Cuprins

C1 - Metode și tehnici de inovare și creativitate

Noțiuni fundamentale privind mecatronica.
Prototiparea rapidă la scară.
Studiu de caz: Platforma LEGO Mindstorm.

C2 - Platforme mecatronice reconfigurabile

Prototiparea rapidă la scară.
Limbajul grafic Lego Mindstorm Nxt – blocuri funcționale.
Elemente Lego de acționare și senzoriale.

C3 - Creativitate

Conceptul Learning by doing.
Conceptul trial and error.
Cursuri de tip Hands-on.

C5 - Metode interactive de lucru în grup

Analiza metodelor de lucru in grup.
Algoritmi de control.
Conceptul „6 thinking hats”

C6 - Dezvoltarea gândirii integratoare

Integrarea în natură și în tehnologie.
Gândirea integratoare, axată pe ansamblu, de tip “top-down”.
Gândirea axată pe detaliu, de tip „bottom-up”

Investeste in oameni!

FlexFORM – PROGRAM DE FORMARE PROFESIONALA FLEXIBILA PE PLATFORME MECATRONICE

1. Metode si tehnici de inovare si creativitate

www.flexform.ro

FESTO

Metode şi tehnici de inovare şi creativitate



Obiectivul general al cursului „Metode şi tehnici de inovare şi creativitate" este exploatarea potenţialului inovator al platformelor mecatronice educaţionale în scopul dezvoltării gândirii integratoare, a stimulării creativităţii, a flexibilităţii şi capacităţii de adaptare a cursantului, respectiv elevului pentru a răspunde cu operativitate nevoilor în continuă schimbare ale pieţei muncii. Cursul vizează stimularea iniţiativei privind realizarea şi utilizarea unor platforme mecatronice modulare în sustinerea activităţilor de curs şi laborator pentru diverse discipline precum: fizică, mecatronică, electronică, informatică, tehnologii etc.

Competenţe generale deprinse de cursant în urma parcurgerii acestui curs sunt:

- Cunoașterea şi înțelegerea conceptelor fundamentale impuse de tehnologia mecatronică;
- Exploatarea platformelor educaţionale în scopul stimulării transferului de cunoaştere a capacităţii de adaptare a cursantului/elevului la dinamica pieţei muncii (programarea funcţionalităţii sistemelor mecatronice, cunoașterea, dezvoltarea şi implementarea tehnicilor de control);
- Prototiparea rapidă la scară redusă a sistemelor mecatronice (realizarea de sisteme mecatronice la scară utilizând structuri modulare reconfigurabile, programarea funcţionării);
- Gândirea sistemică, integratoare şi deprinderea de a lucra în echipă;
- Capacitate de analiză şi sinteză globală a sistemelor moderne (mecatronice).

Conţinutul cursului este structurat în 5 capitole, corespunzătoare celor 5 zile, plus evaluarea:

1. Noţiuni introductive
2. Prototiparea la scară a sistemelor mecatronice
3. Creativitate. Learning by doing
4. Metode interactive de lucru în grup
5. Dezvoltarea gândirii integratoare
6. Evaluare

1. Introducere


Introducere

Este dedicată prezentării succinte a conceptelor mecatronice, necesare pentru promovarea inovării, creativității și interdisciplinarității. (A1)

Sunt vizate noțiuni privitoare la sisteme mecatronice în ansamblul lor, la prototiparea rapidă la scară, sisteme reconfigutabile, noțiuni de mecanisme și robotică, noțiuni de electronică, senzori și actuatori, notiuni de programare, control și prototiparea rapidă a controlului.

Ca suport practic este utilizat setul didactic LEGO Mindstorms NXT. (B1)

FlexFORM - Program de formare profesionala flexibila pe platforme mecatronice

Este dedicată prezentării succinte a conceptelor mecatronice, necesare pentru promovarea inovării, creativităţii şi interdisciplinarităţii. (A1)

Pe parcursul activităţilor se pune accentul atât pe conținutul ştiinţific transmis cât şi pe modalitatea în care acesta este transmis cursanţilor. Forma în care este susţinut acest curs poate reprezenta un punct de plecare, o referinţă pentru cursanţi în momentul în care aceştia vor dori să încerce să aplice la rândul lor metodele şi tehnicile de inovare şi creativitate învățate.

Se dorește un conținut bogat în concepte teoretice de natură tehnică. Sunt vizate noţiuni privitoare la sisteme mecatronice în ansamblul lor, la prototiparea rapidă la scară, sisteme reconfigurabile, noţiuni de mecanisme şi robotică, noţiuni de electronică, senzori şi actuatori, noțiuni de programare, control şi prototiparea rapidă a controlului. Fiecare dintre aceste concepte este exemplificat în una sau mai multe aplicaţii practice.

Ca suport practic este utilizat setul didactic LEGO Mindstorms NXT. (B1)

Se prezintă pe scurt metodele interactive de lucru care vor fi detaliate pe parcursul cursului. Cursanţii sunt împărțiți în echipe de câte două persoane. Echipele îşi vor modifica componenţa la începutul fiecărei zile de lucru.



Conceptul de mecatronică s-a născut în Japonia la începutul deceniului al VIII-lea al secolului trecut. Termenul de „mecatronică” a fost brevetat de către concernul Yaskawa Electric Co. şi a fost utilizat pentru a descrie fuziunea tehnologică: mecanică – electronică – informatică.

Tot ceea ce numim azi un produs de înaltă tehnicitate, este în sine un produs mecatronic. Automobilul modern, maşinile-unelte cu comandă numerică, echipamentele periferice ale calculatoarelor, tehnica de telecomunicaţii, aparatura de cercetare, roboţii, aparatura biomedicală, aparatura electrocasnică etc., sunt doar câteva exemple de produse mecatronice. Practic, mecatronica este prezentă în toate domeniile de activitate, inclusiv în agricultură şi în construcţii.

Mecatronica s-a născut ca tehnologie şi a devenit foarte curând filosofie care s-a răspândit în întreaga lume, valenţele creatoare fiind confirmate în toate domeniile de activitate. Apariţia mecatronicii este rezultatul firesc al evoluţiei în dezvoltarea tehnologică. Această evoluţie este sugestiv evidenţiată în figura 1.1.

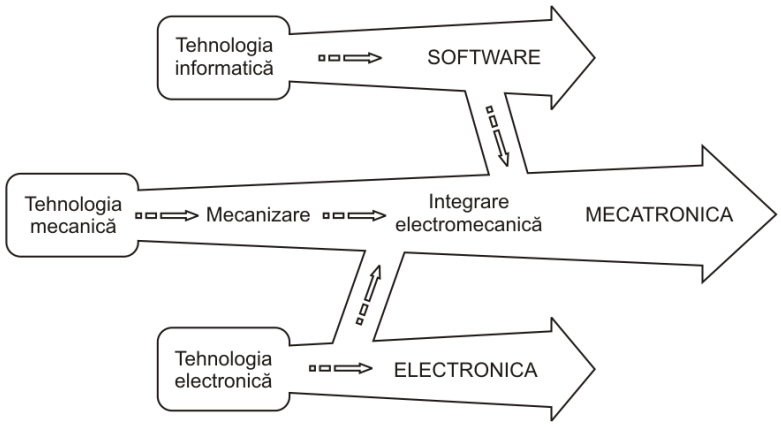


Fig. 1.1. *Fluxul tehnologic către integrarea mecatronică*

Coloana vertebrală a mecatronicii o constituie tehnologia mecanică, care s-a dezvoltat către mecanizare. Progresele în domeniul tehnologiei electronice, apariţia circuitelor integrate, mici ca dimensiuni, ieftine şi fiabile, au permis includerea electronicii în structurile mecanice. Se realizează astfel primul pas către integrare: integrarea electromecanică.

Următorul pas în integrare a fost determinat de apariţia microprocesoarelor. Cu aceleaşi caracteristici constructive ca şi circuitele integrate, microprocesoarele au putut fi integrate în structurile electromecanice realizate anterior. Astfel, acestea pot preleva informaţii privind atât starea externă, a mediului, cât si cea internă, putând prelucra şi stoca aceste informaţii şi luând decizii în consecinţă privind comportarea sistemului.

Mecatronica reprezintă o viziune integratoare în domeniul tehnologic, aşa cum este sugestiv arătat în figura 1.2. Această imagine sugerează că, în activitatea de concepţie, abordarea tradiţională în baza căreia ingineria mecanică studiază probleme specifice mişcării maselor, ingineria electrică respectiv electronică studiază probleme specifice mişcării electronilor, iar informaticienii studiază probleme specifice mişcării informaţiei, nu mai este posibilă. În structura unui produs mecatronic, practic nu se pot separa cele trei componente.

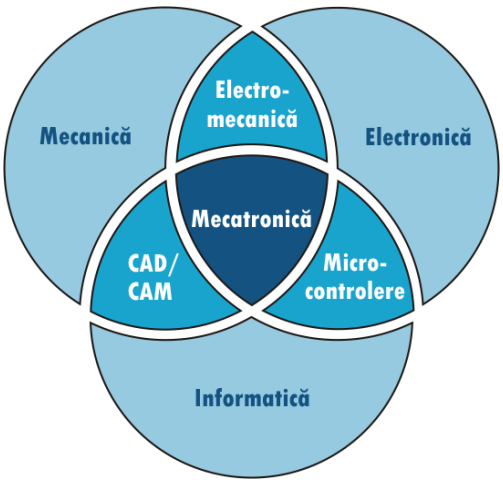


***Fig. 1.2**. Conceptul de mecatronică*



În opinia japonezilor, mecatronica este tehnologia mecanică cerută de societatea informaţională, diferenţiindu-se net de tehnologia tradiţională, în care elementele de bază erau materialul şi energia. În mecatronică, acestor două elemente li se adaugă informaţia.

