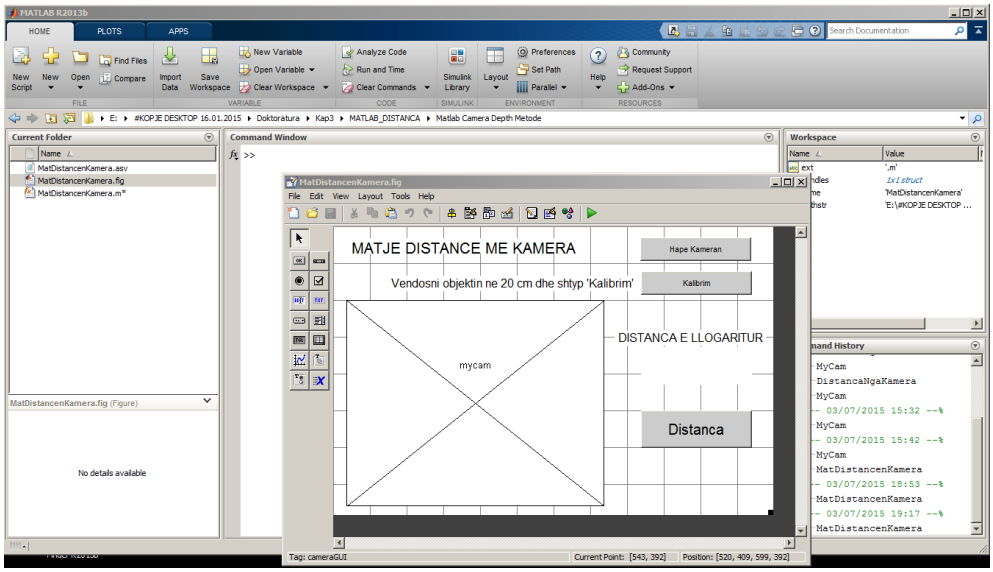
```
diam_left = str2double(get(handles.edit2,'String'));
diam_right = str2double(get(handles.edit4,'String'));

cam_distance = str2double(get(handles.edit5,'String'));

left = imread(left_image_file);
right = imread(right_image_file);

figure , [object,object_left] = imcrop(left);


% interactively
object = object; % choose one petal of object
rsize = size(object);
rect_object = [0 0 rsize(2) rsize(1)];
%[sub_right,rect_right] = imcrop(right); % choose whole object
fsize = size(right);
rect_right = [0 0 fsize(2) fsize(1)];

sub_right=imcrop(right,rect_right);

c = normxcorr2(object(:,:,1),sub_right(:,:,1));
%figure, surf(c), shading flat

% offset found by correlation
[max_c, imax] = max(abs(c(:)));
[ypeak, xpeak] = ind2sub(size(c),imax(1));
corr_offset = [(xpeak-size(object,2))
               (ypeak-size(object,1))];

% relative offset of position of subimages
rect_offset = [(rect_right(1)-rect_object(1))
               (rect_right(2)-rect_object(2))];

% total offset
offset = corr_offset + rect_offset;
absof = offset;
xoffset = absof(1);
yoffset = absof(2);

xbegin = xoffset+1;

xend   = xoffset+ size(object,2);
ybegin = yoffset+1;
yend   = yoffset+size(object,1);

% extract region from right and compare to object
extracted_object = right(ybegin:yend,xbegin:xend,:);
% if isequal(object,extracted_object)
%    disp('object.tif was extracted from right.tif');
% end


recovered_object = uint8(zeros(size(right)));
recovered_object(ybegin:yend,xbegin:xend,:) = object;
%figure, imshow(recovered_object)


[m,n,p] = size(right);
mask = ones(m,n);
i = find(recovered_object(:,:,1)==0);
mask(i) = .2; % try experimenting with different levels of
              % transparency

% overlay images with transparency
figure, imshow(right(:,:,1)); % show only red plane of right
```

```
hold on
h = imshow(recovered_object); % overlay recovered_object
set(h,'AlphaData',mask);

imageSize = fsize(2)
%Vlerat e qendres se objektit ne imazhin e majte dhe te djathte
PLeft = (object_left(1)+object_left(1)+object_left(3))/2
PRight = (xbegin + xend)/2

% llogaritja e distances dhe kthimi i vleres
teta1 = atan(diam_left/(focal_left*2)) % gjysem kendi i pamjes se kameres se majte
teta2 = atan(diam_right/(focal_right*2)) % gjysem kendi i pamjes se kameres se djathte

% objekti eshte midis lentes se najte dhe te dhjathte
if ((PLeft  > (imageSize/2))&&(PRight < (imageSize/2)))
    disp('CASE 1!!!');
    alpha1 = atan(((PLeft - (imageSize/2))/(imageSize/2))*tan(teta1));
    alpha2 = atan((((imageSize/2) - PRight)/(imageSize/2))*tan(teta2));
    mone = tan((pi/2) - alpha1)*tan((pi/2) - alpha2)*cam_distance;
    mehane = tan((pi/2) - alpha1) + tan((pi/2) - alpha2);
    object_distance = mone /mehane
end
if ((PLeft  > (imageSize/2))&&(PRight > (imageSize/2))||(PLeft  < (imageSize/2))&&(PRight <
(imageSize/2)))
         alpha1 = atan((((imageSize/2) - PLeft)/(imageSize/2))*tan(teta1));
         alpha2 = atan(((PRight - (imageSize/2))/(imageSize/2))*tan(teta2));
         if ((PLeft  > (imageSize/2))&&(PRight > (imageSize/2)))
             % objekti eshte ne te majte te lentes se majte dhe te djathte
             disp('CASE 2!!!');
             object_distance = (sin((pi/2) - alpha1)*(sin((pi/2) -
alpha2))*cam_distance)/(sin(-alpha1 + alpha2))
         else
              %objekti eshte ne te djathte te lentes se majte dhe te djathte
             disp('CASE 3!!!');
             object_distance = (sin((pi/2) - alpha1)*(sin((pi/2) -
alpha2))*cam_distance)/(sin(-alpha2 + alpha1))
         end
end
if  (PLeft  == (imageSize/2))
    % objekti eshte perpara kameres se majte
    disp('CASE 4!!!\n');
    alpha2 = atan((((imageSize/2) - PRight)/(imageSize/2))*tan(teta2));
    object_distance = tan((pi/2) - alpha2)*cam_distance
end
if (PRight == (imageSize/2))
    %objekti eshte perpara kameres se djathte
    disp('CASE 5!!!\n');
    alpha1 = atan(((PLeft - (imageSize/2))/(imageSize/2))*tan(teta1));
    object_distance = tan((pi/2) - alpha1)*cam_distance
end


set(handles.edit6, 'String' , object_distance);



function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%        str2double(get(hObject,'String')) returns contents of edit5 as a double


% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6 as a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: delete(hObject) closes the figure
delete(hObject);
```

Shtojca 7

Programi për llogaritjen sipas metodës së vlerësimit të distancës së objekteve prej një kamere referuar vlerësimit të thellësisë (Depth Estimation) në mjedisin Matlab R2013b.



MatDistancenKamera.m

```
function varargout = MatDistancenKamera(varargin)
% MatDistancenKamera M-file for MatDistancenKamera.fig
%      MatDistancenKamera, by itself, creates a new MatDistancenKamera or raises the
existing
%      singleton*.
%
%      H = MatDistancenKamera returns the handle to a new MatDistancenKamera or the handle
to
%      the existing singleton*.
%
%      MatDistancenKamera('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MatDistancenKamera.M with the given input arguments.
%
%      MatDistancenKamera('Property','Value',...) creates a new MatDistancenKamera or raises
the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before MatDistancenKamera_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to MatDistancenKamera_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MatDistancenKamera

% Last Modified by GUIDE v2.5 03-Jul-2015 19:11:32

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
```

```
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @MatDistancenKamera_OpeningFcn, ...
                   'gui_OutputFcn',  @MatDistancenKamera_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before MyCam is made visible.
function MatDistancenKamera_OpeningFcn(hObject, eventdata, handles, varargin) %funcion que
controla el video
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to My (see VARARGIN)

% Choose default command line output for MatDistancenKamera
handles.output = hObject;

%--------------------------------------------------------------------
% Create video object
% Putting the object into manual trigger mode and then
% starting the object will make GETSNAPSHOT return faster
% since the connection to the camera will already have
% been established.
handles.video = videoinput('winvideo',1);                                % thirret kamera
handles.video.ReturnedColorspace = 'rgb';                                % imazhi vjen ne
formatin rgb
set(handles.video,'TimerPeriod', 0.05, ...
                  'TimerFcn',['if(~isempty(gco)),'...
                  'handles=guidata(gcf);'...                             % Update handles
                  'image(getsnapshot(handles.video));'...                % get the image and
place it in Axes
                  'set(handles.mycam,''ytick'',[],''xtick'',[]),'...     % Remove tickmarks
and labels that are inserted when using IMAGE
                  'else '...
                  'delete(imaqfind);'...                                 % pastro objektet
imazhe ekzistues
                  'end']);
triggerconfig(handles.video,'manual');

%--------------------------------------------------------------------

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes MyCam wait for user response (see UIRESUME)
uiwait(handles.cameraGUI);


% --- Outputs from this function are returned to the command line.
function varargout = MatDistancenKamera_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
handles.output = hObject;
varargout{1} = handles.output;


% --- Executes on button press in startcam.
function startcam_Callback(hObject, eventdata, handles)
% hObject    handle to startcam (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Inicio/Fin de Cámara

% --- Executes on button press in Distanca.
function Distanca_Callback(hObject, eventdata, handles)
% hObject    handle to Distanca (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

t = timer;
set(t,'Tag', 'distanceLoop'); % this is opcional... its just a name for the timer
set(t,'Period',0.5); % this will make the timer run every 0.8 seconds
set(t, 'ExecutionMode', 'FixedDelay'); %this means it will run when the other one has
started counting time
set(t, 'TimerFcn', @(x,y)LlogaritDistancen(handles.video,handles.dist)); % this tells the
program what to do when the time is up (call the funtion)
start(t); %start timer

function [ ] = LlogaritDistancen(video,dist)
% marrje imazhi nga kamera per perpunim dhe llogaritja e distances
% ------------------------------
[w h minc minf] = LlogaritGjeresiGjatesiObjektiPlan(video);

%this draws the rectangle on the object and presents the distanca
global factor;
if(w>0 && h>0)
    % nderimi i konturit drejkendesh
    % ------------------------------
    rectangle('Position',[minc,minf,w,h]);

    area = w*h;

    % distancia =( -10.73*log((area/factor)*21720)) + 127.38;
    distancia =( -10.73*log((area/factor)*21720)) + 127.15;

    str=sprintf('%0.4f',distancia);
else
    str= '0';
end

set(dist,'String',str);


function [ ] = DrawRect(video,dist)
% nderimi i konturit drejkendesh
% ------------------------------
[w h minc minf] = LlogaritGjeresiGjatesiObjektiPlan(video);

if(w>0 & h>0)
    rectangle('Position',[minc,minf,w,h]);
end

function [w, h, minc, minf] = LlogaritGjeresiGjatesiObjektiPlan(video)

maxR=214; % red
maxG=214; % green
maxB=214; % blue
```

```
pause(.4);                                  % pritje prej 0.4 sek per inicializim te AGC

I = getsnapshot(video);                     % procesi i analizes se imazhit duke e kaluar te
                                            % zberthyer ne R,G,B ne matricen I
% procesi i gjetjes se permasave te objektit me te ndritshem kundrejt sfondit
% f - vektor per rreshtat me vlere te rgb > 214
% c - vektor per kolona me vlere te rgb > 214
% vleresimi per te kuqen (red)
[f, c] = find(I(:,:,1) > maxR);
mincr = min(c);
minfr = min(f);
maxcr = max(c);
maxfr = max(f);

% vleresimi per te jeshile (green)
[f, c] = find(I(:,:,2) > maxG);
mincg = min(c);
minfg = min(f);
maxcg = max(c);
maxfg = max(f);

[f, c] = find(I(:,:,3) > maxB);
mincb = min(c);
minfb = min(f);
maxcb = max(c);
maxfb = max(f);

% this frames the image that the cam is seeing

if(~isempty(minfb) && ~isempty(minfg) && ~isempty(minfr))
    minf= max([minfb,minfg,minfr]);
else
    minf=0;
end
if(~isempty(mincb) && ~isempty(mincg) && ~isempty(mincr))
    minc= max([mincb,mincg,mincr]);
else
    minc=0;
end
if(~isempty(maxfb) && ~isempty(maxfg) && ~isempty(maxfr))
    maxf= min([maxfb,maxfg,maxfr]);
else
    maxf=0;
end
if(~isempty(maxcb) && ~isempty(maxcg) && ~isempty(maxcr))
    maxc= min([maxcb,maxcg,maxcr]);
else
    maxc=0;
end

%this calculates height and width from the object
w=(maxc-minc);
h=(maxf-minf);;



% --- Executes on button press in Kalibrim.
function Kalibrim_Callback(hObject, eventdata, handles)
% hObject    handle to HapeKameran (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global hiloCalibrate
global  factor
%handles the string on the button that starts the program and cam

[w h minc minf] = LlogaritGjeresiGjatesiObjektiPlan(handles.video);
if(w>0 & h>0)
```

```
    rectangle('Position',[minc,minf,w,h]);
    stop(hiloCalibrate);
    factor=w*h;
end

% --- Executes when user attempts to close cameraGUI.
function cameraGUI_CloseRequestFcn(hObject, eventdata, handles)
stop(handles.video);
% hObject    handle to cameraGUI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: delete(hObject) closes the figure
delete(hObject);

% --- Executes on button press in HapeKameran.
function HapeKameran_Callback(hObject, eventdata, handles)
global hiloCalibrate
start(handles.video);
pause(.4);                                              % Kohe per inicializim te
kameras AGC
hiloCalibrate = timer;
set(hiloCalibrate,'Tag', 'distanceLoop');               % emertimi per timer-in
set(hiloCalibrate,'Period',0.5);                        % intervali per timer-in do te
executohet ne cikel cdo 0.5 sek
set(hiloCalibrate, 'ExecutionMode', 'FixedDelay');
set(hiloCalibrate, 'TimerFcn', @(x,y)DrawRect(handles.video,handles.dist)); % funksioni qe
do te exekutohet ne cdo 0.5 sek
start(hiloCalibrate);                                   % filim i ciklit te timer-in
```

Shtojca 8

Në Shtojcën-8 po japim programin e plotë të ndërtuar në Visual Basic 6.0 për komandimin e modulit PIC18F4550-I/P Microchip Microcontroller & ULN2803 dhe modulit Stepper-Bee në ndjekje të objektit në lëvizje që për rastin në trajtim lidhet me panelin diellor.

Moduli : frmCamera.frm

```
Option Explicit

Private Const WM_USER As Long = &H400
Private Const WM_CAP_DRIVER_CONNECT As Long = (WM_USER + &HA)
Private Const WM_CAP_DRIVER_DISCONNECT As Long = (WM_USER + &HB)
Private Const WM_CAP_SET_PREVIEWRATE As Long = (WM_USER + &H34)
Private Const WM_CAP_SET_PREVIEW As Long = (WM_USER + &H32)
Private Const WM_CAP_GET_STATUS As Long = (WM_USER + &H36)


Private Const HWND_TOP As Long = 0
Private Const SWP_NOMOVE As Long = &H2
Private Const SWP_NOZORDER As Long = &H4
Private Const SWP_NOENDCHANGING As Long = &H400
Private Const WM_CAP_START As Long = WM_USER

Private Const WM_CAP_GET_MCI_DEVICEA As Long = (WM_CAP_START + 67)

Private Const GET_FRAME As Long = 1084
Private Const COPY As Long = 1054
Private Const WM_CAP_EDIT_COPY = WM_USER + 30
Private Const WM_CAP_DLG_VIDEOSOURCE As Long = (WM_CAP_START + 42)
Private Declare Function SendMessage Lib "user32.dll" Alias "SendMessageA" ( _
ByVal hWnd As Long, _
ByVal wMsg As Long, _
ByVal wParam As Long, _
ByVal lParam As Long) As Long

Private Declare Function SendMessage_2 Lib "user32.dll" Alias "SendMessageA" ( _
ByVal hWnd As Long, _
ByVal wMsg As Long, _
ByVal wParam As Long, _
ByRef lParam As CAPSTATUS) As Long

Private Declare Function SetWindowPos Lib "user32.dll" ( _
ByVal hWnd As Long, _
ByVal hWndInsertAfter As Long, _
ByVal X As Long, _
ByVal Y As Long, _
ByVal cx As Long, _
ByVal cy As Long, _
ByVal wFlags As Long) As Long

Private Sub cmdClose_Click()
    TimerCameraImage.Enabled = False
    Unload Me
End Sub

Private Sub cmdInicialisation_Click()

    Call Inicializimi

    Dim Inicialisation As Integer
    Inicialisation = InitStp()
```

```
End Sub

Private Sub cmdStop13_Click()
    ' Ndalim i motorit 1
    NdaloLevizjenMotori 1, 1
End Sub

Private Sub cmdStop24_Click()
    ' Ndalim i motorit 2
    NdaloLevizjenMotori 2, 1
End Sub

Private Sub Inicializimi()
    Dim pCS As CAPSTATUS, ret As Long, mcistr As String
    Dim res As Long, pich As Long

    OptionMode(0).Value = True   ' Kur sistemi eshte ne mode manuale
    OptionMode(1).Value = False  ' Kur sistemi eshte ne mode jo manuale (automatike)


    valRreshtiQenderObjekti.Caption = 0
    valKolonaQenderObjekti.Caption = 0

    valRreshtiQenderImazhi.Caption = 0
    valKolonaQenderImazhi.Caption = 0

    lDistanca_QenderObjekti_QenderImazhi.Caption = 0

    txtStep.Text = 10
    txtInterval.Text = 30
    txtStepsMotor.Text = 10

    txtEpsilon.Text = 4

    hCapWin = capCreateCaptureWindow("CaptureWindow", WS_CHILD Or WS_VISIBLE, 0, 0, 640, 480,
PicCam.hWnd, 0)
    SendMessage hCapWin, WM_CAP_DRIVER_DISCONNECT, 0, 0

    hCapWin = capCreateCaptureWindow("CaptureWindow", WS_CHILD Or WS_VISIBLE, 0, 0, 640, 480,
PicCam.hWnd, 0)
    If Not hCapWin = 0 Then
        For res = 0 To 9   'find webcam all examples i found used 0 but mine was on 1
            ret = SendMessage(hCapWin, WM_CAP_DRIVER_CONNECT, res, 0)
            If ret = 1 Then Exit For
        Next
        If ret = 0 Then MsgBox "no webcam found", vbCritical: Exit Sub

         SendMessage hCapWin, WM_CAP_DLG_VIDEOSOURCE, 0, 0  ' select webcam or other settings from
dialog
         SendMessage hCapWin, WM_CAP_SET_PREVIEWRATE, 66, 0
         SendMessage hCapWin, WM_CAP_SET_PREVIEW, True, 0
         SendMessage_2 hCapWin, WM_CAP_GET_STATUS, Len(pCS), pCS

         w = pCS.uiImageWidth
         h = pCS.uiImageHeight
         SetWindowPos hCapWin, HWND_TOP, 0, 0, w, h, SWP_NOMOVE Or SWP_NOZORDER Or SWP_NOENDCHANGING

         TimerCameraImage.Interval = 100
         TimerCameraImage.Enabled = True
    End If
End Sub


Private Sub Command1_Click()
```

```
    SaveSetting "PANELI", "Pixel", "PerHapa", txtPixelPerHap.Text
End Sub

Private Sub Form_Load()
    txtPixelPerHap.Text = GetSetting("PANELI", "Pixel", "PerHapa", 4)
End Sub

Private Sub Form_Terminate()
    SendMessage hCapWin, WM_CAP_DRIVER_DISCONNECT, 0, 0
End Sub

Private Sub img1_Click()
    ' Rrotullimi para
    FilloLevizjenMotori 1, txtStepsMotor.Text, txtInterval.Text, 0, 1
End Sub

Private Sub img1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    img1.BorderStyle = 1
End Sub

Private Sub img1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    img1.BorderStyle = 0
End Sub

Private Sub img2_Click()
    ' Rrotullimi djathtas
    FilloLevizjenMotori 2, txtStepsMotor.Text, txtInterval.Text, 0, 1
End Sub

Private Sub img2_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    img2.BorderStyle = 1
End Sub

Private Sub img2_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    img2.BorderStyle = 0
End Sub

Private Sub img3_Click()
    ' Rrotullimi prapa
    FilloLevizjenMotori 1, txtStepsMotor.Text, txtInterval.Text, 1, 1
End Sub

Private Sub img3_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    img3.BorderStyle = 1
End Sub

Private Sub img3_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    img3.BorderStyle = 0
End Sub

Private Sub img4_Click()
    ' Rrotullimi majtas
    FilloLevizjenMotori 2, txtStepsMotor.Text, txtInterval.Text, 1, 1
End Sub

Private Sub img4_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
    img4.BorderStyle = 1
End Sub

Private Sub img4_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    img4.BorderStyle = 0
End Sub


Private Sub TimerCameraImage_Timer()
```

```
    Dim temp
    Dim LlogaritKordinataXY As KordinataXY

    If OptionMode(1).Value = True And IsNumericAndMore1(txtInterval.Text) Then      ' Kur sistemi
eshte ne mode jo manuale (automatike)

        TimerCameraImage.Enabled = False

        'Get Current Frame
        SendMessage hCapWin, GET_FRAME, 0, 0
        'Copy Current Frame to ClipBoard
        SendMessage hCapWin, COPY, 0, 0
        'Put ClipBoard's Data to picOutput
        Picture1.Picture = Clipboard.GetData

        SavePicture Picture1.Image, App.Path & "\ImWeb.bmp"

        LlogaritKordinataXY = GjejPozicioninObjektit(Picture1, txtStep.Text)

        valRreshtiQenderObjekti.Caption = LlogaritKordinataXY.NrRreshtitQenderObjekti
        valKolonaQenderObjekti.Caption = LlogaritKordinataXY.NrKolonesQenderObjekti

        valRreshtiQenderImazhi.Caption = LlogaritKordinataXY.NrRreshtitQenderImazhi
        valKolonaQenderImazhi.Caption = LlogaritKordinataXY.NrKolonesQenderImazhi

        lDistanca_QenderObjekti_QenderImazhi.Caption =
LlogaritKordinataXY.Distanca_QenderObjekti_QenderImazhi

        With LlogaritKordinataXY

        'Pozicionojme objektin ne figure
        If (.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi) > 0 Then
            'Objekti ne kuadratet 3, 4

            If (.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) > 0 Then
                'Objekti ne kuadrati e 4_t
                '''''''''''''''''''''''''''

                FilloLevizjenMotori 1, CInt(Abs(.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi)
/ txtPixelPerHap.Text), txtInterval.Text, 0, 1

                FilloLevizjenMotori 2, CInt(Abs(.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) /
txtPixelPerHap.Text), txtInterval.Text, 1, 1


            ElseIf (.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) < 0 Then
                'Objekti ne kuadrati e 3_t
                '''''''''''''''''''''''''''
                FilloLevizjenMotori 1, CInt(Abs(.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi)
/ txtPixelPerHap.Text), txtInterval.Text, 0, 1

                FilloLevizjenMotori 2, CInt(Abs(.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) /
txtPixelPerHap.Text), txtInterval.Text, 0, 1

            Else
                'Objekti ne vijen ndarese te kuadrateve te 3_te dhe te 4_t.
                '''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                FilloLevizjenMotori 1, CInt(Abs(.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi)
/ txtPixelPerHap.Text), txtInterval.Text, 0, 1

            End If

        ElseIf (.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi) < 0 Then
            'Objekti ne kuadratet 1, 2
```

```
            If (.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) > 0 Then
                'Objekti ne kuadrati e 2_te
                ''''''''''''''''''''''''''''
                FilloLevizjenMotori 1, CInt(Abs(.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi)
/ txtPixelPerHap.Text), txtInterval.Text, 1, 1

                FilloLevizjenMotori 2, CInt(Abs(.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) /
txtPixelPerHap.Text), txtInterval.Text, 0, 1

            ElseIf (.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) < 0 Then
                'Objekti ne kuadrati e 1_re
                ''''''''''''''''''''''''''''
                FilloLevizjenMotori 1, CInt(Abs(.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi)
/ txtPixelPerHap.Text), txtInterval.Text, 1, 1

                FilloLevizjenMotori 2, CInt(Abs(.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) /
txtPixelPerHap.Text), txtInterval.Text, 1, 1

            Else
                'Objekti ne vijen ndarese te kuadrateve te 1_re dhe te 2_te.
                ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                FilloLevizjenMotori 1, CInt(Abs(.NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi)
/ txtPixelPerHap.Text), txtInterval.Text, 1, 1

            End If

        Else
             'Objekti ne vijen ndarese ku :  .NrRreshtitQenderObjekti - .NrRreshtitQenderImazhi = 0.

            If (.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) > 0 Then
                'Objekti ne vijen ndarese te kuadrateve te 1_re dhe te 4_te.
                ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                FilloLevizjenMotori 2, CInt(Abs(.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) /
txtPixelPerHap.Text), txtInterval.Text, 1, 1

            ElseIf (.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) < 0 Then
                'Objekti ne vijen ndarese te kuadrateve te 2_te dhe te 3_te.
                ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                FilloLevizjenMotori 2, CInt(Abs(.NrKolonesQenderObjekti - .NrKolonesQenderImazhi) /
txtPixelPerHap.Text), txtInterval.Text, 0, 1

            Else
                'Objekti eshte ne vijen qender te figures.
                ''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
                'Nuk ka veprime per te bere.
            End If

        End If


        End With

        TimerCameraImage.Enabled = True

    End If

End Sub

Private Function GjejPozicioninObjektit(ByVal picColor As PictureBox, HapiKercimit As Integer) As
KordinataXY

    Dim Pozicioni As KordinataXY

    Const pixR As Integer = 3
    Const pixG As Integer = 2
```

```
    Const pixB As Integer = 1


    Dim bitmap_info                 As BITMAPINFO

    Dim pixels()                    As Byte

    Dim bytes_per_scanLine                  As Long
    Dim pad_per_scanLine                    As Integer
    Dim X                                   As Integer
    Dim Y                                   As Integer

    Dim lartesia, gjeresia                  As Integer
    Dim rreshti                             As Integer
    Dim kolona                              As Integer
    Dim VleraMaxKrahasuese                  As Integer

    Dim koeficient_radian_pixel             As Double
    Dim koeficient_kompesimi_gabimi         As Double

    Dim Pozicioni_rreshtiA                  As Integer
    Dim Pozicioni_kolonaA                   As Integer

    Dim Pozicioni_rreshtiB                  As Integer
    Dim Pozicioni_kolonaB                   As Integer

    Dim Pozicioni_rreshtiC                  As Integer
    Dim Pozicioni_kolonaC                   As Integer

    Dim Pozicioni_rreshtiD                  As Integer
    Dim Pozicioni_kolonaD                   As Integer

    Dim Pozicioni_rreshtiQenderObjekti      As Integer
    Dim Pozicioni_kolonaQenderObjekti       As Integer

    Dim Pixels_Prej_Qendres                 As Double


    With bitmap_info.bmiHeader
        .biSize = 40
        .biWidth = picColor.ScaleWidth
        ' Use negative height to scan top-down.
        .biHeight = -picColor.ScaleHeight
        .biPlanes = 1
        .biBitCount = 32
        .biCompression = BI_RGB
        bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \ 32) * 4)
        pad_per_scanLine = bytes_per_scanLine - (((.biWidth * .biBitCount) + 7) \ 8)
        .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
    End With

    VleraMaxKrahasuese = 0

    ' Load the bitmap's data.
    ReDim pixels(1 To 4, 1 To picColor.ScaleWidth, 1 To picColor.ScaleHeight)
    GetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1), bitmap_info,
DIB_RGB_COLORS

    lartesia = picColor.Height / Screen.TwipsPerPixelY
    gjeresia = picColor.Width / Screen.TwipsPerPixelX


    VleraMaxKrahasuese = 0

    ' Kodet per perpunimin e imazhit
```

```
    ' gjetjea e pikes se pare te zhdriteshme
    For rreshti = 1 To lartesia - 1 Step HapiKercimit
        For kolona = 1 To gjeresia - 1 Step HapiKercimit
            If pixels(pixR, kolona, rreshti) > VleraMaxKrahasuese Then
                VleraMaxKrahasuese = pixels(pixR, kolona, rreshti)
                Pozicioni_rreshtiA = rreshti
                Pozicioni_kolonaA = kolona
            End If
        Next
    Next

    ' gjetja e pikes se dyte te zhdriteshme duke levizur sipas kolonave
    Pozicioni_rreshtiC = Pozicioni_kolonaA

    For kolona = Pozicioni_kolonaA + 1 To gjeresia - 1 Step HapiKercimit
        If pixels(pixR, kolona, Pozicioni_rreshtiC) = VleraMaxKrahasuese Then
            Pozicioni_kolonaC = kolona
        Else
            Exit For
        End If
    Next

    ' gjetja e pikes se dyte te zhdriteshme duke levizur sipas reshtave
    Pozicioni_kolonaB = Pozicioni_kolonaA

    For rreshti = Pozicioni_rreshtiA + 1 To lartesia - 1 Step HapiKercimit
        If pixels(pixR, Pozicioni_kolonaB, rreshti) = VleraMaxKrahasuese Then
            Pozicioni_rreshtiB = rreshti
        Else
            Exit For
        End If
    Next

    ' gjetja e pikes se qendres se zhdriteshme duke shfrytezuar pikat A, B, C
    Pozicioni_rreshtiQenderObjekti = Pozicioni_rreshtiC - Pozicioni_rreshtiA
    Pozicioni_kolonaQenderObjekti = Pozicioni_kolonaB - Pozicioni_kolonaA

    ' Llogaritim qendren e imazhit
    Dim x_gender, y_gender
    x_gender = CInt(gjeresia / 2)
    y_gender = CInt(lartesia / 2)

    ' Llogaritim largesine e qendres se objektit rrethor (diellit) prej qendres
    Pixels_Prej_Qendres = Sqr((y_gender - Pozicioni_rreshtiQenderObjekti) ^ 2 + (x_gender -
Pozicioni_kolonaQenderObjekti) ^ 2)

    Pozicioni.Distanca_QenderObjekti_QenderImazhi = Pixels_Prej_Qendres

    ' Vlereso pozicionin e qendres se objektit
    Pozicioni.NrRreshtitQenderObjekti = Pozicioni_rreshtiQenderObjekti
    Pozicioni.NrKolonesQenderObjekti = Pozicioni_kolonaQenderObjekti
    ' Vlereso pozicionin e qendres se imazhit
    Pozicioni.NrRreshtitQenderImazhi = y_gender
    Pozicioni.NrKolonesQenderImazhi = x_gender

    If Pozicioni_kolonaQenderObjekti > 10 And _
       Pozicioni_kolonaQenderObjekti < gjeresia - 10 And _
       Pozicioni_rreshtiQenderObjekti > 10 And _
       Pozicioni_rreshtiQenderObjekti < lartesia - 10 Then
        ' Vizato nje vije te kuqe vertikale
        For rreshti = Pozicioni_rreshtiQenderObjekti - 10 To Pozicioni_rreshtiQenderObjekti + 10
            pixels(pixR, Pozicioni_kolonaQenderObjekti, rreshti) = 255
        Next
```

```
        ' Vizato nje vije te kuqe horizontale
        For kolona = Pozicioni_kolonaQenderObjekti - 10 To Pozicioni_kolonaQenderObjekti + 10
            pixels(pixR, kolona, Pozicioni_rreshtiQenderObjekti) = 255
        Next

        picColor.Circle (Pozicioni_rreshtiQenderObjekti, Pozicioni_kolonaQenderObjekti), 5, 255

        ' Shfaq rezulatin e perpunimit grafik.
        SetDIBits picColor.hdc, picColor.Image, 0, picColor.ScaleHeight, pixels(1, 1, 1),
bitmap_info, DIB_RGB_COLORS

        picColor.Picture = picColor.Image
    End If

    GjejPozicioninObjektit = Pozicioni
End Function

''''''''''''''''''''''''''''''''''''''''
' Hapat e analizes grafike
''''''''''''''''''''''''''''''''''''''''
Private Sub UpDown1_DownClick()
    If CInt(txtStep.Text) - 1 > 0 Then
       txtStep.Text = CInt(txtStep.Text) - 1
    End If
End Sub

Private Sub UpDown1_UpClick()
    If txtStep.Text = "" Then txtStep.Text = "1"
    txtStep.Text = CInt(txtStep.Text) + 1
End Sub

''''''''''''''''''''''''''''''''''''''''
' "Epsiloni" Gabimi i lejuar
''''''''''''''''''''''''''''''''''''''''
Private Sub UpDown2_DownClick()
    If CInt(txtEpsilon.Text) - 1 > 0 Then
       txtEpsilon.Text = CInt(txtEpsilon.Text) - 1
    End If
End Sub

Private Sub UpDown2_UpClick()
    If txtEpsilon.Text = "" Then txtEpsilon.Text = "1"
    txtEpsilon.Text = CInt(txtEpsilon.Text) + 1
End Sub


''''''''''''''''''''''''''''''''''''''''
' Percaktimi i intervalit
''''''''''''''''''''''''''''''''''''''''
Private Sub UpDownInterval_DownClick()
    If CInt(txtInterval.Text) - 10 > 0 Then
       txtInterval.Text = CInt(txtInterval.Text) - 10
    End If
End Sub

Private Sub UpDownInterval_UpClick()
    txtInterval.Text = CInt(txtInterval.Text) + 10
End Sub

''''''''''''''''''''''''''''''''''''''''
' Percaktimi i hapave qe do te bejne motoret
''''''''''''''''''''''''''''''''''''''''
Private Sub UpDownSteps_DownClick()
    If CInt(txtStepsMotor.Text) - 10 > 0 Then
       txtStepsMotor.Text = CInt(txtStepsMotor.Text) - 10
```

```
    End If
End Sub

Private Sub UpDownSteps_UpClick()
    txtStepsMotor.Text = CInt(txtStepsMotor.Text) + 10
End Sub

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Kontrolloj nese variabli 'intVal' eshte numeric dhe me i madh se 1
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
Private Function IsNumericAndMore1(intVal As String) As Boolean
    On Error Resume Next
    If IsNumeric(intVal) Then
       If CInt(intVal) > 1 Then
          IsNumericAndMore1 = True
       Else
          IsNumericAndMore1 = False
       End If
    Else
       IsNumericAndMore1 = False
    End If
    Err.Clear
End Function
```

modGlobals.bas

```
Option Explicit
Declare Function GetObject Lib "gdi32" Alias "GetObjectA" (ByVal hObject As Long, ByVal nCount As
Long, lpObject As Any) As Long
Declare Function GetDIBits Lib "gdi32" (ByVal aHDC As Long, ByVal hBitmap As Long, ByVal nStartScan
As Long, ByVal nNumScans As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
Declare Function SetDIBits Lib "gdi32" (ByVal hdc As Long, ByVal hBitmap As Long, ByVal nStartScan
As Long, ByVal nNumScans As Long, lpBits As Any, lpBI As BITMAPINFO, ByVal wUsage As Long) As Long
Type BITMAP '14 bytes
    bmType As Long
    bmWidth As Long
    bmHeight As Long
    bmWidthBytes As Long
    bmPlanes As Integer
    bmBitsPixel As Integer
    bmBits As Long
End Type
Type BITMAPINFOHEADER '40 bytes
    biSize As Long
    biWidth As Long
    biHeight As Long
    biPlanes As Integer
    biBitCount As Integer
    biCompression As Long
    biSizeImage As Long
    biXPelsPerMeter As Long
    biYPelsPerMeter As Long
    biClrUsed As Long
    biClrImportant As Long
End Type
Type RGBQUAD
    rgbBlue As Byte
    rgbGreen As Byte
    rgbRed As Byte
    rgbReserved As Byte
End Type
Type BITMAPINFO
    bmiHeader As BITMAPINFOHEADER
```

```
    bmiColors As RGBQUAD
End Type
Public Const DIB_RGB_COLORS = 0&
Public Const BI_RGB = 0&

Type KordinataXY
    NrRreshtitQenderObjekti   As Integer
    NrKolonesQenderObjekti    As Integer

    NrRreshtitQenderImazhi   As Integer
    NrKolonesQenderImazhi    As Integer

    Distanca_QenderObjekti_QenderImazhi As Integer
End Type

Type T_SPECIFIKIME
    NrRreshtit          As Integer
    NrKolones           As Integer
    GjeresiaPamjes      As Integer
    LartesiaPamjes      As Integer
    Distanca            As Double
End Type

Global Koeficienti_Sys_Lazer_Kamera        As Double

Type KONFIRMIM
    PamjaLeft       As Integer
    PamjaRight      As Integer
End Type
Global gKONFIRMIM As KONFIRMIM

' ' ' Parametrat kryesore te kamerave
Global DistancaNdermjet2Kamerave    As Double
Global gDistVatroreMajtas           As Double
Global gDistVatroreDjathtas         As Double
Global gDiametriMajtas              As Double
Global gDiametriDjathtas            As Double
```