În tehnologia mecatronică informaţia este componenta dătătoare de ton. Această poziţie a informaţiei în raport cu materialul şi energia este motivată de următoarele:

- informaţia asigură satisfacerea nevoilor spirituale ale omului;

- numai informaţia asigură creşterea valorii nou adăugate a tuturor lucrurilor;

- informaţia înseamnă cultură.

Pe baza figurii 1.3, se pot analiza comparativ cele trei componente ale tehnologiei mecatronice. Comparaţia are în vedere originea resurselor, rezervele, cererea şi ce înseamnă viaţa din punctul de vedere al celor trei elemente.

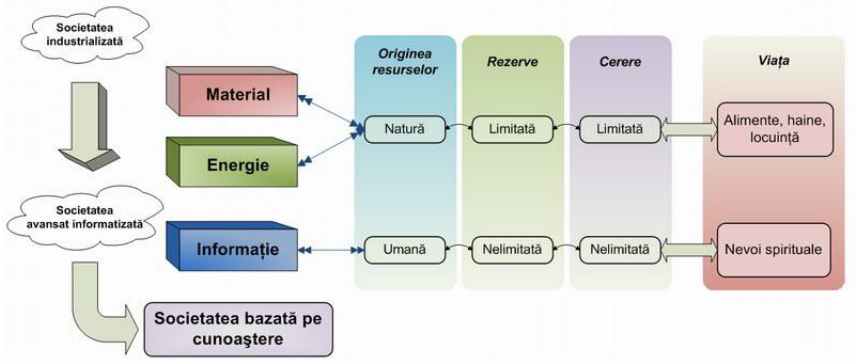


Fig. 1.3. Relaţia material-energie-informaţie în tehnologia mecatronică

Analiza motivează interesul manifestat în întreaga lume pentru promovarea acestei tehnologii. Este evident că, realizând produse care înglobează multă informaţie (inteligenţă), performanţele funcţionale ale acestora cresc. Pe de altă parte, în acest mod se conservă resursele de material şi energie. Dar, consumând mai puţin material şi energie se procesează mai puţin, deci se poluează mai puţin. Rezultă astfel alte valenţe ale tehnologiei mecatronice: este o tehnologie nedisipativă şi mai puţin poluantă.

Mecatronica în practica şi educaţia inginerească



Tehnologia mecatronică şi principiile mecatronice în educaţie au condus la definirea filosofiei mecatronice. Pentru practica inginerească această filosofie a marcat saltul de la ingineria tradiţională, secvenţială, la ingineria simultană sau concurentă (paralelă).

În figura 1.4.a se prezintă principial modul de abordare în proiectarea tradiţională iar în figura 1.4.b, proiectarea în mecatronică. În proiectarea tradiţională controlul este „ataşat" pe când în proiectarea mecatronică acesta este „integrat".

În proiectarea mecatronică, încă din faza de concepţie se are în vedere întregul. Lanţul cinematic informaţional are o structură mult mai compactă. Interconectarea prin magistrale de date permite creşterea simţitoare a vitezei de prelucrare a informaţiilor.

Educaţia mecatronică asigură flexibilitate în acţiune şi în gândire, trăsături definitorii ale specialistului în economia de piaţă. Valenţele creatoare ale mecatronicii au fost confirmate deopotrivă în educaţie, cercetare şi în producţie. Rezultatele economice ale ţărilor dezvoltate sunt dovezi de netăgăduit.

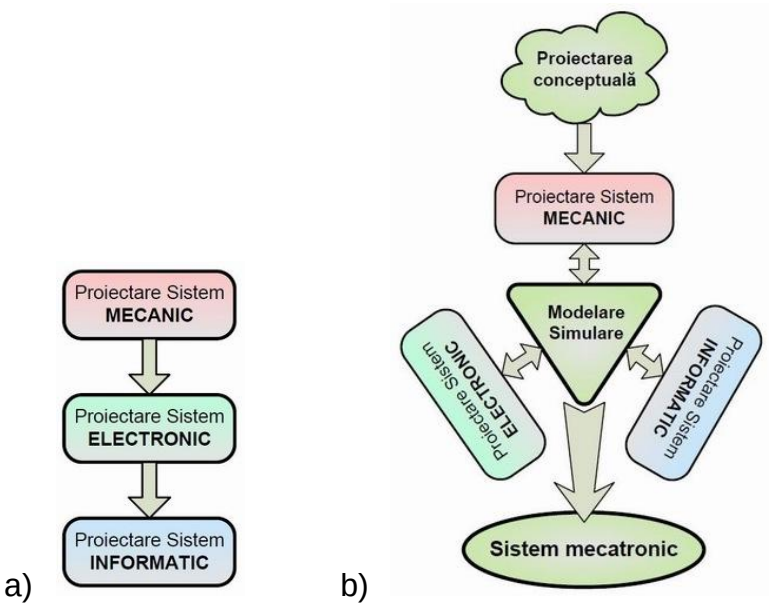


Fig. 1.4. *a) Proiectarea tradiţională, b) Proiectarea în mecatronică*

Prototiparea rapidă la scară


Investeste in oameni!

Prototiparea rapidă la scară

Obiectivele principale vizează familiarizarea cursanților cu conceptul de prototipare rapidă la scară precum și cu modulele mecatronice, respectiv componentele LEGO.

Sunt prezentate avantajele și dezavantajele prototipării rapide la scară a sistemelor mecatronice utilizând componente de tip LEGO Mindstorms.

Este prevazută construirea, pe rand, a două platforme mecatronice mobile, prima echipată cu roți și cea de-a doua de tip păşitor biped.

FlexFORM - Program de formare profesionala flexibila pe platforme mecatronice

Acest capitol începe în a doua jumătate a primei zile de lucru şi se încheie la sfârşitul celei de-a doua zile.

Obiectivele principale vizează familiarizarea cursanţilor cu conceptul de prototipare rapidă la scară precum şi cu modulele mecatronice, respectiv componentele LEGO.

Noţiuni teoretice

Sunt prezentate avantajele şi dezavantajele prototipării rapide la scară a sistemelor mecatronice utilizând componente de tip LEGO Mindstorms. Sunt expuse modurile de lucru în echipă. Se prezintă mediul de programare grafic NXT-G. Sunt evidenţiate diverse aspecte teoretice pe măsura întâlnirii acestora în cadrul aplicaţiilor practice. (A2)

Aplicaţie

Este prevăzută construirea, pe rând, a două platforme mecatronice mobile, prima echipată cu roţi şi cea de-a doua de tip păşitor biped. Sunt puse la dispoziția cursanţilor instrucţiunile de asamblare ale acestora. Se recurge la programarea treptată a funcționalității pe măsură ce structura mecanică este construită, pentru a favoriza gândirea interdisciplinară specifică mecatronicii. (B2, B3)

Pe măsura utilizării acestora, sunt expuse principiile de funcţionare a senzorilor şi actuatorilor integraţi în structură. Sunt analizate de asemenea capabilităţile controller-ului NXT. Toate aceste noţiuni vor fi reluate în detaliu în cea de-a doua zi.

Pe măsura construcţiei platformei bipede, sunt evidenţiate principiile mersului biped, programarea şi sincronizarea fazelor mersului.

La sfârșitul zilei se vor evidenţia avantajele lucrului în echipă.

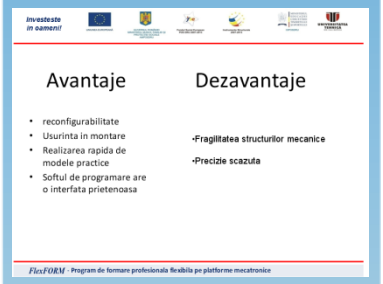
Studiu de caz: Platforma LEGO Mindstorms


Studiu de caz: Platforma LEGO Mindstorms

Kitul **Mindstorms NXT** reprezintă cea de-a doua generaţie de componente LEGO didactice. Faţă de prima versiune, oferă un nou design al componentelor mecanice, un controler programabil cu putere sporită de calcul şi un set extins de senzori. Platforma este de tip *opensource* astfel pot fi creaţi noi tipuri de senzori şi actuatori compatibili.

Kit-ul educaţional LEGO Mindstorms NXT conţine controlerul inteligent NXT, trei servomotoare, un senzor de contact, un senzor ultrasonic, un senzor de lumină şi un senzor de sunet, o baterie reîncărcabilă, conectori şi componente Lego. Aceste componente pot fi utilizate pentru a interacţiona cu mediul cu care vine în contact.

Avantajele și dezavantajele componentelor LEGO



Unul dintre avantajele pe care le au structurile Lego este reconfigurabilitatea. Aceeași componentă poate fi folosită în diferite structuri.

Un alt avantaj îl prezintă ușurința în montare. Piesele sunt compatibile între ele, indiferent de seturile folosite. Astfel, piese din seturi diferite pot fi folosite pentru aceeași structură. Tot datorită ușurinței în montare, modelele se realizează rapid.

Softul de programare are o interfață prietenoasă.

Dezavantajul structurilor construite cu componentele Lego este fragilitatea și precizia scăzută.

Descrierea programului



Construirea unui robot utilizând componente Lego implică parcurgerea a trei pași:

- Construirea robotului
- Programarea robotului
- Testarea programului

Datorita flexibilității componentelor LEGO este posibil realizarea cu acestea a unui număr mare de structuri.



Construcția unei platforme mecatronice mobile



O primă aplicație este o platformă mecatronică mobilă pe șenile.

Aceasta are în componență trei motoare și doi senzori, unul pentru detectarea culorii și unul ultrasonic.

Aplicația este folosită pentru a înlătura obstacolele, astfel: platforma se deplasează până senzorul ultrasonic detectează un obstacol, având lumina verde aprinsă; după detectarea obstacolului, senzorul de culoare își schimbă culoarea din verde în albastru, pentru atenționare, iar dacă obiectul tot nu s-a mișcat de pe traiectoria platformei, culoarea senzorului se schimbă în roșu, iar platforma aruncă cu câte o bilă până când obstacolul este înlăturat.



Baza mobila

Prima parte a construcției platformei o reprezintă baza mobilă. Pentru realizarea acesteia sunt folosite trei motoare și controlerul Lego.

Platforma se deplasează pe șenile.

Instrucțiunile pentru construirea bazei mobile se găsesc în anexe.

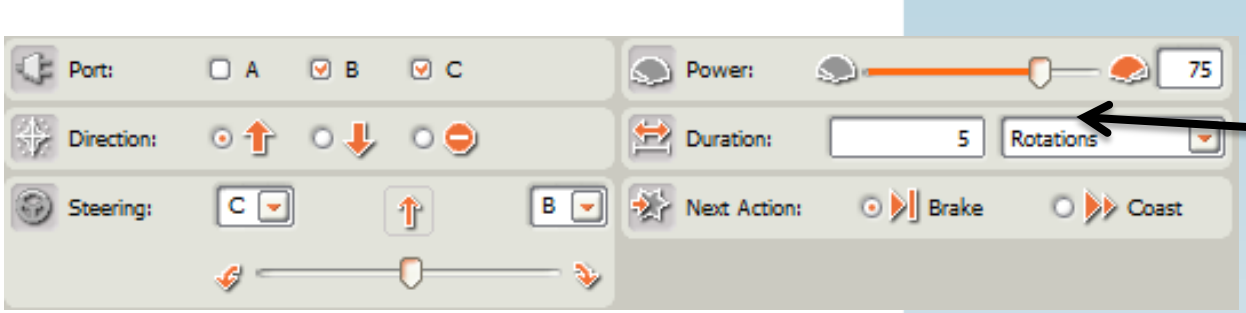


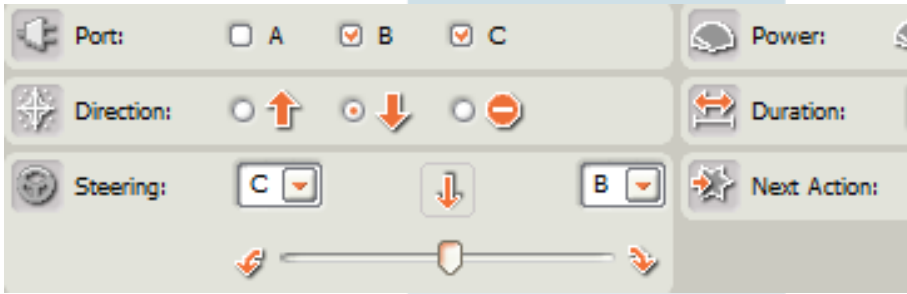


Programarea bazei mobile

În continuare sunt afișate blocurile de comandă pentru programarea bazei mobile, pentru fiecare bloc fiind afișat panoul de comandă cu setările necesare. Primul bloc execută o mișcare de avans (5 rotații pentru fiecare motor), puterea motorului fiind la 75%. Cel de-al doilea bloc are aceleași setări, diferența fiind direcția deplasării motoarelor.







Detectarea culorii

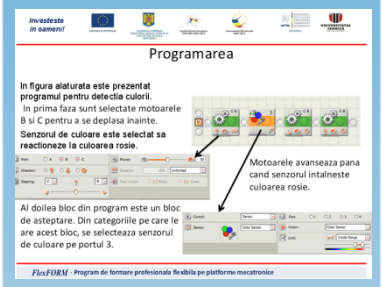
Pentru ca platforma să poată detecta culorile are nevoie de un senzor de culoare, ca și cel din figura de mai jos. Acesta se prinde pe platformă și se conectează la portul 3 al NXT-ului.







Programul pentru detectarea culorii



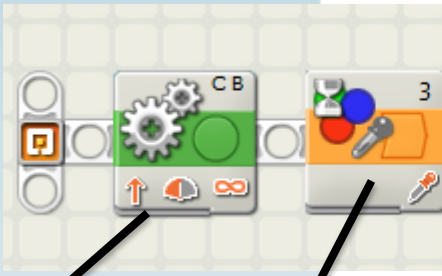
În figura alăturată este prezentat programul pentru detecția culorii.

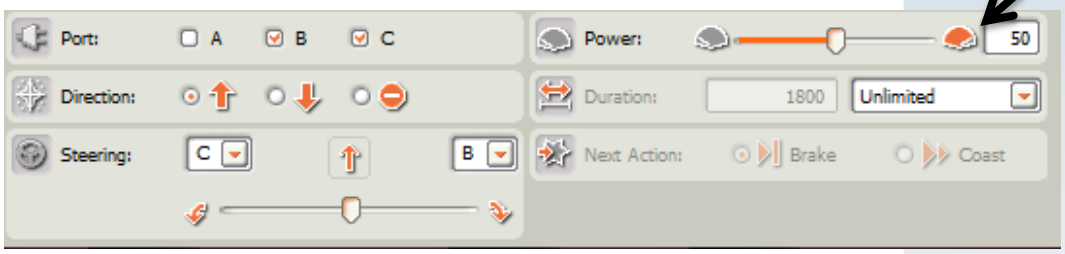
În prima fază sunt selectate motoarele B și C pentru a se deplasa înainte.

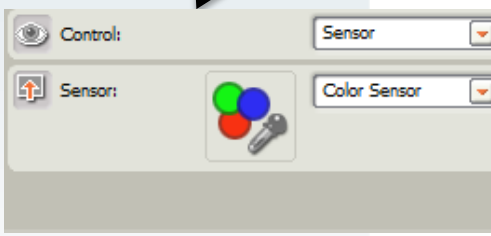
Senzorul de culoare este selectat să reacționeze la culoarea roșie.

Motoarele avansează până când senzorul întâlneşte culoarea roșie.

Al doilea bloc din program este un bloc de aşteptare. Din categoriile pe care le are acest bloc, se selectează senzorul de culoare pe portul 3.

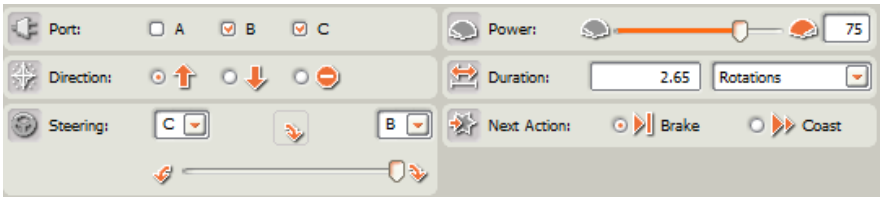




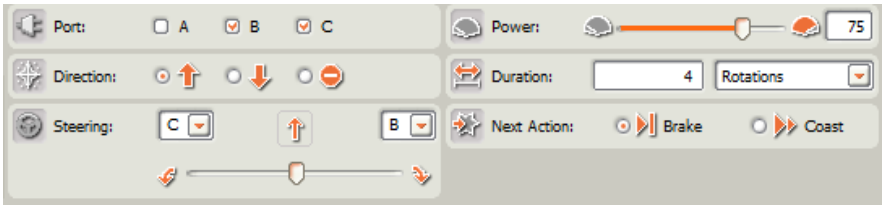


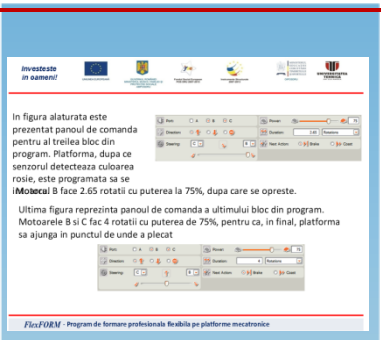
În figura alăturată este prezentat panoul de comandă pentru al treilea bloc din program. Platforma, după ce senzorul detectează culoarea roșie, este programată să se întoarcă.

Motorul B face 2.65 rotații cu puterea la 75%, după care se oprește.



Ultima figură reprezintă panoul de comandă al ultimului bloc din program. Motoarele B și C fac 4 rotații cu puterea de 75%, pentru ca, în final, platforma să ajungă în poziția de start.