modMain.bas

```
Option Explicit
Public Const WS_CHILD As Long = &H40000000
Public Const WS_VISIBLE As Long = &H10000000

Public Type POINT
X As Long
Y As Long
End Type

Public Type CAPSTATUS
uiImageWidth As Long
uiImageHeight As Long
fLiveWindow As Long
fOverlayWindow As Long
fScale As Long
ptScroll As POINT
fUsingDefaultPalette As Long
fAudioHardware As Long
fCapFileExists As Long
dwCurrentVideoFrame As Long
dwCurrentVideoFramesDropped As Long
dwCurrentWaveSamples As Long
dwCurrentTimeElapsedMS As Long
```

```
hPalCurrent As Long
fCapturingNow As Long
dwReturn As Long
wNumVideoAllocated As Long
wNumAudioAllocated As Long
End Type

Public Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias "capCreateCaptureWindowA" (
_
ByVal lpszWindowName As String, _
ByVal dwStyle As Long, _
ByVal X As Long, _
ByVal Y As Long, _
ByVal nWidth As Long, _
ByVal nHeight As Long, _
ByVal hWndParent As Long, _
ByVal nID As Long) As Long

Public Declare Function GetDC Lib "user32.dll" ( _
ByVal hWnd As Long) As Long

Private Declare Function CreateCompatibleBitmap Lib "gdi32.dll" ( _
ByVal hdc As Long, _
ByVal nWidth As Long, _
ByVal nHeight As Long) As Long

Public Declare Function CreateCompatibleDC Lib "gdi32.dll" ( _
ByVal hdc As Long) As Long

Public Declare Function SelectObject Lib "gdi32.dll" ( _
ByVal hdc As Long, _
ByVal hObject As Long) As Long

Public Declare Function BitBlt Lib "gdi32.dll" ( _
ByVal hDestDC As Long, _
ByVal X As Long, _
ByVal Y As Long, _
ByVal nWidth As Long, _
ByVal nHeight As Long, _
ByVal hSrcDC As Long, _
ByVal xSrc As Long, _
ByVal ySrc As Long, _
ByVal dwRop As Long) As Long

Public Declare Function DeleteDC Lib "gdi32.dll" ( _
ByVal hdc As Long) As Long
Public hCapWinLeft  As Long
Public hCapWinRight As Long
Public hCapWin      As Long
Public w As Long
Public h As Long

Sub Main()
    On Error Resume Next

    LoadProfile App.Path & "\kamera.ini"

    frmCamera.Show

    Err.Clear
End Sub

Public Function SaveProfile(ByVal FileName As String, SaveSettings As Boolean) As Boolean
   On Error Resume Next
   Kill FileName
```

```
   SaveProfile = False
   If SaveSettings Then
     ' WritePrivateProfileString "1Kamera", ".....k1", , FileName
     ' WritePrivateProfileString "1Kamera", ".....k2", , FileName
     ' WritePrivateProfileString "1Kamera", ".....k3", , FileName

      WritePrivateProfileString "2Kamera", ".....DistVatrMajtas", CStr(gDistVatroreMajtas), FileName
      WritePrivateProfileString "2Kamera", ".....DistVatrDjathtas", CStr(gDistVatroreDjathtas),
FileName
      WritePrivateProfileString "2Kamera", ".....DiametriMajtas", CStr(gDiametriMajtas), FileName
      WritePrivateProfileString "2Kamera", ".....DiametriDjathtas", CStr(gDiametriDjathtas),
FileName

      WritePrivateProfileString "2Kamera", ".....DistancaNdermjet2Kamerave",
CStr(DistancaNdermjet2Kamerave), FileName
      SaveProfile = True
   End If
   Err.Clear
End Function

Public Function LoadProfile(ByVal FileName As String) As Boolean
   Dim hws                          As String
   Dim tStr                         As String
   Dim Ctr                          As Integer, X As Integer, Pcnt As Integer
   Dim tst                          As String

   On Error Resume Next
   Ctr = FileLen(FileName)
   If Err.Number > 0 Then
      Err.Clear
      LoadProfile = False
      Exit Function
   End If

   On Error Resume Next
   LoadProfile = True
   If Ctr < 1 Then
      Exit Function
   End If

   gDistVatroreMajtas = CDbl(GetFromIni("2Kamera", ".....DistVatrMajtas", FileName))
   gDistVatroreDjathtas = CDbl(GetFromIni("2Kamera", ".....DistVatrDjathtas", FileName))
   gDiametriMajtas = CDbl(GetFromIni("2Kamera", ".....DiametriMajtas", FileName))
   gDiametriDjathtas = CDbl(GetFromIni("2Kamera", ".....DiametriDjathtas", FileName))

   DistancaNdermjet2Kamerave = CDbl(GetFromIni("2Kamera", ".....DistancaNdermjet2Kamerave",
FileName))

   Err.Clear
End Function
```

modProfile.bas

```
Declare Function GetPrivateProfileString Lib "kernel32" Alias "GetPrivateProfileStringA" _
           (ByVal lpApplicationName As String, _
            lpKeyName As Any, _
            ByVal lpDefault As String, _
            ByVal lpReturnedString As String, _
            ByVal nSize As Integer, _
            ByVal lpFileName As String) As Integer

Declare Function WritePrivateProfileString% Lib "kernel32" Alias "WritePrivateProfileStringA" _
           (ByVal lpApplicationName$, _
            ByVal lpKeyName As Any, _
            ByVal lpString As Any, _
```

```
            ByVal lpFileName$)


Public Function GetFromIni(strSectionHeader As String, _
                           strVariableName As String, _
                           strFileName As String) _
                           As String

   Dim strReturn                        As String

   strReturn = VBA.String(255, VBA.Chr(0))
   GetFromIni = VBA.Left$(strReturn, _
                GetPrivateProfileString(strSectionHeader, ByVal strVariableName, "", _
                strReturn, Len(strReturn), strFileName))

End Function
```

modStepperBee.bas

```
Option Explicit
Declare Function InitStp Lib "stp.dll" () As Integer
Declare Function RunMotor1 Lib "stp.dll" (ByVal Steps As Integer, _
                                          ByVal Interval As Integer, _
                                          ByVal Direction As Integer, _
                                          ByVal Outputs As Integer) As Boolean

Declare Function RunMotor2 Lib "stp.dll" (ByVal Steps As Integer, _
                                          ByVal Interval As Integer, _
                                          ByVal Direction As Integer, _
                                          ByVal Outputs As Integer) As Boolean
Declare Function StopMotor1 Lib "stp.dll" (ByVal Outputs As Integer) As Boolean

Declare Function StopMotor2 Lib "stp.dll" (ByVal Outputs As Integer) As Boolean

Declare Function SetStepMode Lib "stp.dll" (ByVal M1Mode As Integer, ByVal M2Mode As Integer) As
Boolean

Declare Function GetStatus Lib "stp.dll" (ByRef M1Active As Integer, _
                                          ByRef M2Active As Integer, _
                                          ByRef M1Steps As Integer, _
                                          ByRef M2Steps As Integer, _
                                          ByRef Inputs As Integer) As Boolean

 ' Fillim i levizjes - Motor 1
Public Function FilloLevizjenMotori(ByVal NumberMotor As Integer, _
                                    ByVal Steps As Integer, _
                                    ByVal Interval As Integer, _
                                    ByVal Direction As Integer, _
                                    ByVal Outputs As Integer) As Boolean
    Select Case NumberMotor
    Case 1
       ' Startim i motorit 1
       FilloLevizjenMotori = RunMotor1(Steps, Interval, Direction, Outputs)
    Case 2
       ' Startim i motorit 2
       FilloLevizjenMotori = RunMotor2(Steps, Interval, Direction, Outputs)
    Case Else
    End Select
End Function

 ' Fillim i levizjes - Motor 1
Public Function NdaloLevizjenMotori(ByVal NumberMotor As Integer, ByVal Outputs As Integer) As
Boolean
    Select Case NumberMotor
```

```
    Case 1
       ' Startim i motorit 1
       NdaloLevizjenMotori = StopMotor1(Outputs)
    Case 2
       ' Startim i motorit 2
       NdaloLevizjenMotori = StopMotor2(Outputs)
    Case Else
    End Select
End Function
```

Shtojca 9

Në vazhdim po japim librarinë e komandimit të lëvizjeve të robotit në Visual Basic .NET bazuar në librarinë .NET System.IO.Ports të Microsoft.

frmMain.vb (GUI)

```
Option Strict On

Imports System.Text
Imports System.Threading
Imports System.IO.Ports


Public Class frmMain
    Inherits System.Windows.Forms.Form

    Private shenja As Integer = -1
    Private m_IsIRobotCreateFound As Boolean = False
    Private gNumberOfPort As Integer = 0
    Friend WithEvents ButtonDistance As System.Windows.Forms.Button
    Friend WithEvents iPortText As System.Windows.Forms.TextBox
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents btnBatteri As System.Windows.Forms.Button
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents cmbBaud As System.Windows.Forms.ComboBox
    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents cmdClosePort As System.Windows.Forms.Button

    'Private comBuffer As Byte()
    Private comBuffer As String
    Friend WithEvents cmdSensore As System.Windows.Forms.Button
    Private Delegate Sub UpdateFormDelegate()
    Private UpdateFormDelegate1 As UpdateFormDelegate

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

        ' So that we only need to set the title of the application once,
        ' we use the AssemblyInfo class (defined in the AssemblyInfo.vb file)
        ' to read the AssemblyTitle attribute.
        Dim ainfo As New AssemblyInfo()

        Me.Text = ainfo.Title
        Me.mnuAbout.Text = String.Format("&About {0} ...", ainfo.Title)

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
```

```
        MyBase.Dispose(disposing)
    End Sub

    Private components As System.ComponentModel.IContainer

    Friend WithEvents mnuMain As System.Windows.Forms.MainMenu
    Friend WithEvents mnuFile As System.Windows.Forms.MenuItem
    Friend WithEvents mnuExit As System.Windows.Forms.MenuItem
    Friend WithEvents mnuHelp As System.Windows.Forms.MenuItem
    Friend WithEvents mnuAbout As System.Windows.Forms.MenuItem
    Friend WithEvents txtStatus As System.Windows.Forms.TextBox
    Friend WithEvents tmrReadCommPort As System.Windows.Forms.Timer
    Friend WithEvents btnCheckForPorts As System.Windows.Forms.Button
    Friend WithEvents btnCheckPort As System.Windows.Forms.Button
    Friend WithEvents txtSelectedModem As System.Windows.Forms.TextBox
    Friend WithEvents btnMoveCommand As System.Windows.Forms.Button
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.components = New System.ComponentModel.Container()
        Dim resources As System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(frmMain))
        Me.mnuMain = New System.Windows.Forms.MainMenu(Me.components)
        Me.mnuFile = New System.Windows.Forms.MenuItem()
        Me.mnuExit = New System.Windows.Forms.MenuItem()
        Me.mnuHelp = New System.Windows.Forms.MenuItem()
        Me.mnuAbout = New System.Windows.Forms.MenuItem()
        Me.btnCheckForPorts = New System.Windows.Forms.Button()
        Me.txtStatus = New System.Windows.Forms.TextBox()
        Me.tmrReadCommPort = New System.Windows.Forms.Timer(Me.components)
        Me.btnCheckPort = New System.Windows.Forms.Button()
        Me.txtSelectedModem = New System.Windows.Forms.TextBox()
        Me.btnMoveCommand = New System.Windows.Forms.Button()
        Me.cmdClosePort = New System.Windows.Forms.Button()
        Me.ButtonDistance = New System.Windows.Forms.Button()
        Me.iPortText = New System.Windows.Forms.TextBox()
        Me.Label1 = New System.Windows.Forms.Label()
        Me.btnBatteri = New System.Windows.Forms.Button()
        Me.Button1 = New System.Windows.Forms.Button()
        Me.cmbBaud = New System.Windows.Forms.ComboBox()
        Me.Label2 = New System.Windows.Forms.Label()
        Me.cmdSensore = New System.Windows.Forms.Button()
        Me.SuspendLayout()
        '
        'mnuMain
        '
        Me.mnuMain.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.mnuFile, Me.mnuHelp})
        '
        'mnuFile
        '
        Me.mnuFile.Index = 0
        Me.mnuFile.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.mnuExit})
        resources.ApplyResources(Me.mnuFile, "mnuFile")
        '
        'mnuExit
        '
        Me.mnuExit.Index = 0
        resources.ApplyResources(Me.mnuExit, "mnuExit")
        '
        'mnuHelp
        '
        Me.mnuHelp.Index = 1
        Me.mnuHelp.MenuItems.AddRange(New System.Windows.Forms.MenuItem() {Me.mnuAbout})
        resources.ApplyResources(Me.mnuHelp, "mnuHelp")
        '
        'mnuAbout
        '
```

```
Me.mnuAbout.Index = 0
resources.ApplyResources(Me.mnuAbout, "mnuAbout")
'
'btnCheckForPorts
'
resources.ApplyResources(Me.btnCheckForPorts, "btnCheckForPorts")
Me.btnCheckForPorts.Name = "btnCheckForPorts"
'
'txtStatus
'
resources.ApplyResources(Me.txtStatus, "txtStatus")
Me.txtStatus.BorderStyle = System.Windows.Forms.BorderStyle.None
Me.txtStatus.Name = "txtStatus"
Me.txtStatus.ReadOnly = True
'
'tmrReadCommPort
'
'
'btnCheckPort
'
resources.ApplyResources(Me.btnCheckPort, "btnCheckPort")
Me.btnCheckPort.Name = "btnCheckPort"
'
'txtSelectedModem
'
resources.ApplyResources(Me.txtSelectedModem, "txtSelectedModem")
Me.txtSelectedModem.Name = "txtSelectedModem"
'
'btnMoveCommand
'
resources.ApplyResources(Me.btnMoveCommand, "btnMoveCommand")
Me.btnMoveCommand.Name = "btnMoveCommand"
'
'cmdClosePort
'
resources.ApplyResources(Me.cmdClosePort, "cmdClosePort")
Me.cmdClosePort.Name = "cmdClosePort"
'
'ButtonDistance
'
Me.ButtonDistance.AutoEllipsis = True
resources.ApplyResources(Me.ButtonDistance, "ButtonDistance")
Me.ButtonDistance.Name = "ButtonDistance"
'
'iPortText
'
resources.ApplyResources(Me.iPortText, "iPortText")
Me.iPortText.Name = "iPortText"
'
'Label1
'
resources.ApplyResources(Me.Label1, "Label1")
Me.Label1.Name = "Label1"
'
'btnBatteri
'
resources.ApplyResources(Me.btnBatteri, "btnBatteri")
Me.btnBatteri.Name = "btnBatteri"
Me.btnBatteri.UseVisualStyleBackColor = True
'
'Button1
'
resources.ApplyResources(Me.Button1, "Button1")
Me.Button1.Name = "Button1"
Me.Button1.UseVisualStyleBackColor = True
```

```
        '
        'cmbBaud
        '
        Me.cmbBaud.FormattingEnabled = True
        resources.ApplyResources(Me.cmbBaud, "cmbBaud")
        Me.cmbBaud.Name = "cmbBaud"
        '
        'Label2
        '
        resources.ApplyResources(Me.Label2, "Label2")
        Me.Label2.Name = "Label2"
        '
        'cmdSensore
        '
        resources.ApplyResources(Me.cmdSensore, "cmdSensore")
        Me.cmdSensore.Name = "cmdSensore"
        Me.cmdSensore.UseVisualStyleBackColor = True
        '
        'frmMain
        '
        resources.ApplyResources(Me, "$this")
        Me.Controls.Add(Me.cmdSensore)
        Me.Controls.Add(Me.Label2)
        Me.Controls.Add(Me.cmbBaud)
        Me.Controls.Add(Me.Button1)
        Me.Controls.Add(Me.btnBatteri)
        Me.Controls.Add(Me.Label1)
        Me.Controls.Add(Me.iPortText)
        Me.Controls.Add(Me.ButtonDistance)
        Me.Controls.Add(Me.cmdClosePort)
        Me.Controls.Add(Me.btnMoveCommand)
        Me.Controls.Add(Me.txtSelectedModem)
        Me.Controls.Add(Me.btnCheckPort)
        Me.Controls.Add(Me.txtStatus)
        Me.Controls.Add(Me.btnCheckForPorts)
        Me.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle
        Me.Menu = Me.mnuMain
        Me.Name = "frmMain"
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub

#End Region

#Region " Standard Menu Code "
    ' <System.Diagnostics.DebuggerStepThrough()> has been added to some procedures since they are
    ' not the focus of the demo. Remove them if you wish to debug the procedures.
    ' This code simply shows the About form.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub mnuAbout_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles mnuAbout.Click
        ' Open the About form in Dialog Mode
        Dim frm As New frmAbout()
        frm.ShowDialog(Me)
        frm.Dispose()
    End Sub

    ' This code will close the form.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub mnuExit_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles mnuExit.Click
        ' Close the current form
        Me.Close()
    End Sub
#End Region
```

```
    ' This subroutine checks for available ports on the local machine. It does
    '   this by attempting to open ports 1 through 4.
    Private Sub btnCheckForPorts_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnCheckForPorts.Click

        ' Check for Availability of each of the 4 Comm Ports, and
        '   place a check in the list box items that have openable ports.
        WriteMessage("Testimi COM" + iPortText.Text)
        If IsPortAvailable(CInt(iPortText.Text)) Then
            gNumberOfPort = CInt(iPortText.Text)
            Me.btnCheckPort.Enabled = True
        Else
            Me.btnCheckPort.Enabled = False
        End If


    End Sub

    ' Kjo subroutine kontrollon per portat seriale aktive.
    Private Sub btnCheckPort_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnCheckPort.Click

        If gNumberOfPort > 0 Then

            If IsPortAvailable(gNumberOfPort) Then
                If IsPortAIRobotCreate(gNumberOfPort) Then
                    Me.m_IsIRobotCreateFound = True
                    m_IRobotCreate = gNumberOfPort
                    ' Mesazh per perdoruesin.
                    WriteMessage("Port " + (gNumberOfPort).ToString() + _
                        " is a responsive iRobot Ceate.")
                Else
                    ' Mesazh per perdoruesin.
                    WriteMessage("Port " + (gNumberOfPort).ToString() + _
                        " is not a responsive iRobot Ceate.")
                End If
            End If
        End If

        ' If a IRobotCreate was found, enable the rest of the buttons, so the user
        '   can interact with the modem.
        If Me.m_IsIRobotCreateFound Then
            Me.txtSelectedModem.Text = "iRobot Create on COM" + _
                m_IRobotCreate.ToString()
            Me.btnMoveCommand.Enabled = True
        End If
    End Sub


    Private Sub btnMoveCommand_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnMoveCommand.Click
        Dim sDistanca As Double

        Try
            ' Enable the timer.
            Me.tmrReadCommPort.Enabled = True

            sDistanca = DistanceSensorRoomba(m_IRobotCreate)
            Debug.Print(CStr(sDistanca))

            shenja = shenja * (-1)

            SetFwdVelRadiusRoomba(m_IRobotCreate, shenja * 0.7, 2)
```

```
            Thread.Sleep(1000)

            SetFwdVelRadiusRoomba(m_IRobotCreate, 0, 1)

            sDistanca = DistanceSensorRoomba(m_IRobotCreate)
            Debug.Print(CStr(sDistanca))

            Application.DoEvents()
        Catch ex As Exception
            'Mesazh per perdoruesin.
            MessageBox.Show("Komunikim i pamundur me iRobot Create")
        Finally
            ' Disable the timer.
            Me.tmrReadCommPort.Enabled = False
        End Try

    End Sub


    ' This subroutine writes a message to the txtStatus TextBox.
    Public Sub WriteMessage(ByVal message As String)
        Me.txtStatus.Text += message + vbCrLf
    End Sub

    ' This subroutine writes a message to the txtStatus TextBox and allows
    '   the line feed to be suppressed.
    Private Sub WriteMessage(ByVal message As String, ByVal linefeed As Boolean)
        Me.txtStatus.Text += message
        If linefeed Then
            Me.txtStatus.Text += vbCrLf
        End If
    End Sub

    Private Sub frmMain_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        cmbBaud.Items.Clear()
        cmbBaud.Items.Add("19200")
        cmbBaud.Items.Add("57600")
        cmbBaud.SelectedIndex = 1

        '' AddHandler m_CommPort.DataReceived, AddressOf mySerialPort_DataReceived
    End Sub

    '  Public Sub mySerialPort_DataReceived(ByVal sender As Object, ByVal e As
SerialDataReceivedEventArgs)
    ' 'Handles serial port data received events
    '     UpdateFormDelegate1 = New UpdateFormDelegate(AddressOf UpdateDisplay)
    ' Dim n As Integer = m_CommPort.BytesToRead 'find number of bytes in buf
    '     comBuffer = New Byte(n - 1) {} 're dimension storage buffer
    '     m_CommPort.Read(comBuffer, 0, n) 'read data from the buffer
    '     Me.Invoke(UpdateFormDelegate1) 'call the delegate
    ' End Sub


    Public Sub mySerialPort_DataReceived(ByVal sender As Object, ByVal e As
SerialDataReceivedEventArgs)
        On Error Resume Next
        'Handles serial port data received events
        UpdateFormDelegate1 = New UpdateFormDelegate(AddressOf UpdateDisplay)

        Dim n As Integer = m_CommPort.BytesToRead 'find number of bytes in buffer
        If n > 0 Then
            comBuffer = m_CommPort.ReadLine()
        End If
        Me.Invoke(UpdateFormDelegate1) 'call the delegate
```

```
    End Sub

    Private Sub UpdateDisplay()
        Me.txtStatus.Text += comBuffer + vbCrLf
    End Sub

    Private Sub cmdClosePort_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdClosePort.Click
        m_CommPort.Close()
        Me.txtStatus.Clear()
    End Sub

    Private Sub ButtonDistance_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles ButtonDistance.Click
        Dim sDistanca As Double
        sDistanca = DistanceSensorRoomba(m_IRobotCreate)
        MsgBox(sDistanca)
    End Sub

    Private Sub btnBatteri_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnBatteri.Click
        Dim a As String
        a = BatteryVoltageRoomba()

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        Dim ssByte As Byte
        Select Case cmbBaud.SelectedIndex
            Case 0
                ssByte = 7
            Case 1
                ssByte = 10
        End Select
        ChangeBaudRateRoomba(ssByte)
    End Sub

    Private Sub cmdSensore_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
cmdSensore.Click
        AllSensorsReadRoomba()
    End Sub

End Class
```

modLibrary_iRobotCreate_VBNET

```
Imports System.IO.Ports

Module modLibrary_iRobotCreate_VBNET

    Structure Sensore
        Private BumpRight As String
        Private BumpLeft As String
        Private BumpFront As String
        Private Wall As String
        Private virtWall As String
        Private CliffLft As String
        Private CliffRgt As String
        Private CliffFrntLft As String
        Private CliffFrntRgt As String
        Private LeftCurrOver As String
        Private RightCurrOver As String
        Private DirtL As String
        Private DirtR As String
```

```
    Private ButtonPlay As String
    Private ButtonAdv As String
    Private Dist As String
    Private Angle As String
    Private Volts As String
    Private Current As String
    Private Temp As String
    Private Charge As String
    Private Capacity As String
    Private pCharge As String
End Structure

Public Const td As Integer = 15         'System.Threading.Thread.Sleep(td)
Public Const Pi As Double = 3.1416
Public Const Eps As Double = 0.00000000000000022204

Public MessageBack As String

' Declare necessary class variables.
Public m_CommPort As New SerialPort
Public m_IRobotCreate As Integer = 0

Public Function HIBYTE(ByVal Intg As Integer) As Byte
    HIBYTE = CByte((Intg And &HFF00&) / 256)
End Function

Public Function LOBYTE(ByVal Intg As Integer) As Byte
    LOBYTE = CByte(Intg And &HFF&)
End Function

Public Function UnHex(ByVal sHex As String) As Long
    UnHex = CLng(Val("&H" & sHex))
End Function

Public Function Min(ByVal Val1 As Double, ByVal Val2 As Double) As Double
    If Val1 > Val2 Then
        Min = Val2
    Else
        Min = Val1
    End If
End Function

Public Function Max(ByVal Val1 As Double, ByVal Val2 As Double) As Double
    If Val1 > Val2 Then
        Max = Val1
    Else
        Max = Val2
    End If
End Function

Public Function isinf(ByVal var As Double) As Integer
    If var > 1.0E+100 Or var < -1.0E+100 Then
        isinf = 1
    Else
        isinf = 0
    End If
End Function


' This function attempts to open the passed Comm Port. If it is
'   available, it returns True, else it returns False. To determine
'   availability a Try-Catch block is used.
Public Function IsPortAvailable(ByVal ComPort As Integer) As Boolean
    Try
```

```
        With m_CommPort
            .PortName = "COM" & CStr(ComPort)
            .BaudRate = CInt(frmMain.cmbBaud.SelectedItem)
            .DataBits = 8
            .Parity = Parity.None
            .StopBits = StopBits.One
            .Handshake = Handshake.None
            .ReadTimeout = 1000
        End With
        ' If it makes it to here, then the Comm Port is available.

        Try
            m_CommPort.Open()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try

        m_CommPort.Close()
        Return True
    Catch
        ' If it gets here, then the attempt to open the Comm Port
        '   was unsuccessful.
        Return False
    End Try
End Function

' This function checks to see if the port is a iRobot Create, by sending
'   an 128, 132 command to the port. The function returns a Boolean.
Public Function IsPortAIRobotCreate(ByVal ComPort As Integer) As Boolean

    ' Always wrap up working with Comm Ports in exception handlers.
    Try
        m_CommPort.DtrEnable = False
        m_CommPort.RtsEnable = False

        ' Attempt to open the port.
        With m_CommPort
            .PortName = "COM" & CStr(ComPort)
            .BaudRate = CInt(frmMain.cmbBaud.SelectedItem)
            .DataBits = 8
            .Parity = IO.Ports.Parity.None
            .StopBits = IO.Ports.StopBits.One
            .Handshake = IO.Ports.Handshake.None
            .ReceivedBytesThreshold = 1
            .RtsEnable = True
            .ReadTimeout = 1000
            .ReadBufferSize = 1000
        End With


        Try
            m_CommPort.Open()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        End Try


        ' Write an 128 Command to the Port.
        m_CommPort.Write({128}, 0, 1)
        ' Sleep long enough for the iRobot create to respond.
        System.Threading.Thread.Sleep(100)

        ' Write an 132 Command to the Port.
        m_CommPort.Write({132}, 0, 1)
        ' Sleep long enough for the iRobot create to respond.
```

```
            System.Threading.Thread.Sleep(100)

            ' light LEDS
            Dim LedsBytes() As Byte = {139, 25, 0, 128}
            m_CommPort.Write(LedsBytes, 0, 4)
            ' Sleep long enough for the iRobot create to respond.
            System.Threading.Thread.Sleep(100)
            Erase LedsBytes

            ' set song
            Dim SongBytes() As Byte = {140, 1, 1, 48, 20}
            m_CommPort.Write(SongBytes, 0, 5)
            ' Sleep long enough for the iRobot create to respond.
            System.Threading.Thread.Sleep(50)

            ' sing it
            Dim SingBytes() As Byte = {141, 1}
            m_CommPort.Write(SingBytes, 0, 2)

            System.Threading.Thread.Sleep(15)

            Dim returnvalue As String
            returnvalue = m_CommPort.ReadExisting()
            frmMain.WriteMessage(returnvalue)


            ' Try to get info from the Comm Port.
            Try
                IsPortAIRobotCreate = True

                Return True
            Catch exc As Exception
                ' Nothing to read from the Comm Port, so set to False
                m_CommPort.Close()

                IsPortAIRobotCreate = False

                Return False
            End Try
        Catch exc As Exception
            ' Port could not be opened or written to.
            MsgBox("Could not open port.", MsgBoxStyle.OkOnly)

            IsPortAIRobotCreate = False

            Return False
        End Try

    End Function


    Public Function SetFwdVelRadiusRoomba(ByVal m_Port As Integer, ByVal FwdVel As Double, ByVal
Radius As Double) As Boolean
        ' Dim strN As Integer

        Dim FwdVelMM As Integer = 0
        Dim RadiusMM As Integer = 0

        Dim BYTE_HI As Byte = 0
        Dim BYTE_LO As Byte = 0

        If m_CommPort.IsOpen = True Then

            ''''''''''''''''''''''''''''''''''''''''''''
            ' strN = m_CommPort.Read(4)
```

```
            ''''''''''''''''''''''''''''''''''''''''''''
            ' Debug.Print("strN " & strN)

            FwdVelMM = CInt(Min(Max(FwdVel, -0.5), 0.5) * 1000)

            If CBool(isinf(Radius)) Then
                RadiusMM = 32768
            ElseIf Radius = Eps Then
                RadiusMM = 1
            ElseIf Radius = -Eps Then
                RadiusMM = -1
            Else
                RadiusMM = CInt(Min(Max(Radius * 1000, -2000), 2000))
            End If

            ' Write an 137 Command to the Port.
            ''''''''''''''''''''''''''''''''''''''''''''
            ''Komanda per levizje
            ''''''''''''''''''''''''''''''''''''''''''''
            m_CommPort.Write({137}, 0, 1)

            ''''''''''''''''''''''''''''''''''''''''''''
            ''FwdVelMM
            ''''''''''''''''''''''''''''''''''''''''''''
            BYTE_HI = HIBYTE(FwdVelMM)
            m_CommPort.Write({BYTE_HI}, 0, 1)

            BYTE_LO = LOBYTE(FwdVelMM)
            m_CommPort.Write({BYTE_LO}, 0, 1)

            ''''''''''''''''''''''''''''''''''''''''''''
            ''RadiusMM
            ''''''''''''''''''''''''''''''''''''''''''''
            BYTE_HI = HIBYTE(RadiusMM)
            m_CommPort.Write({BYTE_HI}, 0, 1)

            BYTE_LO = LOBYTE(RadiusMM)
            m_CommPort.Write({BYTE_LO}, 0, 1)

        End If

        ' Sleep long enough for the iRobot Create to respond and the timer to fire.
        System.Threading.Thread.Sleep(100)

        Application.DoEvents()
    End Function

    Public Function DistanceSensorRoomba(ByVal m_port As Integer) As Double
        Dim strDistance As Integer = 0
        Dim Distance As Double = 0

        If m_CommPort.IsOpen = True Then

            Dim DataSend() As Byte = {142, 19}
            m_CommPort.Write(DataSend, 0, 2)

            ' As long as there is information, read one byte at a time and
            '   output it.
            Dim n As Integer = m_CommPort.BytesToRead
            Dim buffer As Byte() = New Byte(n - 1) {}
            Dim offset As Integer = 0
            Dim count As Integer = 1

            If n > 0 Then
                strDistance = CInt(m_CommPort.Read(buffer, offset, count))
```

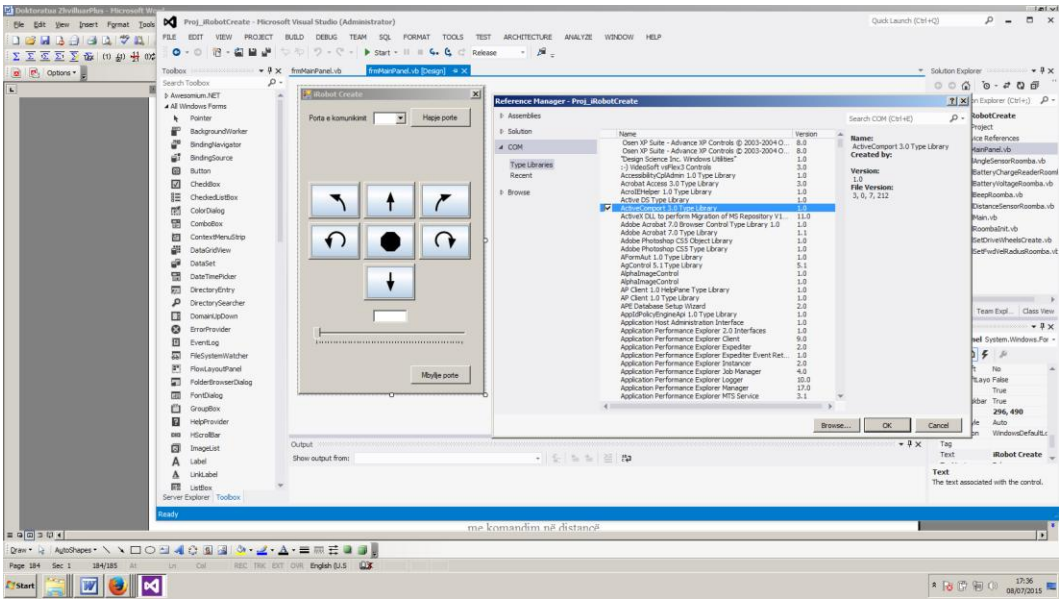
```
            Else
                strDistance = 0
            End If
            frmMain.WriteMessage(CStr(strDistance))

            Err.Clear()

            If strDistance <> 0 Then
                If IsNumeric(strDistance) Then
                    Distance = CDbl(strDistance) / 1000
                Else
                    Distance = 0
                End If
            Else
                Distance = 0
            End If

            If Distance > 32 Or Distance < -32 Then
                DistanceSensorRoomba = 0
                MsgBox("Warning:  May have overflowed", vbInformation)
            Else
                DistanceSensorRoomba = Distance
            End If

        Else
            DistanceSensorRoomba = 0
            MsgBox("Porta jo e hapur.", vbInformation)
        End If

        ' Sleep long enough for the iRobot Create to respond and the timer to fire.
        System.Threading.Thread.Sleep(100)

        Application.DoEvents()
    End Function

    ' Convert two Byte array elements to a UInt16 and display it.
    Public Function BAToUInt16(ByVal bytes() As Byte, ByVal index As Integer) As UInt16
        BAToUInt16 = BitConverter.ToUInt16(bytes, index)
    End Function

    Public Function BatteryVoltageRoomba() As String
        ' Indicates the voltage of Create's battery in Volts
        ' By; Genti PROGRI,  2014
        ' Initialize preliminary return values

        If m_CommPort.IsOpen = True Then

            Dim DataSend() As Byte = {142, 25}
            m_CommPort.Write(DataSend, 0, 2)

            BatteryVoltageRoomba = "?"
            Exit Function

            '     Dim n As Integer = m_CommPort.BytesToRead
            '     Dim buffer As Byte() = New Byte(n - 1) {}
            '     Dim offset As Integer = 0
            '     Dim count As Integer = 1
            '     Dim returnvalue As Integer
            '     If n > 0 Then
            ' returnvalue = m_CommPort.Read(Buffer, offset, count)
            '
            'BatteryVoltageRoomba = CStr(BAToUInt16(Buffer, 0) / 1000)
            ' Else
            '     BatteryVoltageRoomba = "?"
            ' End If
```

```
    Else
        BatteryVoltageRoomba = "?"
    End If
End Function


Public Function ChangeBaudRateRoomba(ByVal indexBaudCode As Byte) As String
    ' Indicates the voltage of Create's battery in Volts
    ' By; Genti PROGRI,  2014
    ' Initialize preliminary return values

    If m_CommPort.IsOpen = True Then

        Dim n As Integer = m_CommPort.BytesToRead
        Dim buffer As Byte() = New Byte(n - 1) {}
        Dim offset As Integer = 0
        Dim count As Integer = 1
        Dim returnvalue As Integer
        If n > 0 Then
            returnvalue = m_CommPort.Read(buffer, offset, count)
        End If

        m_CommPort.Write({129}, 0, 1)
        m_CommPort.Write({indexBaudCode}, 0, 1)

        n = m_CommPort.BytesToRead
        buffer = New Byte(n - 1) {}
        If n > 0 Then
            returnvalue = m_CommPort.Read(buffer, offset, count)

            ChangeBaudRateRoomba = CStr(BAToUInt16(buffer, 0) / 1000)
        Else
            ChangeBaudRateRoomba = "?"
        End If
    Else
        ChangeBaudRateRoomba = "?"
    End If
End Function

Public Function AllSensorsReadRoomba() As Boolean
    ' Indicates the voltage of Create's battery in Volts
    ' By; Genti PROGRI,  2014
    ' Initialize preliminary return values

    If m_CommPort.IsOpen = True Then

        Dim n As Integer = m_CommPort.BytesToRead
        Dim buffer As Byte() = New Byte(n - 1) {}
        Dim offset As Integer = 0
        Dim count As Integer = 1
        Dim returnvalue As Integer
        If n > 0 Then
            returnvalue = m_CommPort.Read(buffer, offset, count)
        End If

        '  Get (142) ALL(0) data fields
        m_CommPort.Write({142}, 0, 1)
        m_CommPort.Write({0}, 0, 1)

        n = m_CommPort.BytesToRead
        buffer = New Byte(n - 1) {}
        If n > 0 Then
            returnvalue = m_CommPort.Read(buffer, offset, count)
            Dim i As Integer
            For i = 1 To n - 1
```

```
                    frmMain.txtStatus.Text += CStr(buffer(i))
                Next

            End If

            AllSensorsReadRoomba = True
        Else
            AllSensorsReadRoomba = False
        End If
    End Function

End Module
```

Shtojca 10

Në vazhdim po japim librarinë e komandimit të lëvizjeve së robotit në Visual Basic 6.0 dhe Visual Basic .NET bazuar në librarinë ActiveComport Serial Port.