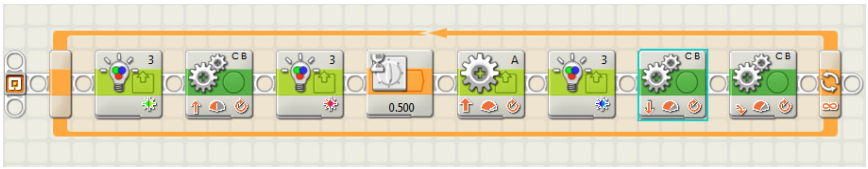




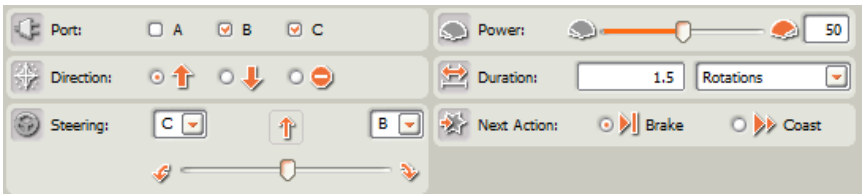
Programarea

Blocurile pentru controlul *shooter*-ului sunt prezentate în figura alăturata.

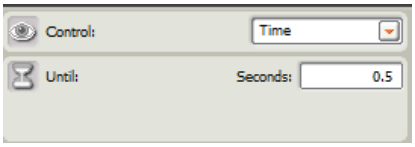
Platforma se deplasează înainte având lumina verde aprinsă. Motoarele execută o rotație şi jumătate cu o putere de 50%, după care platforma se oprește pentru 0.5 secunde. Culoarea senzorului se schimbă în roșu, după care platforma se oprește.

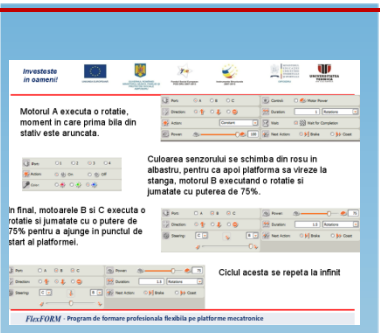












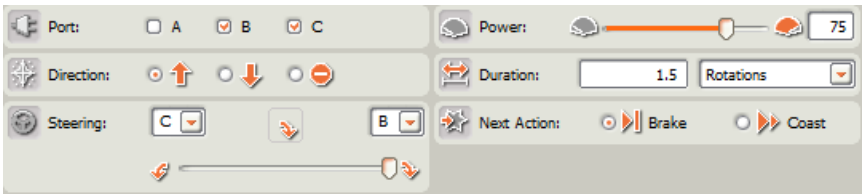
Motorul A execută o rotație, moment în care prima bilă din stativ este aruncată.



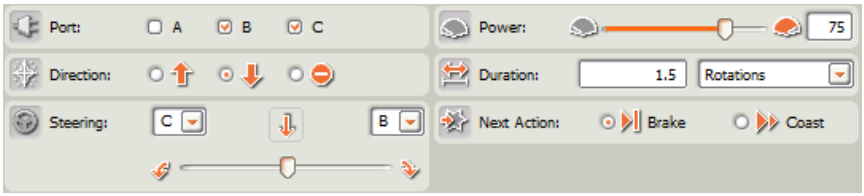
Culoarea senzorului se schimbă din roșu în albastru,



pentru ca apoi platforma să vireze la stânga, motorul B executând o rotație și jumătate cu puterea de 75%.



În final, motoarele B și C execută o rotație și jumătate cu o putere de 75% pentru a ajunge în punctul de start al platformei.

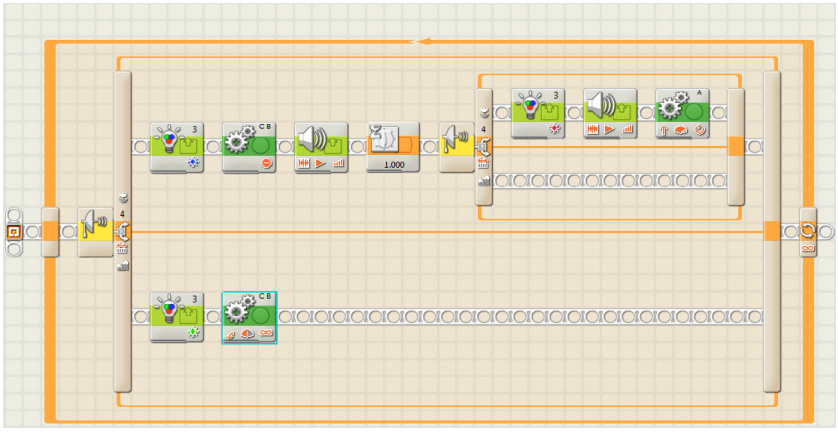


Acest ciclu se repetă la infinit.

Localizarea obiectelor



Programul pentru localizarea obiectelor este cel din figura alăturată, iar descrierea sa este prezentată mai jos.



Platforma se mișcă în cerc având lumina verde aprinsă.





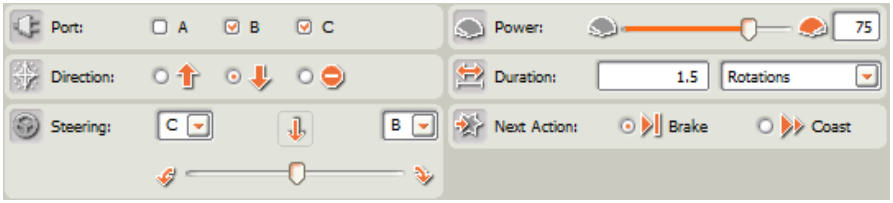
Senzorul de culoare se conectează la portul 3 și se selectează culoarea verde.

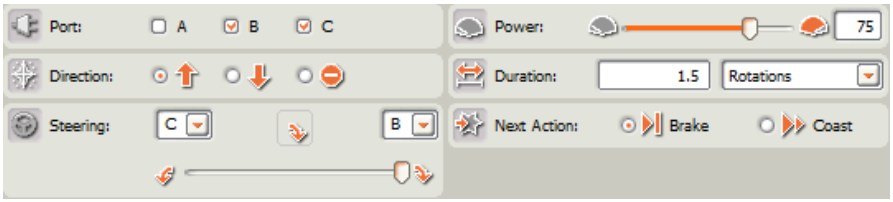
În momentul în care platforma întâlnește un obstacol la o distanță mai mică de 40 cm, platforma se oprește iar senzorul își schimbă culoarea din verde în albastru și scoate un sunet de avertizare.

Dacă obiectul nu își modifică poziția, senzorul își schimbă culoarea din albastru în roșu, iar platforma începe să arunce câte o bilă la un interval de 2 secunde până obstacolul își modifică poziția.











Senzori și actuatori aferenți platformei



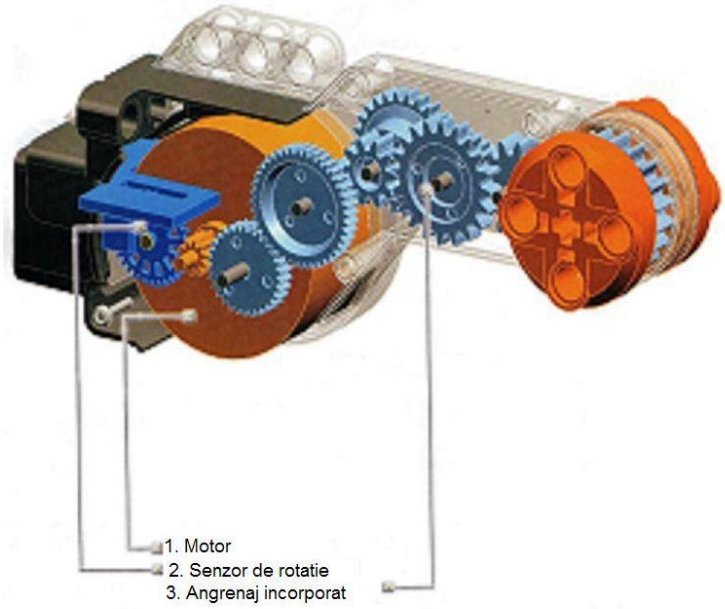
Platforma are în dotare 4 senzori, și anume: senzorul tactil, senzorul de sunet, senzorul de lumină și senzorul de ultrasunete. Cu ajutorul acestor senzori, robotul poate executa diferite comenzi, cum ar fi să reacționeze la impactul cu un anumit obstacol, să răspundă la comenzi vocale, să urmărească un anumit traseu sau să evite diverse obstacole.





Robotul se deplasează pe 3 roți, dintre care 2 sunt conduse de 2 servomotoare. Se poate deplasa astfel înainte, înapoi, și de asemenea, poate executa viraje. Cea de-a treia roată este o roată pivotantă. Al 3-lea servo motor are rolul de a acționa griperul, utilizat în operațiile de manipulare a obiectelor.

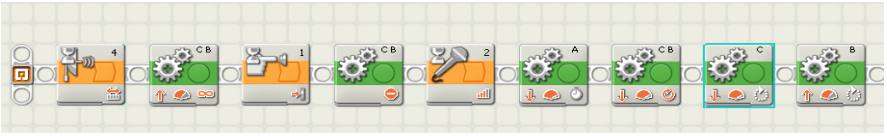
În figura alăturată se poate observa structura internă a unui servomotor LEGO.



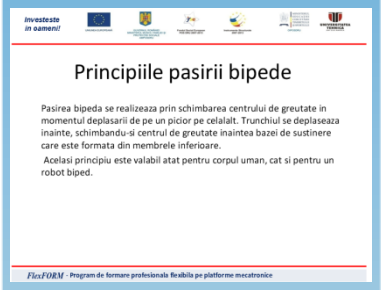
Programarea structurii



Programarea se face cu ajutorul softului Lego Mindstorms NXT. După ce programul este gata, se descărcă în NXT („creierul robotului"). Astfel robotul va executa mișcările care i-au fost comandate.