frmMainPanel.Designer.vb

```
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class frmMainPanel
    Inherits System.Windows.Forms.Form

    'Form overrides dispose to clean up the component list.
    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Dim resources As System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(frmMainPanel))
        Me.btMajtas = New System.Windows.Forms.Button()
        Me.btPara = New System.Windows.Forms.Button()
        Me.btDjathtas = New System.Windows.Forms.Button()
        Me.btRrotullimmajtas = New System.Windows.Forms.Button()
```

```
Me.btRrotullimdjathtas = New System.Windows.Forms.Button()
Me.btStop = New System.Windows.Forms.Button()
Me.btPrapa = New System.Windows.Forms.Button()
Me.SliderShpejtesia = New System.Windows.Forms.TrackBar()
Me.buttonOpen = New System.Windows.Forms.Button()
Me.buttonClose = New System.Windows.Forms.Button()
Me.Label1 = New System.Windows.Forms.Label()
Me.cmbPorta = New System.Windows.Forms.ComboBox()
Me.txtShpejtesia = New System.Windows.Forms.TextBox()
CType(Me.SliderShpejtesia, System.ComponentModel.ISupportInitialize).BeginInit()
Me.SuspendLayout()
'
'btMajtas
'
Me.btMajtas.Image = CType(resources.GetObject("btMajtas.Image"), System.Drawing.Image)
Me.btMajtas.Location = New System.Drawing.Point(12, 129)
Me.btMajtas.Name = "btMajtas"
Me.btMajtas.Size = New System.Drawing.Size(82, 59)
Me.btMajtas.TabIndex = 0
Me.btMajtas.UseVisualStyleBackColor = True
'
'btPara
'
Me.btPara.Image = CType(resources.GetObject("btPara.Image"), System.Drawing.Image)
Me.btPara.Location = New System.Drawing.Point(100, 129)
Me.btPara.Name = "btPara"
Me.btPara.Size = New System.Drawing.Size(82, 59)
Me.btPara.TabIndex = 1
Me.btPara.UseVisualStyleBackColor = True
'
'btDjathtas
'
Me.btDjathtas.Image = CType(resources.GetObject("btDjathtas.Image"), System.Drawing.Image)
Me.btDjathtas.Location = New System.Drawing.Point(188, 129)
Me.btDjathtas.Name = "btDjathtas"
Me.btDjathtas.Size = New System.Drawing.Size(82, 59)
Me.btDjathtas.TabIndex = 2
Me.btDjathtas.UseVisualStyleBackColor = True
'
'btRrotullimmajtas
'
Me.btRrotullimmajtas.Image = CType(resources.GetObject("btRrotullimmajtas.Image"), System.Drawing.Image)
Me.btRrotullimmajtas.Location = New System.Drawing.Point(12, 194)
Me.btRrotullimmajtas.Name = "btRrotullimmajtas"
Me.btRrotullimmajtas.Size = New System.Drawing.Size(82, 59)
Me.btRrotullimmajtas.TabIndex = 3
Me.btRrotullimmajtas.UseVisualStyleBackColor = True
'
'btRrotullimdjathtas
'
Me.btRrotullimdjathtas.Image = CType(resources.GetObject("btRrotullimdjathtas.Image"), System.Drawing.Image)
Me.btRrotullimdjathtas.Location = New System.Drawing.Point(188, 194)
Me.btRrotullimdjathtas.Name = "btRrotullimdjathtas"
Me.btRrotullimdjathtas.Size = New System.Drawing.Size(82, 59)
Me.btRrotullimdjathtas.TabIndex = 4
Me.btRrotullimdjathtas.UseVisualStyleBackColor = True
'
'btStop
'
Me.btStop.Image = CType(resources.GetObject("btStop.Image"), System.Drawing.Image)
Me.btStop.Location = New System.Drawing.Point(100, 194)
Me.btStop.Name = "btStop"
Me.btStop.Size = New System.Drawing.Size(82, 59)
Me.btStop.TabIndex = 5
Me.btStop.UseVisualStyleBackColor = True
'
```

```
'btPrapa
'
Me.btPrapa.Image = CType(resources.GetObject("btPrapa.Image"), System.Drawing.Image)
Me.btPrapa.Location = New System.Drawing.Point(100, 259)
Me.btPrapa.Name = "btPrapa"
Me.btPrapa.Size = New System.Drawing.Size(82, 59)
Me.btPrapa.TabIndex = 6
Me.btPrapa.UseVisualStyleBackColor = True
'
'SliderShpejtesia
'
Me.SliderShpejtesia.Location = New System.Drawing.Point(12, 359)
Me.SliderShpejtesia.Maximum = 50
Me.SliderShpejtesia.Name = "SliderShpejtesia"
Me.SliderShpejtesia.Size = New System.Drawing.Size(258, 45)
Me.SliderShpejtesia.TabIndex = 7
'
'buttonOpen
'
Me.buttonOpen.Location = New System.Drawing.Point(175, 12)
Me.buttonOpen.Name = "buttonOpen"
Me.buttonOpen.Size = New System.Drawing.Size(95, 30)
Me.buttonOpen.TabIndex = 8
Me.buttonOpen.Text = "Hapje porte"
Me.buttonOpen.UseVisualStyleBackColor = True
'
'buttonClose
'
Me.buttonClose.Location = New System.Drawing.Point(175, 422)
Me.buttonClose.Name = "buttonClose"
Me.buttonClose.Size = New System.Drawing.Size(95, 29)
Me.buttonClose.TabIndex = 9
Me.buttonClose.Text = "Mbyllje porte"
Me.buttonClose.UseVisualStyleBackColor = True
'
'Label1
'
Me.Label1.AutoSize = True
Me.Label1.Location = New System.Drawing.Point(10, 21)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(99, 13)
Me.Label1.TabIndex = 10
Me.Label1.Text = "Porta e komunikimit"
'
'cmbPorta
'
Me.cmbPorta.FlatStyle = System.Windows.Forms.FlatStyle.System
Me.cmbPorta.FormattingEnabled = True
Me.cmbPorta.Location = New System.Drawing.Point(115, 18)
Me.cmbPorta.Name = "cmbPorta"
Me.cmbPorta.Size = New System.Drawing.Size(54, 21)
Me.cmbPorta.TabIndex = 11
'
'txtShpejtesia
'
Me.txtShpejtesia.Location = New System.Drawing.Point(115, 333)
Me.txtShpejtesia.Name = "txtShpejtesia"
Me.txtShpejtesia.Size = New System.Drawing.Size(54, 20)
Me.txtShpejtesia.TabIndex = 12
'
'frmMainPanel
'
Me.AutoScaleDimensions = New System.Drawing.SizeF(6.0!, 13.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.ClientSize = New System.Drawing.Size(288, 463)
Me.Controls.Add(Me.txtShpejtesia)
```

```
        Me.Controls.Add(Me.cmbPorta)
        Me.Controls.Add(Me.Label1)
        Me.Controls.Add(Me.buttonClose)
        Me.Controls.Add(Me.buttonOpen)
        Me.Controls.Add(Me.SliderShpejtesia)
        Me.Controls.Add(Me.btPrapa)
        Me.Controls.Add(Me.btStop)
        Me.Controls.Add(Me.btRrotullimdjathtas)
        Me.Controls.Add(Me.btRrotullimmajtas)
        Me.Controls.Add(Me.btDjathtas)
        Me.Controls.Add(Me.btPara)
        Me.Controls.Add(Me.btMajtas)
        Me.MaximizeBox = False
        Me.MinimizeBox = False
        Me.Name = "frmMainPanel"
        Me.Text = "iRobot Create"
        CType(Me.SliderShpejtesia, System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub
    Friend WithEvents btMajtas As System.Windows.Forms.Button
    Friend WithEvents btPara As System.Windows.Forms.Button
    Friend WithEvents btDjathtas As System.Windows.Forms.Button
    Friend WithEvents btRrotullimmajtas As System.Windows.Forms.Button
    Friend WithEvents btRrotullimdjathtas As System.Windows.Forms.Button
    Friend WithEvents btStop As System.Windows.Forms.Button
    Friend WithEvents btPrapa As System.Windows.Forms.Button
    Friend WithEvents SliderShpejtesia As System.Windows.Forms.TrackBar
    Friend WithEvents buttonOpen As System.Windows.Forms.Button
    Friend WithEvents buttonClose As System.Windows.Forms.Button
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents cmbPorta As System.Windows.Forms.ComboBox
    Friend WithEvents txtShpejtesia As System.Windows.Forms.TextBox

End Class
```

frmMainPanel.vb

```
Public Class frmMainPanel

    Private Sub frmMainPanel_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        cmbPorta.Items.Add("COM1")
        cmbPorta.Items.Add("COM2")
        cmbPorta.Items.Add("COM3")
        cmbPorta.Items.Add("COM4")
        cmbPorta.Items.Add("COM5")
        cmbPorta.Items.Add("COM6")
        cmbPorta.Items.Add("COM7")
        cmbPorta.Items.Add("COM8")
        cmbPorta.Items.Add("COM9")
        cmbPorta.Items.Add("COM10")
        cmbPorta.SelectedIndex = 0

        SliderShpejtesia.Value = 20
        txtShpejtesia.Text = SliderShpejtesia.Value * 10
        Try
            objComport = CreateObject("ActiveXperts.ComPort")
            objComport.ComTimeout = 200

        Catch
            MsgBox("Problem ne procesin e krijimit te objectit ActiveXperts.ComPort", vbCritical, Application.ProductName)
        End Try
    End Sub
```

```
    Private Sub SliderShpejtesia_EnabledChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles
SliderShpejtesia.EnabledChanged
        txtShpejtesia.Text = SliderShpejtesia.Value * 10
    End Sub

    Private Sub SliderShpejtesia_Scroll(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SliderShpejtesia.Scroll
        txtShpejtesia.Text = SliderShpejtesia.Value * 10
    End Sub

    Private Sub buttonOpen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles buttonOpen.Click
        RoombaInit(Mid(cmbPorta.Text, 4))
    End Sub

    Private Sub buttonClose_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles buttonClose.Click
        If objComport.IsOpened Then
            objComport.Close()
        End If
    End Sub

    Private Sub btPrapa_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btPrapa.Click
        If objComport.IsOpened = True Then
            'sets forward velocity 1 m/s and turning radius 2 m
            SetFwdVelRadiusRoomba(-SliderShpejtesia.Value / 100, 2)
        Else
            MsgBox("Porta jo e hapur", vbCritical, Application.ProductName)
        End If
    End Sub

    Private Sub btMajtas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btMajtas.Click
        SetDriveWheelsCreate(0, SliderShpejtesia.Value / 100)
    End Sub

    Private Sub btPara_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btPara.Click
        If objComport.IsOpened = True Then
            'sets forward velocity 1 m/s and turning radius 2 m
            SetFwdVelRadiusRoomba(SliderShpejtesia.Value / 100, 2)
        Else
            MsgBox("Porta jo e hapur", vbCritical, Application.ProductName)
        End If
    End Sub

    Private Sub btDjathtas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btDjathtas.Click
        SetDriveWheelsCreate(SliderShpejtesia.Value / 100, 0)
    End Sub

    Private Sub btRrotullimmajtas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btRrotullimmajtas.Click
        SetDriveWheelsCreate(-SliderShpejtesia.Value / 100, SliderShpejtesia.Value / 100)
    End Sub

    Private Sub btStop_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btStop.Click
        If objComport.IsOpened = True Then
            ' stop the robot
            SetFwdVelRadiusRoomba(0, 1)
        Else
            MsgBox("Porta jo e hapur", vbCritical, Application.ProductName)
        End If
    End Sub

    Private Sub btRrotullimdjathtas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btRrotullimdjathtas.Click
        SetDriveWheelsCreate(SliderShpejtesia.Value / 100, -SliderShpejtesia.Value / 100)
    End Sub
End Class
```

modMain.vb

```
Module modMain

    Public Const Pi As Double = 3.1416
    Public Const Eps As Double = 0.00000000000000022204

    Public objComport As ACOMPORTLib.ComPort
    Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
    Public MessageBack As String

    Public Function HIBYTE(ByVal Intg As Integer) As Byte
        HIBYTE = (Intg And &HFF00&) / 256
    End Function

    Public Function LOBYTE(ByVal Intg As Integer) As Byte
        LOBYTE = Intg And &HFF&
    End Function

    Public Function UnHex(ByVal sHex As String) As Long
        UnHex = Val("&H" & sHex)
    End Function

    Public Function Min(ByVal Val1 As Double, ByVal Val2 As Double) As Double
        If Val1 > Val2 Then
            Min = Val2
        Else
            Min = Val1
        End If
    End Function

    Public Function Max(ByVal Val1 As Double, ByVal Val2 As Double) As Double
        If Val1 > Val2 Then
            Max = Val1
        Else
            Max = Val2
        End If
    End Function

    Public Function isinf(ByVal var As Double) As Integer
        If var > 1.0E+100 Or var < -1.0E+100 Then
            isinf = 1
        Else
            isinf = 0
        End If
    End Function
End Module
```

modRoombaInit.vb

```
Imports Microsoft.VisualBasic
Module modRoombaInit
    ' Krijuar nga Genti Progri
    '                          Date : 01/12/2013
    ' Funksioni hapes i portes se komunikimit me iRobotCreate

    ' % Initializes serial port for use with Roomba
    ' % COMMport is the number of the comm port
    ' % ex. RoombaInit(1) sets port = 'COM1'
    ' % note that it sets baudrate to a default of 57600
    ' % can be changed (see SCI).
    ' % An optional time delay can be added after all commands
```

```
' % if your code crashes frequently.
Public Const td = 15
' % This code puts the robot in CONTROL(132) mode, which means does NOT stop
' % when cliff sensors or wheel drops are true; can also run while plugged into charger
Public Contrl As Integer = 132

Public WaitConfirmationFrom_RoombaInit As Boolean

Public Function RoombaInit(ByVal my_COM As String) As Boolean
    Dim Key As Integer
    Dim comm As String
    Dim confirmation As String

    ' Output buffer must be big enough to take largest message size
    If IsNumeric(my_COM) = False Then
        MessageBack = "Porte komunikimi e pa percaktuar."
        RoombaInit = False
    End If

    ' Establishing connection to Roomba...
    comm = "COM" & my_COM


    objComport.Device = comm

    objComport.ComTimeout = 100

    objComport.LogFile = System.Windows.Forms.Application.ExecutablePath & "\Errors.Log"

    ' defaults at 57600, can change
    objComport.BaudRate = 57600

    objComport.HardwareFlowControl = 0
    objComport.SoftwareFlowControl = 0

    objComport.DataBits = objComport.asDATABITS_8
    objComport.StopBits = objComport.asSTOPBITS_1
    objComport.Parity = objComport.asPARITY_NONE

    ' objComport.RaiseDTR (0)
    ' objComport.RaiseRTS (0)

    objComport.Open()

    confirmation = GetResult()


    If objComport.IsOpened = True And objComport.LastError = 0 Then

        Key = 128
        objComport.WriteByte(Key)
        Sleep(100)

        Key = 132
        objComport.WriteByte(Key)
        Sleep(100)

        ' light LEDS
        Key = 139
        objComport.WriteByte(Key)
        Key = 25
        objComport.WriteByte(Key)
        Key = 0
        objComport.WriteByte(Key)
        Key = 128
```

```
            objComport.WriteByte(Key)
            Sleep(100)


            ' set song
            Key = 140
            objComport.WriteByte(Key)
            Key = 1
            objComport.WriteByte(Key)
            Key = 1
            objComport.WriteByte(Key)
            Key = 48
            objComport.WriteByte(Key)
            Key = 20
            objComport.WriteByte(Key)
            Sleep(50)


            ' sing it
            Key = 141
            objComport.WriteByte(Key)
            Key = 1
            objComport.WriteByte(Key)

            Application.DoEvents()

            Dim StartTime As Single
            confirmation = String.Empty
            StartTime = Microsoft.VisualBasic.Timer
            Do While Not Microsoft.VisualBasic.Timer - StartTime > 1
                confirmation = confirmation & objComport.ReadString
            Loop
            Debug.Print(confirmation)

            WaitConfirmationFrom_RoombaInit = True

            RoombaInit = True
        Else
            MessageBack = "Pamundesi ne hapjen e portes se komunikimit."

            WaitConfirmationFrom_RoombaInit = False

            RoombaInit = False
        End If



    End Function

    Public Function GetResult() As String
        If objComport.LastError = 0 Then
            GetResult = "SUCCESS"
        Else
            GetResult = "ERROR " & objComport.LastError & " ( " &
objComport.GetErrorDescription(objComport.LastError) & " )"
        End If
    End Function

End Module
```

modAngleSensorRoomba.vb

```
Module modAngleSensorRoomba
```

```
    Public Function AngleSensorRoombaVB6() As Double
        '[AngleR] = AngleSensorRoombaVB6()
        ' Displays the angle in radians and degrees that Create has turned since the angle was last
requested.
        ' Counter-clockwise angles are positive and Clockwise angles are negative.
        Dim Key As Integer
        Dim strN As String

        If objComport.IsOpened = True Then

            Key = 142
            objComport.WriteByte(Key)
            Key = 20
            objComport.WriteByte(Key)

            strN = objComport.ReadByte
            If Not IsNumeric(strN) Then
                strN = "0"
            End If

            '''''''''''''''''''''''''''''''''''''''''''''
            If IsNumeric(strN) Then
                AngleSensorRoombaVB6 = CDbl(strN) * Pi / 180
            Else
                AngleSensorRoombaVB6 = 0
            End If

        Else
            AngleSensorRoombaVB6 = 0
        End If

    End Function

End Module
```

modBatteryChargeReaderRoomba.vb

```
Module modBatteryChargeReaderRoomba

    Structure Type_BatteryCharge
        Public Charge As Single
        Public Capacity As Single
        Public Percent As Double
    End Structure

    Public Function BatteryChargeReaderRoombaVB6() As Type_BatteryCharge
        ' Displays the current percent charge remaining in Create's Battery
        ' By; Genti PROGRI, 2014
        ' Initialize preliminary return values

        Dim Key As Integer

        Dim BatteryCharge As Type_BatteryCharge

        Dim sCharge As String
        Dim sCapacity As String

        If objComport.IsOpened = True Then

            ' Charge
            '''''''''''''''''''''''''''''''''''''''''''''
            Key = 142
            objComport.WriteByte(Key)
```

```
            Key = 25
            objComport.WriteByte(Key)

            sCharge = objComport.ReadByte
            If IsNumeric(sCharge) Then
                BatteryCharge.Charge = CLng(sCharge)
            Else
                BatteryCharge.Charge = 0
            End If
            '''''''''''''''''''''''''''''''''''''''''''


            ' Capacity
            '''''''''''''''''''''''''''''''''''''''''''
            Key = 142
            objComport.WriteByte(Key)
            Key = 26
            objComport.WriteByte(Key)

            sCapacity = objComport.ReadByte
            If IsNumeric(sCapacity) Then
                BatteryCharge.Capacity = CLng(sCapacity)
            Else
                BatteryCharge.Capacity = 0
            End If
            '''''''''''''''''''''''''''''''''''''''''''



            ' Percent
            '''''''''''''''''''''''''''''''''''''''''''
            If IsNumeric(BatteryCharge.Charge) And _
               IsNumeric(BatteryCharge.Capacity) And _
               BatteryCharge.Capacity <> 0 Then

                BatteryCharge.Percent = 100 * BatteryCharge.Charge / BatteryCharge.Capacity

            Else
                BatteryCharge.Charge = 0
                BatteryCharge.Capacity = 0
                BatteryCharge.Percent = 0
            End If

        Else
            BatteryCharge.Charge = 0
            BatteryCharge.Capacity = 0
            BatteryCharge.Percent = 0
        End If

        BatteryChargeReaderRoombaVB6 = BatteryCharge
    End Function
End Module
```

modBatteryVoltageRoomba.vb

```
Module modBatteryVoltageRoomba

    Public Function BatteryVoltageRoombaVB6() As String
        ' Indicates the voltage of Create's battery in Volts
        ' By; Genti PROGRI,  2014
        ' Initialize preliminary return values

        Dim Key As Byte
```

```
        If objComport.IsOpened = True Then
            Key = 142
            objComport.WriteByte(Key)
            Key = 25
            objComport.WriteByte(Key)
            objComport.Sleep(0.1)
            BatteryVoltageRoombaVB6 = CDbl((objComport.ReadByte)) / 1000
        Else
            BatteryVoltageRoombaVB6 = "?"
        End If
    End Function
End Module
```

modBeepRoomba.vb

```
Module modBeepRoomba

    Public Sub BeepRoombaVB6()
        ' [] = BeepRoomba()
        '  Plays a song made by RoombaInit command.
        '  By  Genti PROGRI, 2014
        '  sing it

        Dim Key As Byte
        Dim strN As String

        If objComport.IsOpened = True Then

            Key = 141
            objComport.WriteByte(Key)
            Key = 1
            objComport.WriteByte(Key)

            strN = objComport.ReadString
            Debug.Print("Beep Roomba " & strN)
        Else
            Debug.Print("Beep Roomba " & "Porta jo e hapur.")
        End If
    End Sub

End Module
```

modDistanceSensorRoomba.vb

```
Module modDistanceSensorRoomba

    Public Function DistanceSensorRoomba() As Double
        ' [Distance] = DistanceSensorRoomba(serPort)
        '  Gives the distance traveled in meters since last requested. Positive
        '  values indicate travel in the forward direction. Negative values indicate
        '  travel in the reverse direction. If not polled frequently enough, it is
        '  capped at its minimum or maximum of +/- 32.768 meters.

        '  By  Genti PROGRI, 2014
        Dim Key As Integer
        Dim strDistance As String
        Dim Distance As Double

        If objComport.IsOpened = True Then

            Key = 142
            objComport.WriteByte(Key)
```

```
            Key = 19
            objComport.WriteByte(Key)

            strDistance = objComport.ReadByte
            If strDistance <> "" Then
                If IsNumeric(strDistance) Then
                    Distance = CDbl(strDistance) / 1000
                Else
                    Distance = 0
                End If
            Else
                Distance = 0
            End If

            If Distance > 32 Or Distance < -32 Then
                DistanceSensorRoomba = 0
                MsgBox("Kujdes:  Jashte kufijve te mundur (error)", vbInformation,
Application.ProductName)
            Else
                DistanceSensorRoomba = Distance
            End If

        Else
            DistanceSensorRoomba = 0
            MsgBox("Porta jo e hapur.", vbInformation, Application.ProductName)
        End If
    End Function
End Module
```

modSetDriveWheelsCreate.vb

```
Module modSetDriveWheelsCreate

'  By  Genti PROGRI, 2014

    Public Function SetDriveWheelsCreate(ByVal leftWheelVel As Double, ByVal rightWheelVel As
Double) As Boolean
        Dim Key As Byte

        Dim BYTE_HI As Byte
        Dim BYTE_LO As Byte


        If objComport.IsOpened = True Then

            rightWheelVel = Min(Max(1000 * rightWheelVel, -500), 500)
            leftWheelVel = Min(Max(1000 * leftWheelVel, -500), 500)


            ''''''''''''''''''''''''''''''''
            ''Komanda per levizje
            ''''''''''''''''''''''''''''''''
            Key = 145
            objComport.WriteByte(Key)

            ''''''''''''''''''''''''''''''''
            ''rightWheelVel
            ''''''''''''''''''''''''''''''''
            BYTE_HI = HIBYTE(rightWheelVel)
            objComport.WriteByte(BYTE_HI)

            BYTE_LO = LOBYTE(rightWheelVel)
            objComport.WriteByte(BYTE_LO)
```

```
            ''''''''''''''''''''''''''''''
            ''leftWheelVel
            ''''''''''''''''''''''''''''''
            BYTE_HI = HIBYTE(leftWheelVel)
            objComport.WriteByte(BYTE_HI)

            BYTE_LO = LOBYTE(leftWheelVel)
            objComport.WriteByte(BYTE_LO)

            objComport.Sleep(td)

            SetDriveWheelsCreate = True
        Else
            SetDriveWheelsCreate = False
        End If
    End Function
End Module
```

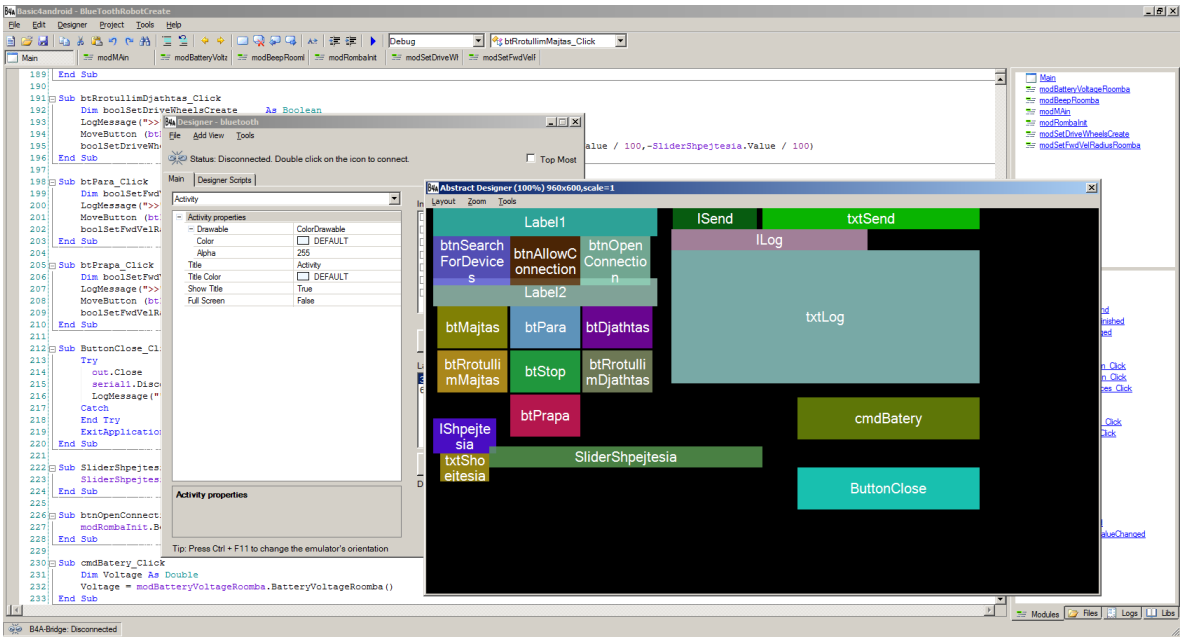
modSetFwdVelRadiusRoomba.vb

```
Module modSetFwdVelRadiusRoomba

    ' [] = SetFwdVelRadiusRoomba(serPort, FwdVel, Radius)

    '   Moves Roomba by setting forward vel and turn radius
    '   serPort is a serial port object created by Roombainit
    '   FwdVel is forward vel in m/sec [-0.5, 0.5],
    '   Radius in meters, postive turns left, negative turns right [-2,2].
    '   Special cases: Straight = inf
    '   Turn in place clockwise = -eps
    '   Turn in place counter-clockwise = eps

    '   By  Genti PROGRI, 2014

    Public Function SetFwdVelRadiusRoomba(ByVal FwdVel As Double, ByVal Radius As Double) As Boolean
        Dim Key As Byte

        Dim FwdVelMM As Integer
        Dim RadiusMM As Integer

        Dim BYTE_HI As Byte
        Dim BYTE_LO As Byte

        If objComport.IsOpened = True Then

            FwdVelMM = Min(Max(FwdVel, -0.5), 0.5) * 1000

            If isinf(Radius) Then
                RadiusMM = 32768
            ElseIf Radius = Eps Then
                RadiusMM = 1
            ElseIf Radius = -Eps Then
                RadiusMM = -1
            Else
                RadiusMM = Min(Max(Radius * 1000, -2000), 2000)
            End If

            ''''''''''''''''''''''''''''''
            ''Komanda per levizje
            ''''''''''''''''''''''''''''''
            Key = 137
            objComport.WriteByte(Key)

            '\   ''''''''''''''''''''''''''''''
```

```
            '\   ''FwdVelMM
            '\   '''''''''''''''''''''''''''''
            '\   objComport.WriteString (FwdVelMM)
            '\
            '\   '''''''''''''''''''''''''''''
            '\   ''RadiusMM
            '\   '''''''''''''''''''''''''''''
            '\   objComport.WriteString (RadiusMM)

            '''''''''''''''''''''''''''''''
            ''FwdVelMM
            '''''''''''''''''''''''''''''''
            BYTE_HI = HIBYTE(FwdVelMM)
            objComport.WriteByte(BYTE_HI)

            BYTE_LO = LOBYTE(FwdVelMM)
            objComport.WriteByte(BYTE_LO)

            '''''''''''''''''''''''''''''''
            ''RadiusMM
            '''''''''''''''''''''''''''''''
            BYTE_HI = HIBYTE(RadiusMM)
            objComport.WriteByte(BYTE_HI)

            BYTE_LO = LOBYTE(RadiusMM)
            objComport.WriteByte(BYTE_LO)

            objComport.Sleep(td)

            SetFwdVelRadiusRoomba = True
        Else
            SetFwdVelRadiusRoomba = False
        End If
    End Function
End Module
```

Shtojca 11

Në vazhdim po japim editorin e gjuhës Basic4Androide dhe programet e librarisë për komandimin e lëvizjeve të robotit në Basic4Androide bazuar në librarinë seriale Androide 2.0.



BlueToothRobotCreate.b4a

```
#Region Module Attributes
	#FullScreen: False
	#IncludeTitle: True
	#ApplicationLabel: iRobot Create
	#VersionCode: 1
	#VersionName:
	#SupportedOrientations: unspecified
	#CanInstallToExternalStorage: False
#End Region

'Activity module
Sub Process_Globals
	Dim admin As BluetoothAdmin
	Dim out As OutputStream
	Dim Inp As InputStream
	Dim serial1 As Serial
	Dim foundDevices As List
	Type NameAndMac (Name As String, Mac As String)
	Dim connectedDevice As NameAndMac
End Sub

Sub Globals
	Dim btnAllowConnection As Button
	Dim btnSearchForDevices As Button
	Dim btDjathtas As ImageView
	Dim btMajtas As ImageView
	Dim btPara As ImageView
	Dim btPrapa As ImageView
	Dim btRrotullimDjathtas As ImageView
	Dim btRrotullimMajtas As ImageView
```

```
	Dim btStop 				As ImageView
	Dim ButtonClose 			As Button
	Dim SliderShpejtesia 			As SeekBar
	Dim txtShpejtesia 			As Label
	Dim txtLog 				As EditText
	Dim txtSend 				As EditText
	Dim SuccessConnection 			As Boolean
End Sub

Sub Activity_Create(FirstTime As Boolean)
	Activity.LoadLayout("bluetooth")
	If FirstTime Then
		admin.Initialize("admin")
		serial1.Initialize("serial1")

		btnSearchForDevices.Initialize("btnSearchForDevices")
	btnAllowConnection.Initialize("btnAllowConnection")
	End If
End Sub

Sub Activity_Resume
	btnSearchForDevices.Enabled = False
	btnAllowConnection.Enabled = False
	If admin.IsEnabled = False Then
		If admin.Enable = False Then
			ToastMessageShow("Error enabling Bluetooth adapter.", True)
		Else
			ToastMessageShow("Enabling Bluetooth adapter...", False)
			'the StateChanged event will be soon raised
		End If
	Else
		Admin_StateChanged(admin.STATE_ON, 0)
	End If
End Sub

Sub Activity_Pause (UserClosed As Boolean)
	If UserClosed = True Then
		serial1.Disconnect
	End If
End Sub

Sub btnSearchForDevices_Click
	foundDevices.Initialize
	If admin.StartDiscovery	= False Then
		ToastMessageShow("Error starting discovery process.", True)
	Else
		ProgressDialogShow("Searching for devices...")
	End If
End Sub

Sub Admin_StateChanged (NewState As Int, OldState As Int)
	btnSearchForDevices.Enabled = (NewState = admin.STATE_ON)
	btnAllowConnection.Enabled = btnSearchForDevices.Enabled
End Sub

Sub Admin_DiscoveryFinished
	ProgressDialogHide
	If foundDevices.Size = 0 Then
		ToastMessageShow("No device found.", True)
	Else
		Dim l As List
		l.Initialize
		For i = 0 To foundDevices.Size - 1
			Dim nm As NameAndMac
			nm = foundDevices.Get(i)
			l.Add(nm.Name)
		Next
```

```
                Dim res As Int
                res = InputList(l, "Choose device to connect", -1)
                If res <> DialogResponse.CANCEL Then
                        connectedDevice = foundDevices.Get(res)
                        ProgressDialogShow("Trying to connect to: " & connectedDevice.Name &
" (" & connectedDevice.Mac & ")")
                        serial1.Connect(connectedDevice.Mac)
                End If
        End If
End Sub

Sub Admin_DeviceFound (Name As String, MacAddress As String)
        Log(Name & ":" & MacAddress)
        Dim nm As NameAndMac
        nm.Name = Name
        nm.Mac = MacAddress
        foundDevices.Add(nm)
        ProgressDialogShow("Searching for devices (~ device found)...".Replace("~",
foundDevices.Size))
End Sub

Sub btnAllowConnection_Click
        'this intent makes the device discoverable for 300 seconds.
        Dim i As Intent
        i.Initialize("android.bluetooth.adapter.action.REQUEST_DISCOVERABLE", "")
        i.PutExtra("android.bluetooth.adapter.extra.DISCOVERABLE_DURATION", 300)
        StartActivity(i)
        serial1.Listen
End Sub

Sub Serial1_Connected (Success As Boolean)
        ProgressDialogHide

        Log("connected: " & Success)
        LogMessage("", "Connected: " & Success)

        If Success = False Then
                Log(LastException.Message)
                ToastMessageShow("Error connecting: " & LastException.Message, True)
                SuccessConnection=False
        Else
                out = serial1.OutputStream
                Inp = serial1.InputStream

                SuccessConnection=True
        End If
End Sub

Sub LogMessage(From As String, Msg As String)
        txtLog.Text = txtLog.Text & From & ": " & Msg & CRLF
        txtLog.SelectionStart = txtLog.Text.Length
End Sub

Sub MoveButton(bt As ImageView, intMove As Int)
        bt.Left=bt.Left + intMove
        bt.top=bt.top + intMove
        modMAin.Sleep(100)
        bt.Left=bt.Left - intMove
        bt.top=bt.top - intMove
End Sub

Sub btStop_Click
        LogMessage(">>", "Stop")
        MoveButton (btStop, 2)
        Dim boolSetFwdVelRadiusRoomba As Boolean
        Try
           boolSetFwdVelRadiusRoomba = modSetFwdVelRadiusRoomba.SetFwdVelRadiusRoomba( 0, 1)
```

```
        Catch
           ToastMessageShow("Error connecting: " & LastException.Message, True)
        End Try
End Sub

Sub btMajtas_Click
        Dim boolSetDriveWheelsCreate   As Boolean
        LogMessage(">>", "Majtas")
        MoveButton (btMajtas, 2)

        boolSetDriveWheelsCreate = modSetDriveWheelsCreate.SetDriveWheelsCreate( 0,
SliderShpejtesia.Value / 100)
End Sub

Sub btDjathtas_Click
        Dim boolSetDriveWheelsCreate   As Boolean
        LogMessage(">>","Djathtas")
        MoveButton (btDjathtas, 2)
        boolSetDriveWheelsCreate =
modSetDriveWheelsCreate.SetDriveWheelsCreate(SliderShpejtesia.Value / 100,0)
End Sub

Sub btRrotullimMajtas_Click
        Dim boolSetDriveWheelsCreate   As Boolean
        LogMessage(">>","Rrotullim majtas")
        MoveButton (btRrotullimMajtas, 2)
        boolSetDriveWheelsCreate = modSetDriveWheelsCreate.SetDriveWheelsCreate(-
SliderShpejtesia.Value / 100,SliderShpejtesia.Value / 100)
End Sub

Sub btRrotullimDjathtas_Click
        Dim boolSetDriveWheelsCreate   As Boolean
        LogMessage(">>","Rrotullim djathtas")
        MoveButton (btRrotullimDjathtas, 2)
        boolSetDriveWheelsCreate =
modSetDriveWheelsCreate.SetDriveWheelsCreate(SliderShpejtesia.Value / 100,-
SliderShpejtesia.Value / 100)
End Sub

Sub btPara_Click
        Dim boolSetFwdVelRadiusRoomba As Boolean
        LogMessage(">>","Para")
        MoveButton (btPara, 2)
        boolSetFwdVelRadiusRoomba =
modSetFwdVelRadiusRoomba.SetFwdVelRadiusRoomba(SliderShpejtesia.Value / 100, 2)
End Sub

Sub btPrapa_Click
        Dim boolSetFwdVelRadiusRoomba As Boolean
        LogMessage(">>","Prapa")
        MoveButton (btPrapa, 2)
        boolSetFwdVelRadiusRoomba = modSetFwdVelRadiusRoomba.SetFwdVelRadiusRoomba(-
SliderShpejtesia.Value / 100, 2)
End Sub

Sub ButtonClose_Click
        Try
          out.Close
          serial1.Disconnect
          LogMessage("", "Connected: Close")
        Catch
        End Try
        ExitApplication
End Sub

Sub SliderShpejtesia_ValueChanged (Value As Int, UserChanged As Boolean)
        SliderShpejtesia.Value=Value
```

```
End Sub

Sub btnOpenConnection_Click
        modRoombaInit.BeepRoombaConnect()
End Sub

Sub cmdBatery_Click
        Dim Voltage    As Double
        Voltage = modBatteryVoltageRoomba.BatteryVoltageRoomba()
End Sub
```

modRoombaInit.bas

```
Sub Process_Globals
        'These global variables will be declared once when the application starts.
        'These variables can be accessed from all modules.
End Sub

Public Sub BeepRoombaConnect()
        If Main.out.IsInitialized = True Then

                Dim buffer128() As Byte = Array As Byte(128)
                Main.out.WriteBytes(buffer128,0,1)
                modMAin.Sleep(100)

                'Dim buffer132() As Byte = Array As Byte(132)
                buffer128(0)=132
                Main.out.WriteBytes(buffer128,0,1)
                modMAin.Sleep(100)

                Dim buffer139() As Byte = Array As Byte(139)
                Main.out.WriteBytes(buffer139,0,1)

                Dim buffer25() As Byte = Array As Byte(25)
                Main.out.WriteBytes(buffer25,0,1)

                Dim buffer0() As Byte = Array As Byte(0)
                Main.out.WriteBytes(buffer0,0,1)

                Dim buffer128() As Byte = Array As Byte(128)
                Main.out.WriteBytes(buffer128,0,1)

                modMAin.Sleep(100)


                Dim buffer140() As Byte = Array As Byte(140)
                Main.out.WriteBytes(buffer140,0,1)

                Dim buffer1() As Byte = Array As Byte(1)
                Main.out.WriteBytes(buffer1,0,1)

                Dim buffer48() As Byte = Array As Byte(48)
                Main.out.WriteBytes(buffer48,0,1)

                Dim buffer20() As Byte = Array As Byte(20)
                Main.out.WriteBytes(buffer20,0,1)
                modMAin.Sleep(100)

                Dim buffer141() As Byte = Array As Byte(141)
                Main.out.WriteBytes(buffer141,0,1)

                Main.out.WriteBytes(buffer1,0,1)
                modMAin.Sleep(100)
        End If
End Sub
```

modMain.bas

```
Sub Process_Globals
        'These global variables will be declared once when the application starts.
        'These variables can be accessed from all modules.
End Sub

Public Sub Sleep(ms As Long)
Dim now As Long
   If ms > 1000 Then ms =1000   'avoid application not responding error
   now=DateTime.now
   Do Until (DateTime.now>now+ms)
     DoEvents
   Loop
End Sub

Public Sub HIBYTE( Intg As Int) As Int
        Dim andBit     As Int
        andBit = Bit.AND(Intg ,0xFF00)
        andBit = andBit / 256
    Return andBit
End Sub

Public Sub LOBYTE( Intg As Int) As Int
   Dim andBit  As Int
   andBit = Bit.AND(Intg ,0xFF)
   Return andBit
End Sub

Public Sub Minval(Val1 As Double, Val2 As Double) As Double
    Dim MinvalLlog     As Double
        If Val1 > Val2 Then
       MinvalLlog = Val2
    Else
       MinvalLlog = Val1
    End If
        Return MinvalLlog
End Sub

Public Sub Maxval(Val1 As Double, Val2 As Double) As Double
    Dim MaxvalLlog     As Double
        If Val1 > Val2 Then
       MaxvalLlog = Val1
    Else
       MaxvalLlog = Val2
    End If
        Return MaxvalLlog
End Sub

Public Sub isinf(var As Double) As Long
    Dim isinfResult As Long
        If var > 1E+100 OR var < -1E+100 Then
       isinfResult = 1
    Else
       isinfResult = 0
    End If
        Return isinfResult
End Sub

Public Sub CStr(o As Object) As String
 Return "" & o
End Sub

Public Sub CInt(o As Object) As Int
  Return Floor(o)
```

```
End Sub

Public Sub CLng(o As Object) As Long
  Return Floor(o)
End Sub
```