Principiile păşirii bipede



Păşirea bipedă se realizează prin schimbarea centrului de greutate în momentul deplasării de pe un picior pe celălalt. Trunchiul se deplasează înainte, schimbându-şi centrul de greutate înaintea bazei de susţinere care este formată din membrele inferioare.

Acelaşi principiu este valabil atât pentru corpul uman, cât şi pentru un robot biped.

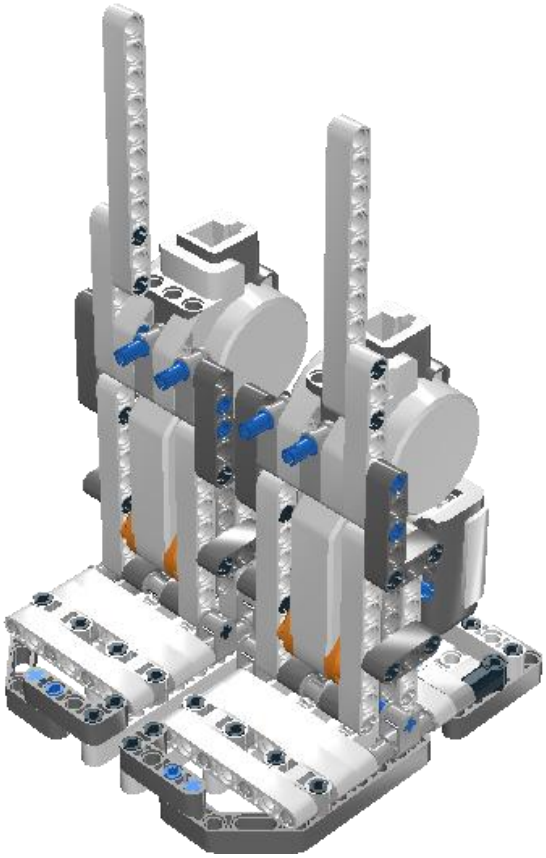
Asamblarea picioarelor după instrucțiuni



Pentru ca robotul să se poată deplasa, are nevoie de picioare.

În figura alăturată se pot observa picioarele robotului. Acestea includ in structură și două motoare, respectiv doi senzori tactili.

Instrucțiunile pentru construcția picioarelor robotului se găsesc în anexe.



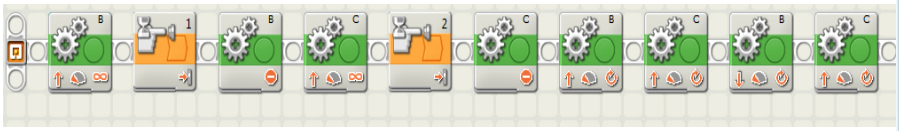
Programarea fazelor mersului



Pentru programarea robotului se folosește softul NXT-G.

Fiecare picior se programează individual, acestea fiind acționate fiecare de către un motor.

Blocurile folosite în programarea fazelor mersului sunt cele de "*move*" și "*wait*", cu ajutorul cărora robotul este programat să se deplaseze înainte, înapoi și să se rotească. Senzorii tactili folosiți pentru construcția păsitorului biped sunt folosite pentru oprirea robotului în momentul apăsării acestora.



FlexFORM – PROGRAM DE FORMARE PROFESIONALA FLEXIBILA PE PLATFORME MECATRONICE

2. Platforme mecatronice reconfigurabile



2. Prototiparea rapidă la scară

Se urmăreşte familiarizarea cursanţilor cu detaliile constructiv-funcţionale ale elementelor şi modulelor LEGO inteligente: controller, senzori, actuatori. Se prezintă importanţa interfeţei cu utilizatorul precum şi aspectele privind comunicarea om – maşină respectiv maşină – maşină.

La sfârşitul acestei zile, cursanţii vor fi capabili să descrie funcţionarea diverselor tipuri de senzori, a actuatorilor şi a controlerului LEGO NXT împreună cu componentele funcţionale ale acestuia – ecran, tastatură, porturi de intrare/ieşire, modul radio, difuzor. Se pune accentul şi pe capacitatea cursanţilor de a realiza o replică functională a unei structuri date, fără a avea la îndemână instrucţiuni detaliate de montaj.

Noțiuni teoretice:

Se prezintă principiile programării în medii grafice, cu accent pe mediul NXT-G. Este prezentată structura hardware a LEGO Mindstorms NXT.

Se trec în revistă următoarele componente, atât din punct de vedere teoretic cât şi cu ajutorul unei aplicaţii practice, pentru a evidenţia funcţionalitatea:

- **Interfaţa cu utilizatorul**

Sisteme vizuale; gestiunea ecranului

Sisteme auditive; generarea sunetelor

- **Interfaţarea senzorilor. Achiziţii de date**

Utilizarea practică a senzorilor: ultrasonic (distanţă), tactil, senzor de lumină, senzor de tip busolă, senzor de acceleraţie

Afișarea informaţiilor de la senzori pe ecran

Stocarea şi afişarea informaţiilor provenite de la senzori sub formă de histogramă

- **Interfaţarea actuatorilor**

Aspecte privind utilizarea practică a actuatorilor

Servomotoare

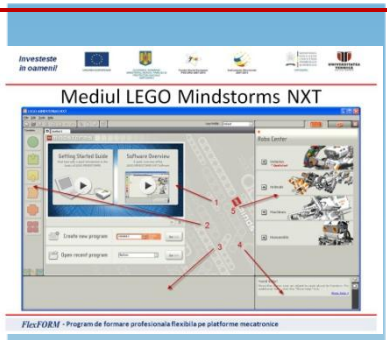
Sisteme de feed-back

- **Comunicaţii la distanţă. Telegestiune**

Modulul Bluetooth (radio) din dotarea controllerului NXT

Programarea comunicaţiei radio

Proiectarea unui sistem Master-Slave

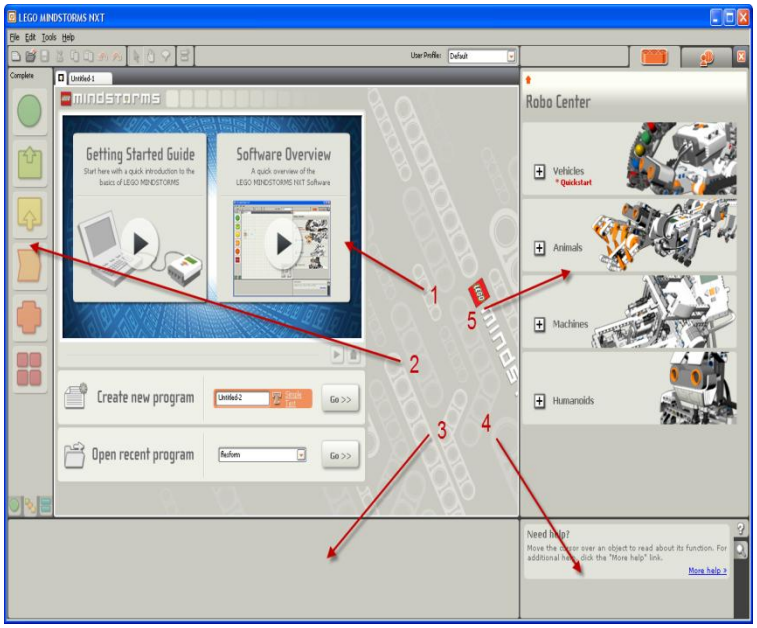


Aplicaţie:

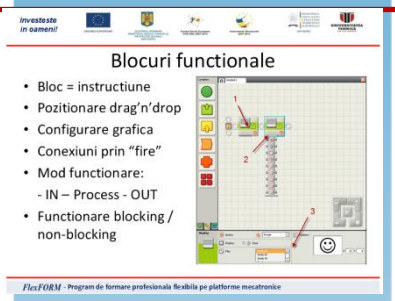
Se cere realizarea unei platforme omnidirecţionale. Această platformă va solicita aplicarea integrată a tuturor noţiunilor teoretice invăţate pană acum. Va fi pus la dispoziţie un model al structurii mecanice, urmând ca fiecare echipă să realizeze, pe baza experienţei de lucru acumulate în prima zi, o copie funcţională a acesteia.

Se discută importanta experienţei acumulate de fiecare echipă în prima zi.

Se discută despre controlul motoarelor şi sincronizarea acestora. Se implementează un algoritm de control de la distanţă a platformei mobile.



Blocuri funcționale



Blocul funcțional reprezintă elementul de bază al limbajului grafic LEGO Mindstorms NXT. Este derivat din conceptul de bloc avansat al limbajului Labview.

Fiecare bloc este echivalent cu o instrucțiune (subrutină) a limbajelor bazate pe text (C++, basic, pascal etc.).

Sunt împărțite în grupuri cu funcționalități asemănătoare: Comune, Acțiuni, Sensori, Control, Date, Avansați. În plus față de acestea, utilizatorul are posibilitatea de a crea noi blocuri (user-defined).