modBatteryVoltageRoomba.bas

```
Sub Process_Globals
        'These global variables will be declared once when the application starts.
        'These variables can be accessed from all modules.
End Sub

Public Sub BatteryVoltageRoomba() As Double
    Dim Voltage     As Double
    Dim s                       As String

        Voltage = 0

        If Main.out.IsInitialized = True Then

                If Main.Inp.IsInitialized = True Then

                        Dim buffer142() As Byte = Array As Byte(142)
                        Main.out.WriteBytes(buffer142,0,1)

                        Dim buffer25() As Byte = Array As Byte(25)
                        Main.out.WriteBytes(buffer25,0,1)

                        If Main.inp.BytesAvailable >0 Then
                           Dim buffer(Main.inp.BytesAvailable) As Byte
                           Main.inp.ReadBytes( buffer,0, Main.inp.BytesAvailable )
                           s = BytesToString(buffer,0,buffer.Length ,"Windows-1253")
                           Msgbox( s,"")
                        End If

                Else
                        Msgbox("input stream i pa inicializuar","Serial")
                End If
        Else
                Voltage = -9999
        End If

        Return  Voltage
End Sub
```

modBeepRoomba.bas

```
Sub Process_Globals
        'These global variables will be declared once when the application starts.
        'These variables can be accessed from all modules.
End Sub

Public Sub BeepRoombaVB6()
    Dim Key         As String
        Dim buffer()    As Byte

        Dim buffer145() As Byte = Array As Byte(145)
        Main.out.WriteBytes(buffer145,0,1)

        Dim buffer1() As Byte = Array As Byte(1)
        Main.out.WriteBytes(buffer1,0,1)
End Sub
```

modSetDriveWheelsCreate.bas

```
Sub Process_Globals
        'These global variables will be declared once when the application starts.
        'These variables can be accessed from all modules.
End Sub

Public Sub SetDriveWheelsCreate(leftWheelVel As Double, rightWheelVel As Double) As Boolean

        Dim boolSetDriveWheelsCreate  As Boolean
        Dim maxValue                  As Int
    Dim RealRightWheelVel     As Int
        Dim RealLeftWheelVel   As Int

        Dim BYTE_HI                           As Int
        Dim BYTE_LO                           As Int

        If Main.out.IsInitialized = True Then
                maxValue = modMAin.Maxval(1000 * rightWheelVel, -500)
                RealRightWheelVel = modMAin.Minval(maxValue, 500)

                maxValue = modMAin.Maxval(1000 * leftWheelVel, -500)
                RealLeftWheelVel = modMAin.Minval(maxValue, 500)

        ''''''''''''''''''''''''''''''''
        ''Komanda per levizje
        ''''''''''''''''''''''''''''''''
                Dim buffer() As Byte = Array As Byte(145)
                Main.out.WriteBytes(buffer,0,1)

        ''''''''''''''''''''''''''''''''
        ''rightWheelVel
        ''''''''''''''''''''''''''''''''
                BYTE_HI = modMAin.HIBYTE(RealRightWheelVel)
                BYTE_LO = modMAin.LOBYTE(RealRightWheelVel)
                Dim bufferHI1() As Byte = Array As Byte(BYTE_HI)
                Main.out.WriteBytes(bufferHI1,0,1)
                Dim bufferLO1() As Byte = Array As Byte(BYTE_LO)
                Main.out.WriteBytes(bufferLO1,0,1)

        ''''''''''''''''''''''''''''''''
        ''leftWheelVel
        ''''''''''''''''''''''''''''''''
                BYTE_HI = modMAin.HIBYTE(RealLeftWheelVel)
                BYTE_LO = modMAin.LOBYTE(RealLeftWheelVel)
                Dim bufferHI2() As Byte = Array As Byte(BYTE_HI)
                Main.out.WriteBytes(bufferHI2,0,1)
                Dim bufferLO2() As Byte = Array As Byte(BYTE_LO)
                Main.out.WriteBytes(bufferLO2,0,1)

                modMAin.Sleep(100)

        boolSetDriveWheelsCreate = True
    Else
                boolSetDriveWheelsCreate = False
        End If

        Return boolSetDriveWheelsCreate
End Sub
```

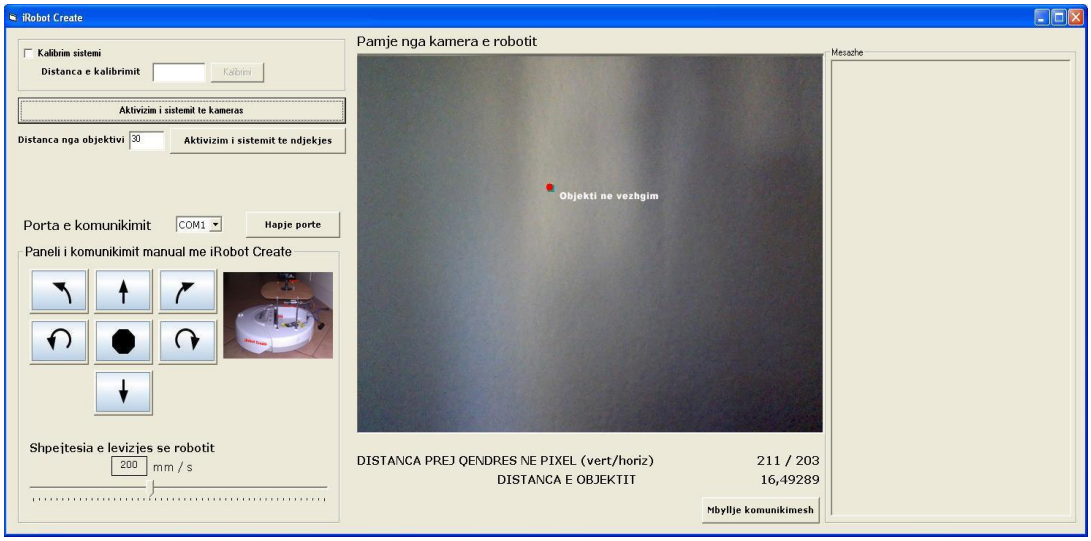
modSetDriveWheelsCreate.bas

```
Sub Process_Globals
        Public Pi    As Double = 3.1416
        Public Eps   As Double = 2.2204E-16
```

```
End Sub

Public Sub SetFwdVelRadiusRoomba(FwdVel As Double, Radius As Double) As Boolean
        Dim boolSetFwdVelRadiusRoomba As Boolean
        Dim Key                       As String
        Dim buffer()                  As Byte

        Dim maxValue                  As Double
        Dim FwdVelMM                  As Double
        Dim RadiusMM                  As Double

        Dim BYTE_HI                   As Int
        Dim BYTE_LO                   As Int

        If Main.out.IsInitialized = True Then

                maxValue = modMAin.Maxval(FwdVel, -0.5)
                FwdVelMM = modMAin.Minval(maxValue, 0.5)* 1000

                If modMAin.isinf(Radius) = 1 Then
                     RadiusMM = 32768
                 Else If Radius = Eps Then
                     RadiusMM = 1
                 Else If Radius = -Eps Then
                     RadiusMM = -1
                 Else
                      maxValue = modMAin.Maxval(1000 * Radius, -2000)
                      RadiusMM = modMAin.Minval(maxValue, 2000)
                End If

        ''''''''''''''''''''''''''''''''
        ''Komanda per levizje
        ''''''''''''''''''''''''''''''''
                Dim buffer137() As Byte = Array As Byte(137)
                Main.out.WriteBytes(buffer137,0,1)

        ''''''''''''''''''''''''''''''''
        ''FwdVelMM
        ''''''''''''''''''''''''''''''''
                BYTE_HI = modMAin.HIBYTE(FwdVelMM)
                BYTE_LO = modMAin.LOBYTE(FwdVelMM)
                Dim bufferHI1() As Byte = Array As Byte(BYTE_HI)
                Main.out.WriteBytes(bufferHI1,0,1)
                Dim bufferLO1() As Byte = Array As Byte(BYTE_LO)
                Main.out.WriteBytes(bufferLO1,0,1)


        ''''''''''''''''''''''''''''''''
        ''RadiusMM
        ''''''''''''''''''''''''''''''''
                BYTE_HI = modMAin.HIBYTE(RadiusMM)
                BYTE_LO = modMAin.LOBYTE(RadiusMM)
                Dim bufferHI2() As Byte = Array As Byte(BYTE_HI)
                Main.out.WriteBytes(bufferHI2,0,1)
                Dim bufferLO2() As Byte = Array As Byte(BYTE_LO)
                Main.out.WriteBytes(bufferLO2,0,1)


                modMAin.Sleep(100)

            boolSetFwdVelRadiusRoomba = True
        Else
           boolSetFwdVelRadiusRoomba = False
        End If
        Return boolSetFwdVelRadiusRoomba
End Sub
```

Shtojca 12

Programi i kontrollit të pozicionit të robotit Roomba 4400 (iRobot Create)



Më poshtë po japim dy funksionet kryesore llogaritëse të programit që realizojnë detyrat e parashtruara në kontrollin e pozicionit.

```
Private Sub CallProcess()
    Dim DistancaLlogaritur, DiferencaNgaDistancaKerkuar, DistancaHorizonale, KendiNeRadian As Double
    Dim ShpejtesiaKendore , ShpejtesiaLineare              As Double
    Dim oldValueDistanceMin, oldValueDistanceMax           As Double

    On Error Resume Next

    oldValueDistanceMin = 0
    oldValueDistanceMax = 10000000000#
    boolQenderPozicioniHorizontal = False
    boolPozicioniPerfundimtar = False

    Do While boolIsSTOP = False
        SendMessage hCap, GET_FRAME, 0, 0
        SendMessage hCap, COPY, 0, 0
        Picture1.Picture = Clipboard.GetData
        Clipboard.Clear
        DoEvents
        If KoeficientiLlogarites <> 0 And cKalibrimSistemi.Value = False Then

           LlogaritjeDistance Picture1
           '-----------------------------------------------------------------------------
           ' Llogarit distancen ne cm
           '-----------------------------------------------------------------------------
           DistancaLlogaritur=KoeficientiLlogarites /
gLlogaritjeDistance.pixels_prej_qendres_vertikale

           lDistanca.Caption = Format(DistancaLlogaritur, "#0.####0")
           '-----------------------------------------------------------------------------
           ' Njekje e objektivit ne menyre automatike ne dy drejtime
           '                   1. para - prapa
           '                   2. mjtas - djathtas
           ' ndjekja behet ne plan
           '-----------------------------------------------------------------------------
```

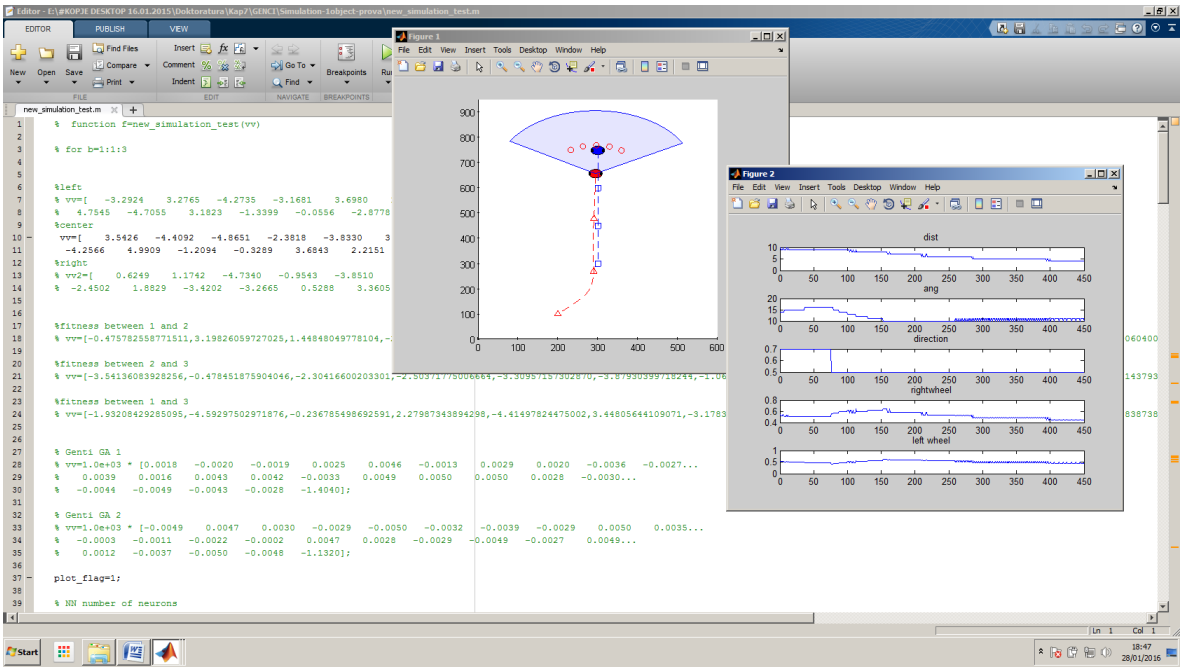
```
          If boolTracingObject = True Then
             If objComport.IsOpened = True Then
                   EndTime = Timer
                   DiferencaNgaDistancaKerkuar = (DistanceFromObject - DistancaLlogaritur)
                   DistancaHorizonale = pixel_ne_cm *
                        gLlogaritjeDistance.pixels_prej_qendres_horizontale
                   If DistancaHorizonale > epsTolerancaNgaQendra And boolQenderPozicioniHorizontal
                        = False Then
                        KendiNeRadian = Atn(Abs(DistancaHorizonale) / Abs(DistancaLlogaritur))
                        ShpejtesiaKendore = Abs(KendiNeRadian) / (EndTime - StartTime)
                        ShpejtesiaLineare = ShpejtesiaKendore * distancamidisrotave / 2
                        ShpejtesiaLineare = Format(ShpejtesiaLineare, "#0.#0")
                        If ShpejtesiaLineare = 0 Then ShpejtesiaLineare = 0.005
                        SliderShpejtesia.Value = ShpejtesiaLineare * 100
                        If ObjectIsLeftRightCenter = majtas Then
                             ' levizja e robotit duhet kryhet majtas
                               SetDriveWheelsCreate -ShpejtesiaLineare, ShpejtesiaLineare
                               HistorikLevizje = majtas
                               lMessage.AddItem "Levizja majtas (-) " & ShpejtesiaLineare
                        ElseIf ObjectIsLeftRightCenter = djathtas Then
                             ' levizja e robotit duhet kryhet djathtas
                               SetDriveWheelsCreate ShpejtesiaLineare, -ShpejtesiaLineare
                               HistorikLevizje = djathtas
                               lMessage.AddItem "Levizja djathtas (-) " & ShpejtesiaLineare
                        ElseIf ObjectIsLeftRightCenter = ndaluar Then
                             ' stop the robot
                               SetFwdVelRadiusRoomba 0, 1
                               HistorikLevizje = ndaluar
                               lMessage.AddItem "Qendrim ne vend () " & ShpejtesiaLineare
                        End If
                        lMessage.ListIndex = lMessage.ListCount - 1
                        boolQenderPozicioniHorizontal = False
                    Else
                        ''' u pozicionua ne qender roboti
                        boolQenderPozicioniHorizontal = True
                    End If

                    If boolQenderPozicioniHorizontal = True And boolPozicioniPerfundimtar = False
Then
                        If Abs(DiferencaNgaDistancaKerkuar) > epsTolerancaDistance And _
                           (DistanceFromObject > oldValueDistanceMin And DistanceFromObject <
oldValueDistanceMax) Then
                           If Abs(DistanceFromObject - DistancaLlogariturPerpara) >
Abs(DistanceFromObject - DistancaLlogaritur) Then
                              ShpejtesiaLineare = Abs(DistanceFromObject - DistancaLlogaritur) /
(EndTime - StartTime)
                           Else
                              ShpejtesiaLineare = Abs(DistanceFromObject -
DistancaLlogariturPerpara) / (EndTime - StartTime)
                           End If
                           ShpejtesiaLineare = ShpejtesiaLineare / 100
                           ShpejtesiaLineare = Format(ShpejtesiaLineare, "#0.#0")
                           If ShpejtesiaLineare = 0 Then ShpejtesiaLineare = 0.005
                           SliderShpejtesia.Value = ShpejtesiaLineare * 100
                           If DiferencaNgaDistancaKerkuar > 0 Then
                               ' levizja e robotit duhet kryhet mbrapsh
                                 SetDriveWheelsCreate -ShpejtesiaLineare, -ShpejtesiaLineare
                                 HistorikLevizje = prapa
                                lMessage.AddItem "Levizja prara (-) " & ShpejtesiaLineare
                           ElseIf DiferencaNgaDistancaKerkuar = 0 Then
                               ' stop the robot
                                 SetFwdVelRadiusRoomba 0, 1
                                 HistorikLevizje = ndaluar
                                lMessage.AddItem "Qendrim ne vend ()  " & ShpejtesiaLineare
                           ElseIf DiferencaNgaDistancaKerkuar < 0 Then
```

```
                            ' levizja e robotit duhet kryhet para
                              SetDriveWheelsCreate ShpejtesiaLineare, ShpejtesiaLineare
                              HistorikLevizje = para
                              lMessage.AddItem "Levizja para (+) " & ShpejtesiaLineare
                        End If
                        lMessage.ListIndex = lMessage.ListCount - 1
                        DistancaLlogariturPerpara = DistancaLlogaritur
                        If DistancaLlogariturPerpara < DiferencaNgaDistancaKerkuar Then
                           oldValueDistanceMin = DistancaLlogariturPerpara
                        ElseIf DistancaLlogariturPerpara > DiferencaNgaDistancaKerkuar Then
                           oldValueDistanceMax = DistancaLlogariturPerpara
                        End If
                        boolPozicioniPerfundimtar = False
                    Else
                        ' stop the robot
                          SetFwdVelRadiusRoomba 0, 1
                          HistorikLevizje = ndaluar
                          lMessage.AddItem "Qendrim ne vend ()  " & ShpejtesiaLineare
                          boolPozicioniPerfundimtar = True
                    End If
                End If
            Else
               lMessage.AddItem "Porta e komunikimit me iRobot Create e pa hapur."
            End If
        End If
    End If
    DoEvents
    StartTime = Timer
    DoSleep 0.3
  Loop
  Err.Clear
End Sub

Public Function LlogaritjeDistance(Pic As PictureBox) As Boolean
    Dim gs                                  As Long
    Dim bytes_per_scanLine              As Long
    Dim pad_per_scanLine                As Integer
    Dim x, y                            As Long
    Dim i, j                            As Integer
    Dim red, green, blue                As Integer
    Dim gRed, gGreen, gBlue      As Integer
    Dim pixel                    As Long
    Dim pixels_prej_qendres_horizontale  As Long
    Dim pixels_prej_qendres_vertikale    As Long
    Dim max_ColorSelected, max_ColorSelected2   As Long

    max_ColorSelected = 0
    max_ColorSelected2 = 0

    Dim bitmap_info    As BITMAPINFO
    Dim pixels()       As Byte

    Const pixR As Integer = 3: Const pixG As Integer = 2: Const pixB As Integer = 1

    Dim ColorSelected_rreshti, ColorSelected_kolona      As Integer
    Dim ColorSelected_rreshti2, ColorSelected_kolona2    As Integer

    ' Pic is the image to be converted
    Pic.ScaleMode = vbPixels

    x = Pic.ScaleWidth
    y = Pic.ScaleHeight

    ReDim pixels(1 To 4, 1 To Pic.ScaleWidth, 1 To Pic.ScaleHeight) As Byte
```

```
    ' Prepare the bitmap description.
    With bitmap_info.bmiHeader
        .biSize = 40
        .biWidth = Pic.ScaleWidth
        ' Use negative height to scan top-down.
        .biHeight = -Pic.ScaleHeight
        .biPlanes = 1
        .biBitCount = 32
        .biCompression = BI_RGB
        bytes_per_scanLine = ((((.biWidth * .biBitCount) + 31) \ 32) * 4)
        pad_per_scanLine = bytes_per_scanLine - (((.biWidth * .biBitCount) + 7) \ 8)
        .biSizeImage = bytes_per_scanLine * Abs(.biHeight)
    End With
    GetDIBits Pic.hdc, Pic.Image, 0, Pic.ScaleHeight, pixels(1, 1, 1), bitmap_info, DIB_RGB_COLORS
    For j = 1 To y - intstepcheck Step intstepcheck
        For i = 1 To x - intstepcheck Step intstepcheck
            If pixels(pixR, i, j) > max_ColorSelected Then
                max_ColorSelected = pixels(pixR, i, j)
                ColorSelected_rreshti = j
                ColorSelected_kolona = i
            End If
        Next
    Next
    ColorSelected_rreshti = ColorSelected_rreshti + intstepcheck
    ColorSelected_kolona = ColorSelected_kolona + intstepcheck
    lXY(0).Caption = ColorSelected_rreshti
    lXY(1).Caption = ColorSelected_kolona
    ' llogarit distancen nga mesi i figures
    Dim x_gender, y_gender
    x_gender = CInt(x / 2)
    y_gender = CInt(y / 2)
    ColorSelected_rreshti2 = ColorSelected_rreshti + 2 * intstepcheck
    ColorSelected_kolona2 = ColorSelected_kolona + 2 * intstepcheck
    pixels_prej_qendres_vertikale = Abs(y_gender - ColorSelected_rreshti -
CInt((ColorSelected_rreshti2 - ColorSelected_rreshti) / 2))
    pixels_prej_qendres_horizontale = Abs(x_gender - ColorSelected_kolona -
CInt((ColorSelected_kolona2 - ColorSelected_kolona) / 2))
    Select Case x_gender - ColorSelected_kolona - CInt((ColorSelected_kolona2 -
ColorSelected_kolona) / 2)
    Case Is > 0
        ObjectIsLeftRightCenter = majtas
    Case Is < 0
        ObjectIsLeftRightCenter = djathtas
    Case 0
        ObjectIsLeftRightCenter = ndaluar
    End Select
    lDistancaPixelNgaQendra.Caption = pixels_prej_qendres_vertikale & " / " &
pixels_prej_qendres_horizontale
    If ColorSelected_kolona > 0 And ColorSelected_rreshti > 0 Then
        ' vija vertikale dhe horizontale
        If ColorSelected_rreshti > 5 And ColorSelected_kolona > 5 Then
            For j = ColorSelected_rreshti - intstepcheck To ColorSelected_rreshti2 + intstepcheck
                For i = ColorSelected_kolona - intstepcheck To ColorSelected_kolona2 + intstepcheck
                    pixels(pixR, i, j) = 0
                Next
            Next
        End If

        For j = y / 2 - 5 To y / 2 + 5
            pixels(pixR, x_gender, j) = 127
        Next

        For i = x / 2 - 5 To x / 2 + 5
            pixels(pixR, i, y_gender) = 127
        Next
```

```
        ' pamja e fundit
        SetDIBits Pic.hdc, Pic.Image, 0, Pic.ScaleHeight, pixels(1, 1, 1), bitmap_info,
DIB_RGB_COLORS
        Pic.Picture = Pic.Image
    End If
    gLlogaritjeDistance.pixels_prej_qendres_horizontale = pixels_prej_qendres_horizontale
    gLlogaritjeDistance.pixels_prej_qendres_vertikale = pixels_prej_qendres_vertikale
    LlogaritjeDistance = True
    Err.Clear
End Function
```

Shtojca 13



Disa prej funksioneve që realizojne simulimet dhe ato në kohë reale po i japim në vazhdim.

Moduli kryesor

```
% majtas
% vv=[    -3.2924     3.2765    -4.2735    -3.1681     3.6980     1.6085     2.4807    -1.2793...
%    4.7545    -4.7055     3.1823    -1.3399    -0.0556    -2.8778     1.5047];
% qender
 vv=[     3.5426    -4.4092    -4.8651    -2.3818    -3.8330     3.6298    -2.2900     3.8378...
   -4.2566     4.9909    -1.2094    -0.3289     3.6843     2.2151    -3.4542];
% djathtas
% vv2=[     0.6249     1.1742    -4.7340    -0.9543    -3.8510     2.9401    -4.2097     1.5991...
%   -2.4502     1.8829    -3.4202    -3.2665     0.5288     3.3605    -2.0689];


plot_flag=1;

% NN numri i neuroneve
nr_i=3;
nr_h=3;
nr_o=2;

R1_total_sa1_ang=0;
R1_total_f_dist=0;

in1_di=[];
in1_a=[];
in1_d=[];

in2_di=[];
in2_a=[];
in2_d=[];

out1_r=[];
```

```
out1_l=[];

out2_r=[];
out2_l=[];


% robot strukture
rob(1)=struct('x',300,'y',300,'th',pi/2,'r',17,'trajectory',[],'himg',[],'color','b');
rob(2)=struct('x',200,'y',100,'th',pi/4,'r',17,'trajectory',[],'himg',[],'color','r');

% Destination
gool(1)=struct('dist',5,'ang',5,'himg',[]);
gool(2)=struct('dist',5,'ang',8,'himg',[]);
gool(3)=struct('dist',5,'ang',11,'himg',[]);
gool(4)=struct('dist',5,'ang',14,'himg',[]);
gool(5)=struct('dist',5,'ang',17,'himg',[]);

num_cr=size(rob,2)-1;
field=[0,600,0,800];        % [xmin,xmax,ymin,ymax]

% Inicializimi grafik
if plot_flag==1
    hfig=InitFig(field);            % Inicializimi i figures
    hcam=[];                        % Handle of the visual range of the camera image
    rob(1)=DrawRob(hfig,rob(1));    % roboti lider
    rob(2)=DrawRob(hfig,rob(2));    % qëllimi
end

max_step=450;   % Numri max i hapave
dt=0.05;        % koha e llogaritjes per levizjen gjate nje hapi

%
t_p_x1=[];
t_p_y1=[];
t_p_x2=[];
t_p_y2=[];

step=0;

while(step<max_step)
    if step==0
        plot(rob(1).x,rob(1).y,'s','MarkerSize',6,'MarkerEdgeColor','b');
        plot(rob(2).x,rob(2).y,'^','MarkerSize',6,'MarkerEdgeColor','r');
    end

    step=step+1;
    % (Range: 1 out of range visible 0) robot recognized leader in visual Robo
    [visible,ang,dist]=CameraSens_test(rob);
    % Orientation of the robot (.3 .5 .7 -1)
    direction=RobDirection_new(rob);

    in1_di=[in1_di,dist(1)];
    in1_a=[in1_a,ang(1)];
    in1_d=[in1_d,direction(1)];

    for i=1:num_cr
        if visible(i)==0;
            dist(i)=1;
            ang(i)=0;
            direction(i)=-1;
        end
        Input=[dist(i)/10,ang(i)/21,direction(i)];                  % NN input
        Output=nn_A(Input,vv,nr_i,nr_h,nr_o);                       % NN calculation
        rob(i+1)=MoveiRob_t(rob(i+1),Output(1)*50,Output(2)*50,dt);  % Robot move
    end
    out1_r=[out1_r,Output(1)];
    out1_l=[out1_l,Output(2)];
```

```
    % Movement of leader
    rob(1)=MoveiRob_t(rob(1),20,20,dt);

    % Graph depiction
    if plot_flag==1
        hcam=DrawCamera_2(hfig,hcam,rob);      % Camera vision
        gool=DrawGool_2(hfig,rob,gool);        % gool
        rob(1)=DrawRob(hfig,rob(1));           % rob
        rob(2)=DrawRob(hfig,rob(2));
        plot(t_p_x1,t_p_y1,'s','MarkerSize',6,'MarkerEdgeColor','b');
        plot(t_p_x2,t_p_y2,'^','MarkerSize',6,'MarkerEdgeColor','r');
        drawnow
    end

     if mod(step,150)==0
         t_p_x1=[t_p_x1, rob(1).x];
         t_p_y1=[t_p_y1, rob(1).y];
         t_p_x2=[t_p_x2, rob(2).x];
         t_p_y2=[t_p_y2, rob(2).y];
     end
end

if plot_flag==1
        figure(2)
        subplot(5,1,1)
        plot(1:450,in1_di(1:450))
        title('dist')
        subplot(5,1,2)
        plot(1:450,in1_a(1:450))
        title('ang')
        subplot(5,1,3)
        plot(1:450,in1_d(1:450))
        title('direction')
        subplot(5,1,4)
        plot(1:450,out1_r(1:450))
        title('rightwheel')
        subplot(5,1,5)
        plot(1:450,out1_l(1:450))
        title('left wheel')
 end
```

DrawGool.m

```
function gool=DrawGool(hfig,rob,gool)
num_g=size(gool,2);
max_lange=250;      % Distanca e kerkuar
view_ang=pi*2/3;    % Kendi i pamjes
div_ang=21;         % Numri pjestues i kendit te pamjes
div_dist=10;        % Numri pjesetues I distances

for i=1:num_g
    real_dist=gool(i).dist*max_lange/div_dist-(max_lange/div_dist)/2;
    real_ang=-view_ang/2+gool(i).ang*view_ang/div_ang-(view_ang/div_ang)/2;

    if isempty(gool(i).himg)
       gool(i).himg=plot(rob(1).x+real_dist*sin(rob(1).th+real_ang),rob(1).y+real_dist*cos(
       rob(1).th+real_ang),'b:o','Parent',hfig);
    end
set(gool(i).himg,'Xdata',rob(1).x+real_dist*cos(rob(1).th+real_ang),'Ydata',rob(1).y+real_di
st*sin(rob(1).th+real_ang));
end
```

nn_A.m

```
function [action]=nn_A(Input_V,vv,nr_i,nr_h,nr_o)
for i=0:nr_h-1
    W_i_h(i+1,:)=vv(i*nr_i+1:i*nr_i+nr_i);
end
for i=0:nr_o-1
    W_h_o(i+1,:)=vv(nr_h*nr_i+i*nr_h+1:nr_h*nr_i+i*nr_h+nr_h);
end
for i=1:nr_h
    sum_h = 0.0;
    for j = 1:nr_i
            sum_h = sum_h+W_i_h(i,j) * Input_V(j);
    end
    out_h(i) = 1.0 / (1.0 + exp(-sum_h));
end
for i = 1:nr_o
    sum_o = 0;
    for j=1:nr_h
            sum_o = sum_o + W_h_o(i,j) *out_h(j);
    end
    p(i) = 1.0 / (1.0 + exp(-sum_o));
end
action=p;
```

DrawCamera.m

```
function hcam=DrawCamera(hfig,hcam,rob)
max_lange = 250;       % distance e njohur
view_ang = pi*2/3;     % kendi i pamjes
num_div = 21;          % numri pjesetues per kendin e pamjes

div_ang=linspace(-view_ang/2,view_ang/2,num_div);

if (isempty(hcam))
hcam=patch([rob(1).x,rob(1).x+max_lange*cos(rob(1).th+div_ang)],[rob(1).y,rob(1).y+max_lange
*sin(rob(1).th+div_ang)],'b','EdgeColor','b','FaceAlpha',0.1,'Parent',hfig);
end

set(hcam,'Xdata',[rob(1).x,rob(1).x+max_lange*cos(rob(1).th+div_ang)],'Ydata',[rob(1).y,rob(
1).y+max_lange*sin(rob(1).th+div_ang)]);
```

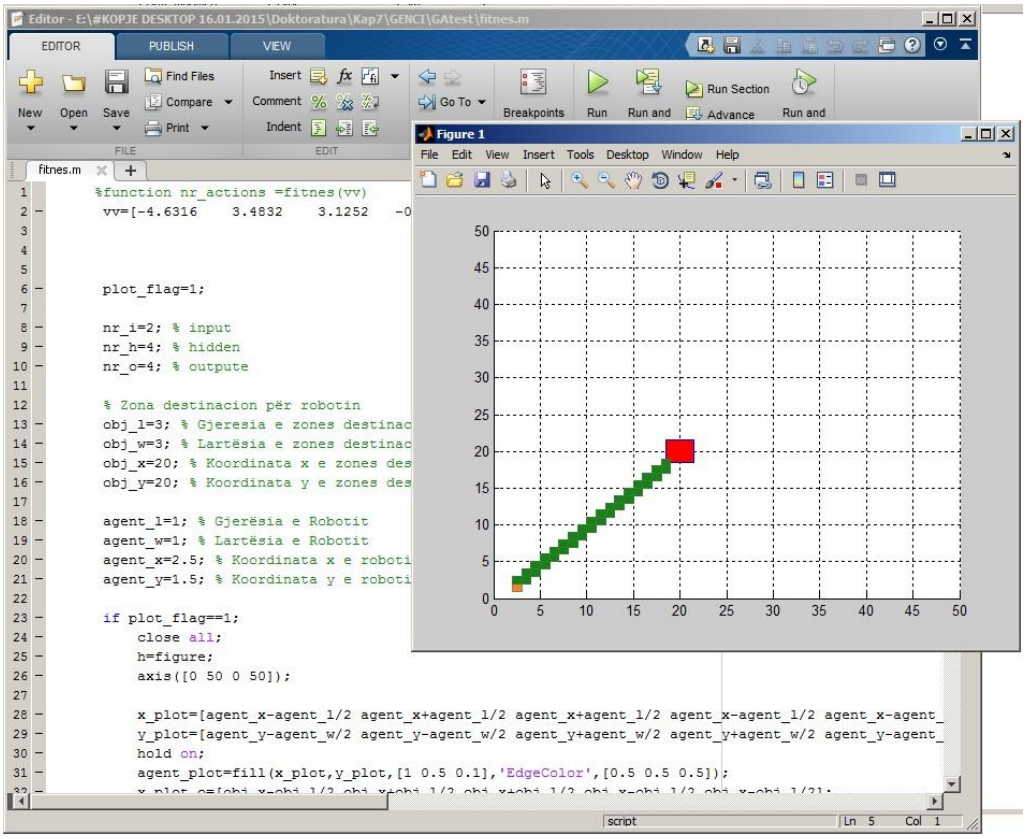
CameraSens_test.m

```
function [visible,par_ang,par_dist]=CameraSens_test(rob)
num_r=size(rob,2);  % Numri I roboteve
max_lange=250;      %
view_ang=pi*2/3;    %
div_ang=21;         %
div_dist=10;        %
a=rand(1);

if a<0.5
    a=1;
else
    a=2;
end

visible=zeros(num_r-1,1);
par_ang=zeros(num_r-1,1);
par_dist=zeros(num_r-1,1);

for i=2:1:num_r
    % Distanca ndermjet roboteve
    dist_t=sqrt((rob(i).x-rob(1).x)^2+(rob(i).y-rob(1).y)^2);
```

```
    % Korrigim distance per gabimin pas perpunimit grafik
    dist=dist_t+(-1)^a*0.3268*rand(1)*exp(0.0138*dist_t);

    ang=pi+atan2(rob(i).y-rob(1).y,rob(i).x-rob(1).x)-rob(i).th;
    if (ang>pi)
        ang=ang-2*pi;
    elseif (ang<-pi)
        ang=ang+2*pi;
    end

    % Kontrolloni nëse hyri në fushën e pamjes së liderit
    if (dist<=max_lange && ang<view_ang/2 && ang>-view_ang/2);
        visible(i-1)=1;
    end

    % Ndaj kendin. Psh: I-pi / 4 ~ pi / 4 = 1 ~ 16
    par_ang(i-1)=ceil((ang+view_ang/2)/(view_ang/div_ang));
    % Ndaj distancen. Psh: I 0-250 = 0-10
    par_dist(i-1)=ceil(dist/(max_lange/div_dist));

end
```

Shtojca 14

Programi për orientimin e lëvizjes së robotit drejt një zone të paracaktuar me ndihmën e rrjetave neuronike (NN).




```
% function nr_actions =fitnes(vv)

% peshat e NN
 vv=[-4.6316    3.4832    3.1252   -0.7172    4.5325   -4.5294    4.3658    2.3638    3.3242
        0.6160 -1.7243    3.0580    4.7748   -4.3653    3.9131   -4.6044    4.4511    0.1749
        0.1807    1.7906 4.5619    2.8825   -3.7450   -0.9865  -32.0000];


   plot_flag=1;

   nr_i=2; % input
   nr_h=4; % hidden
   nr_o=4; % outpute

   % Zona destinacion për robotin
   obj_l=3; % Gjeresia e zones destinacion
   obj_w=3; % Lartësia e zones destinacion
   obj_x=20; % Koordinata x e zones destinacion (cepi poshte majtas ne xy)
   obj_y=20; % Koordinata y e zones destinacion (cepi poshte majtas ne xy)

   agent_l=1; % Gjerësia e Robotit
   agent_w=1; % Lartësia e Robotit
```

```
agent_x=2.5; % Koordinata x e robotit
agent_y=1.5; % Koordinata y e robotit

if plot_flag==1;
    close all;
    h=figure;
    axis([0 50 0 50]);

    x_plot=[agent_x-agent_l/2 agent_x+agent_l/2 agent_x+agent_l/2 agent_x-agent_l/2
            agent_x-agent_l/2];
    y_plot=[agent_y-agent_w/2 agent_y-agent_w/2 agent_y+agent_w/2 agent_y+agent_w/2
            agent_y-agent_w/2];
    hold on;
    agent_plot=fill(x_plot,y_plot,[1 0.5 0.1],'EdgeColor',[0.5 0.5 0.5]);
    x_plot_o=[obj_x-obj_l/2 obj_x+obj_l/2 obj_x+obj_l/2 obj_x-obj_l/2 obj_x-obj_l/2];
    y_plot_o=[obj_y-obj_w/2 obj_y-obj_w/2 obj_y+obj_w/2 obj_y+obj_w/2 obj_y-obj_w/2];
    hold on;
    obj_plot=fill(x_plot_o,y_plot_o,[1 0 0],'EdgeColor',[0 0 1]);
    grid;
end
nr_actions=0;
while (nr_actions<100)
   if (agent_x>=obj_x-obj_l/2 & _
       agent_x<=obj_x+obj_l/2 & _
       agent_y>=obj_y-obj_w/2 & _
       agent_y<=obj_y+obj_w/2)

       nr_actions

       break;
   end

   Input_V=[agent_x agent_y];

   action=NN_A(Input_V,vv,nr_i,nr_h,nr_o); % - rrejta neural NN

   if action(1)==1
      agent_x = agent_x+1; % - leviz djathtas
   elseif action(2)==1
      agent_x = agent_x-1; % - leviz majtas
   elseif action(3)==1
      agent_y = agent_y+1; % - leviz lart
   elseif action(4)==1
      agent_y = agent_y-1; % - leviz poshte
   end

   nr_actions=nr_actions+1;

   if plot_flag==1
        x_plot=[agent_x-agent_l/2 agent_x+agent_l/2 agent_x+agent_l/2 agent_x-agent_l/2
                agent_x-agent_l/2];
        y_plot=[agent_y-agent_w/2 agent_y-agent_w/2 agent_y+agent_w/2 agent_y+agent_w/2
                agent_y-agent_w/2];
        hold on;
        agent_plot=fill(x_plot,y_plot,[0.1 0.5 0.1],'EdgeColor',[0.2 0.5 0.2]);
        pause
       % drawnow
   end

end
% nr_actions
```