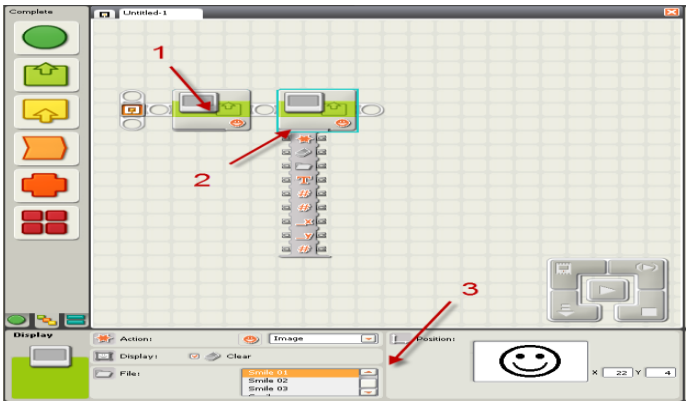
Blocurile pot fi depuse pe schema aplicației printr-o operațiune de tip "Drag-and-Drop", de-a lungul firului de execuție a programului (indcația 1).

În momentul în care un bloc este selectat, în partea de jos a ferestrei apar proprietățile acestuia, care pot fi configurate pe cale grafică (indcația 3). Parametrii de configurare diferă in funcție de tipul blocului selectat.

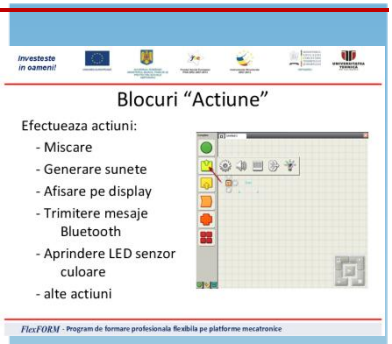
Parametrii unui bloc pot fi setați și "on-line" în timpul execuției programului cu ajutorul conexiunilor acestuia. Pentru a avea acces la grila de conexiuni a blocului, se va efectua un click la baza acestuia (indcația 2).

Modul de operare a blocurilor se bazează pe principiul IN-> Process -> OUT. Se așteaptă apariția tuturor datelor de intrare necesare (IN); Se efectuează operațiile necesare, în funcție de tipul blocului și valorile de intrare (Process); Se validează datele de ieșire, dacă este cazul (OUT).

În funcție de tipul acțiunii "Process", blocurile pot să suspende execuția programului (blocking type) sau să se execute instantaneu (non-blocking).



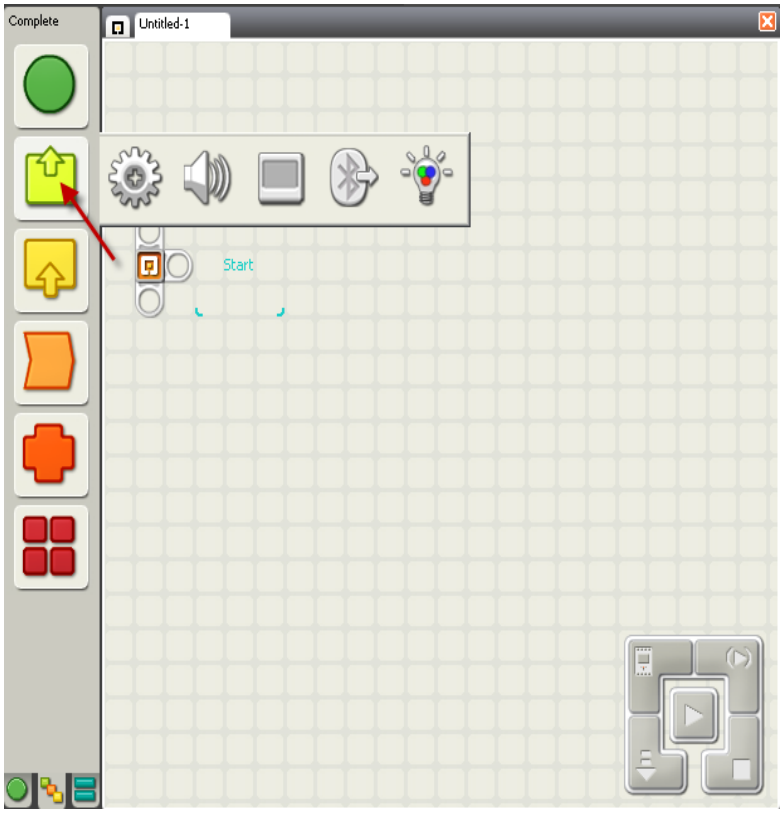
Blocuri acțiune



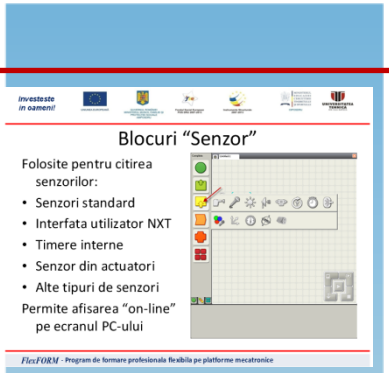
Sunt blocurile utilizate pentru a efectua diverse acțiuni. Acestea includ:

- Acționarea unui motor
- Afișarea unei informații pe ecran
- Generarea unui sunet
- Trimiterea unui mesaj BlueTooth
- Aprinderea unui LED al senzorului Color

Sunt grupate în cadrul "*Action*" (săgeata în figură)



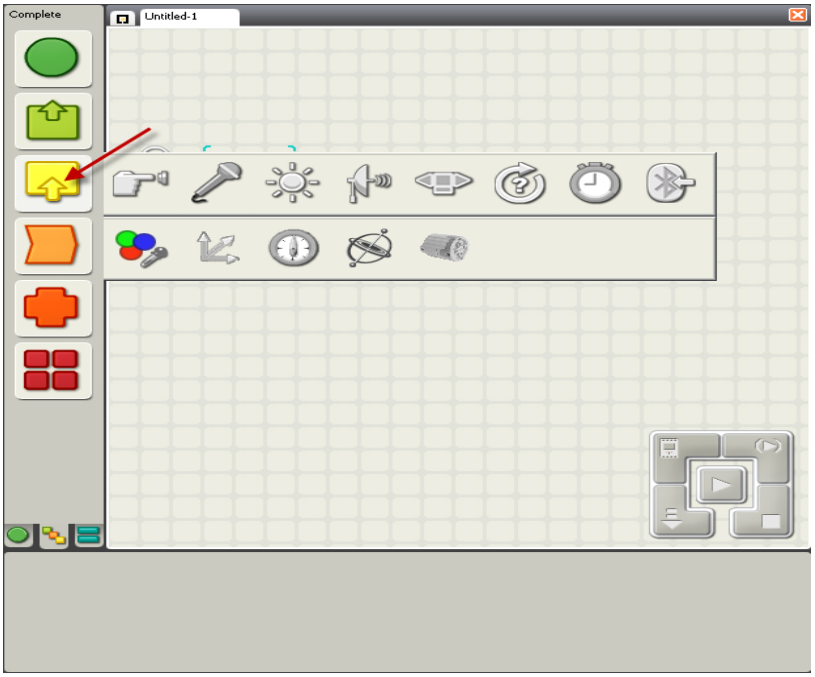
Blocuri Senzor



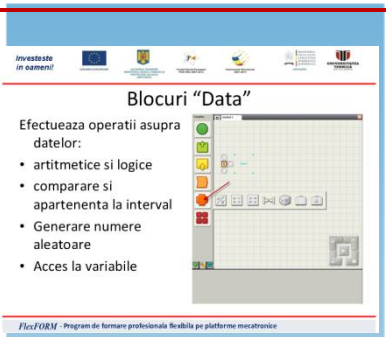
Se utilizează pentru a citi diverse informații. Acestea pot fi:

- informații provenite de la diversele tipuri de senzori (atingere, ultrasonic, culoare etc.)
- informații interne (butoanele interfeței utilizator, timere etc.)
- informații referitoare la comunicație (mesaje bluetooth)

Pot fi utilizate pentru a afla valoarea curentă a unui senzor direct pe ecranul PC-ului.



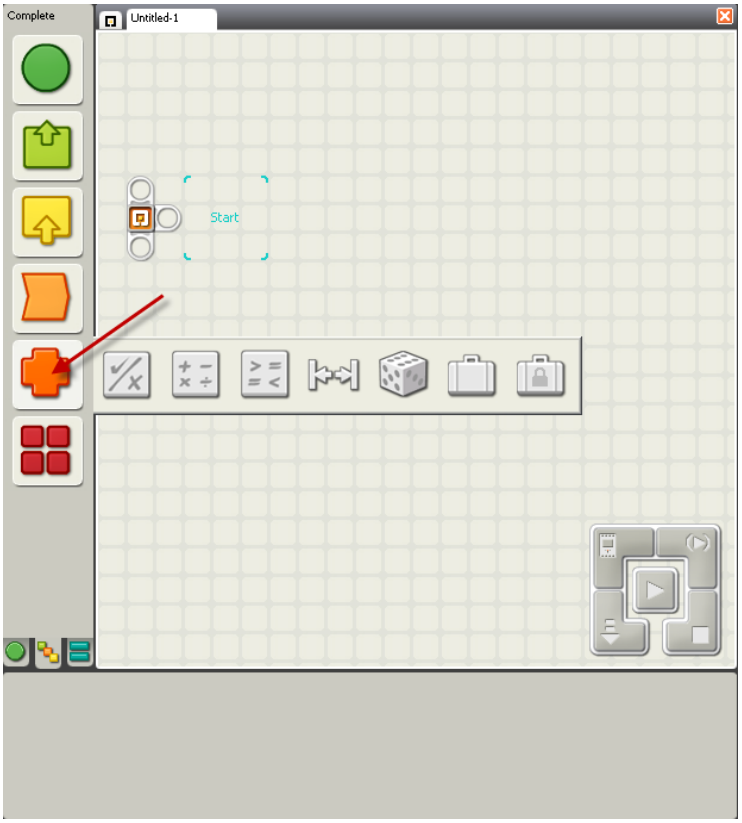
Blocuri DATA



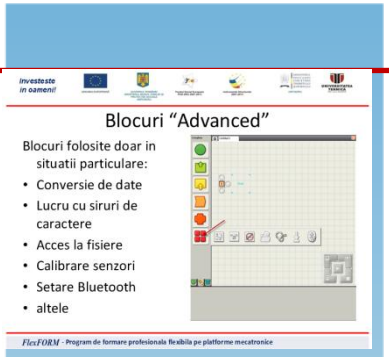
Operează asupra datelor. Se folosesc întotdeauna cu ajutorul conexiunilor de date.

Se împart în mai multe categorii:

- operații aritmetice și logice – se pot efectua operații aritmetice simple (adunări, scăderi, înmulțiri și împărțiri) asupra variabilelor numerice sau operații logice (și, sau, sau exclusiv, negare logică) asupra variabilelor logice
- verificări de depasire ale unor valori – se pot compara două valori (mai mic, mai mare, egal) sau se poate face testul de apartenență (inclus, neinclus) a unei valori într-un interval definit. Rezultatul acestor operațiuni este întotdeauna de tip logic
- generare de numere aleatoare – se generează un număr aleator într-un interval definit.
- manipulare de constante și variabile (mai multe detalii în secțiunea Variabile)



Blocuri Advanced

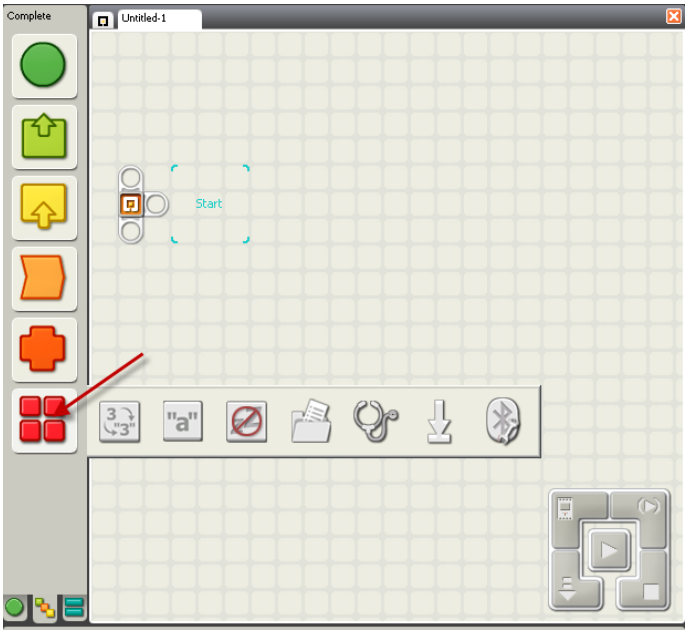


Sunt diverse tipuri de blocuri care nu se încadrează în categoriile de mai sus, și care probabil nu vor fi folosite decât în anumite situații particulare.

Putem aminti blocurile pentru operații asupra variabilelor (și datelor în general) de tip text. Acestea sunt folosite pentru a converti o valoare numerică într-una de tip text (Number to Text) și pentru a alipi două sau mai multe texte (Text). Acesta din urmă este echivalent cu instrucțiunea "Concat" a limbajelor de programare uzuale.

Tot din acest grup mai face parte blocul care nu permite controller-ului NXT să intre în Stand-By (blocul Keep Alive), blocul de calibrare a senzorilor, de resetare a senzorilor rotativi ai motoarelor sau blocul de setare a conexiunii bluetooth.

Mai trebuie să menționam blocul "File Access" care permite accesul la sistemul de fișiere a controlerului. Cu ajutorul acestuia se pot crea, șterge, citi sau edita fișiere. Este predominant utilizat pentru operații de tip data-log, în care se citesc valori de la diverși senzori pe intervale diferite de timp și se salvează în memoria nevolatilă a controlerului sub formă de fișiere text. Aceste fișiere sunt apoi descărcate pe calculator pentru a fi procesate, filtrate sau afișate sub formă de grafice, tabele etc.



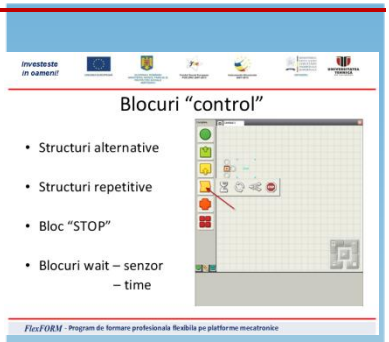
Blocuri control(flow)

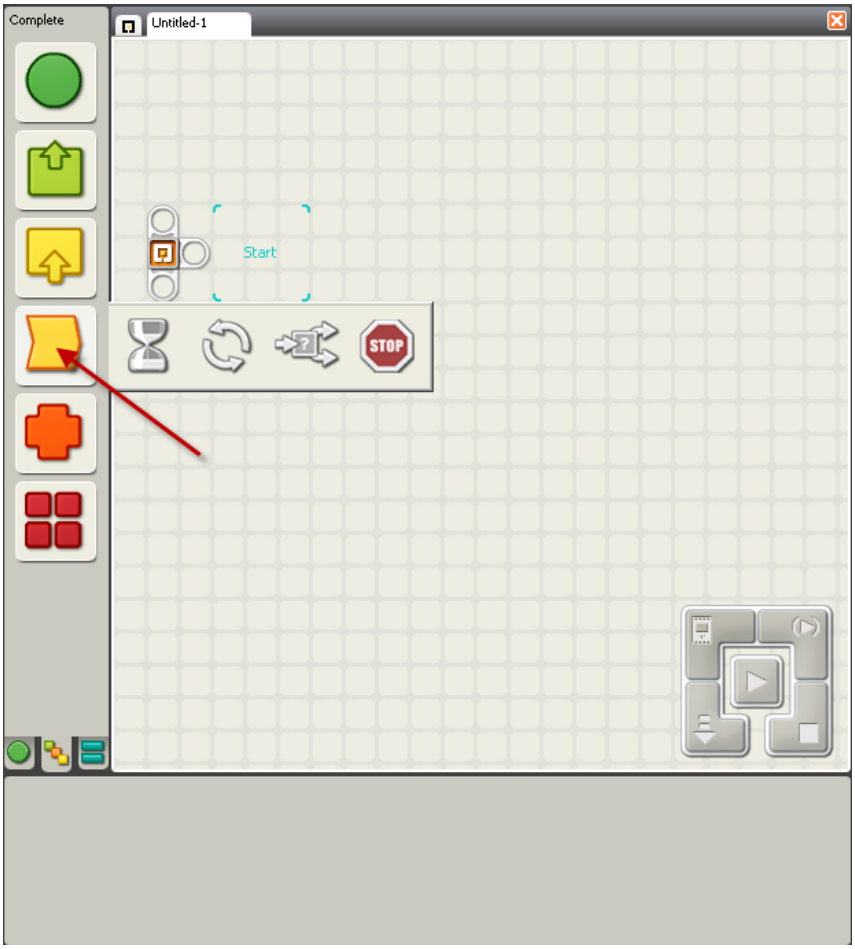
Sunt blocuri cu ajutorul cărora se poate controla execuția programului. Sunt similare cu structurile de control din limbajele de programare clasice.

Cu ajutorul acestora, structura programului scris nu mai este una liniară (de la primul până la ultimul bloc), ci se pot crea bifurcații (Switch) ale execuției, dintre care doar una dintre ramuri se vor executa, sau repetiții (loop) ale unor secvențe de cod.

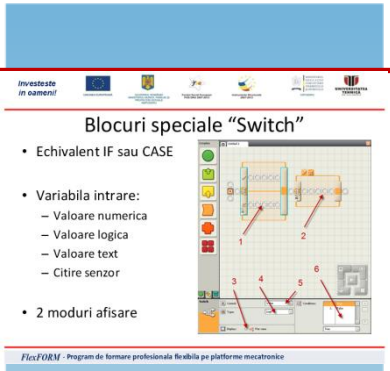
O particularizare a structurii repetitive este blocul Wait, care determină programul să aștepte până când valoarea citită de la un senzor (sau timer) se incadrează intr-un interval prestabilit.

Pe lângă cele două tipuri de structuri clasice, mai există posibilitatea utilizării blocului de tip Stop, care va determina terminarea necondiționată a programului și oprirea actuatorilor structurii controlate.





Blocuri speciale – SWITCH



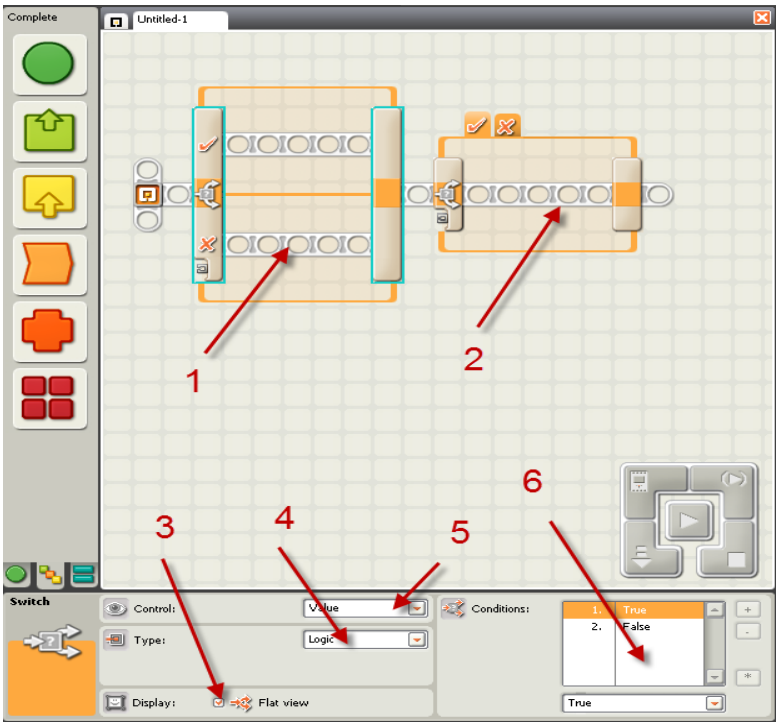
Sunt blocurile echivalente ale structurilor IF și CASE. Se găsesc în cadrul blocurilor de tip FLOW.

În forma cea mai simplă, blocul Switch arată ca în figură, indcația 1. În funcție de valoarea de adevăr a firului de intrare, se vor executa fie blocurile de sus (marcate prin semnul ✓), fie blocurile de jos (marcate cu X).

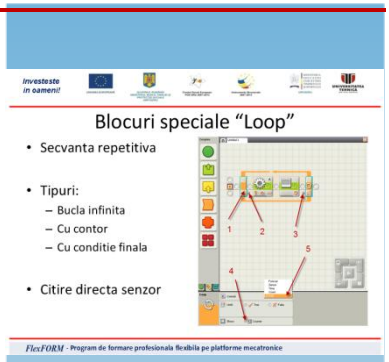
Dacă se dorește o afișare compactă a acestei structuri, se poate debifa opțiunea "Flat view" (indcația 3), caz în care aspectul blocului va fi ca cel din figură, indcația 2. În acest moment, se afișează ramura selectată (✓ sau X).

Decizia de execuție a uneia cau celeilalte ramuri se face în funcție de selecția făcută la "control" (indcația 5), unde se poate selecta "value" sau "senzor". Dacă se selectează "value", în cadrul "type" (indcația 4) se poate selecta tipul variabilei de selecție: logic, number sau text. La selectarea unei variabile de intrare de tip Number, structura switch permite selecții multiple, care se pot seta cu ajutorul tabelului de adevăr "conditions" (indcația 6).

Selectarea la parametrul "control" (indcația 5) atrage după sine execuția unei ramuri sau a celeilalte, în funcție de tipul senzorului selectat la "type". În acest caz, proprietățile structurii "switch" vor fi asemănătoare cu proprietățile senzorului selectat. Este permisă alegera portului de conexiune și a acțiunii care va avea ca efect execuția ramurii "✓".



Blocuri speciale – LOOP



Blocurile loop (buclă inchisă) se folosesc pentru a repeta o secvență de cod, și arată ca în indicația 1.

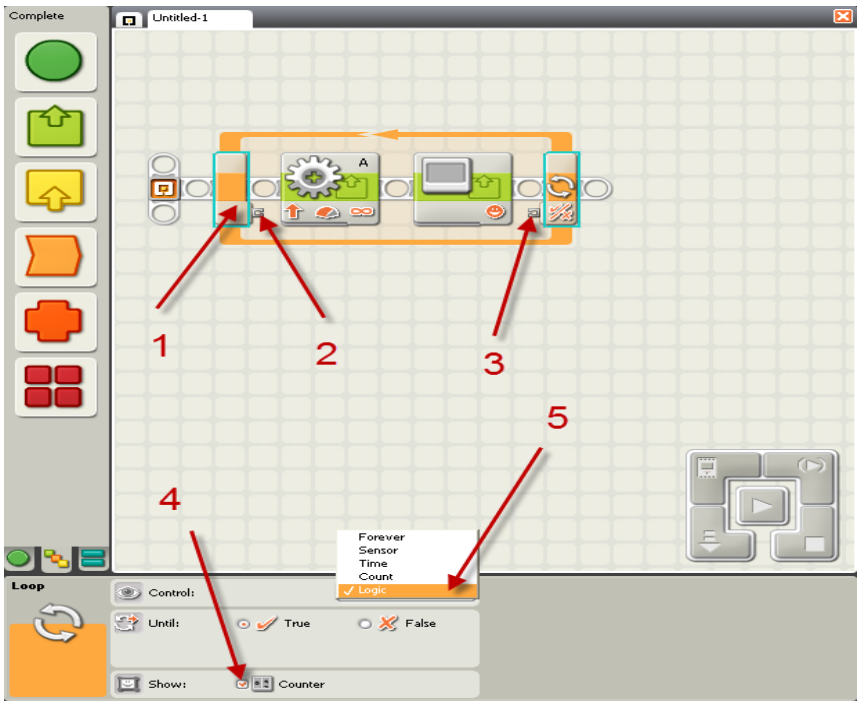
Pentru a opri rularea secvenței de cod, este nevoie de o condiție de ieșire din buclă: un semnal de la sensor, un semnal logic sau un număr de repetări a buclei. De asemenea, ciclul poate rula și la infinit, dacă este necesar. Setările pentru aceste condiții se pot observa in indicația 5.

Când blocul loop este setat pe buclă infinită, simbolul infinit (∞) este afipat în colțul din dreapta jos a blocului.

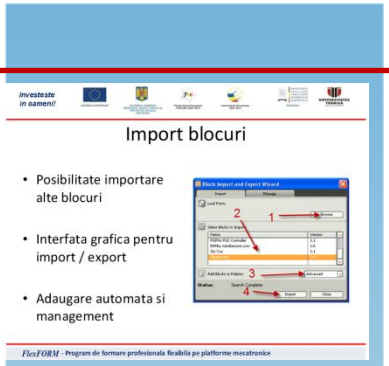
În cazul în care este selectată căsuța "Counter", va apărea o intrare care permite utilizarea unui număr finit și cunoscut de bucle (indicația 3).

O altă opțiune pentru folosirea blocului loop este citirea directă de la senzori. Când această opțiune este selectată, senzorul ales va apărea pe bloc. Ieșirea din buclă se va face în funcție de valoarea citită de la senzor și condiția dată.

După cum se poate observa în indicația 4, se poate selecta opțiunea counter. Caâd această opțiune este selectată, la începutul blocului apare o intrare (indicația 2), care permite legarea conectorilor cu alte blocuri folosite în program.



Import blocuri



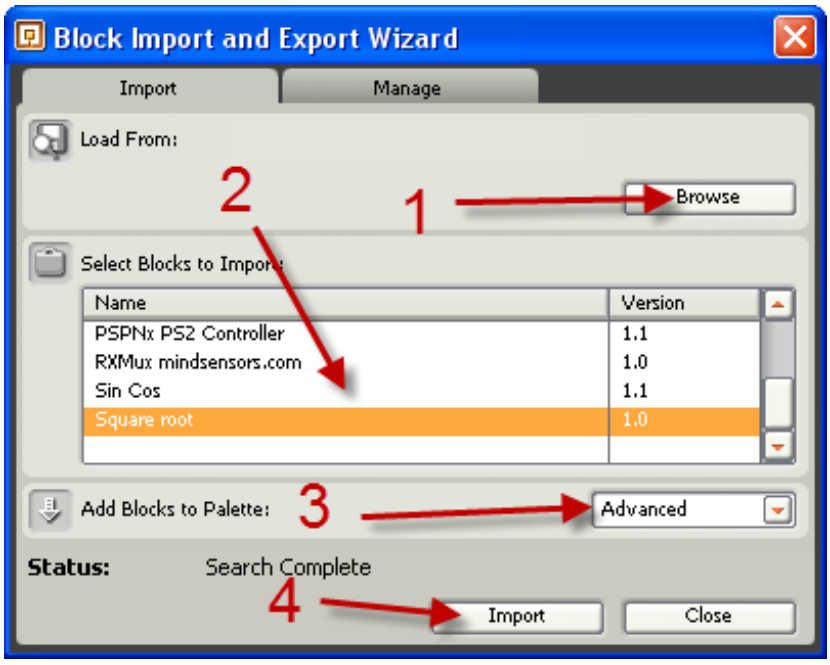
Deoarece nu toate blocurile necesare programării se găsesc in program, există opțiunea de importare a blocurilor.

Din meniul programului, din blocul “tools” se alege opțiunea de “block import and export wizard”. Se deschide fereastra de importare a blocurilor, după cum se vede în figura alaturată.

Se selectează butonul browse (indicația 1) pentru a căuta blocurile ce se doresc a fi importate. În indicația 2 se poate observa fereastra în care sunt adăugate blocurile găsite. Se selectează un bloc.

Se selectează paleta în care va fi adăugat blocul (indicația 3), după care se apasă butonul import (indicația 4).

Blocul va fi automat adăugat în paleta selectată.

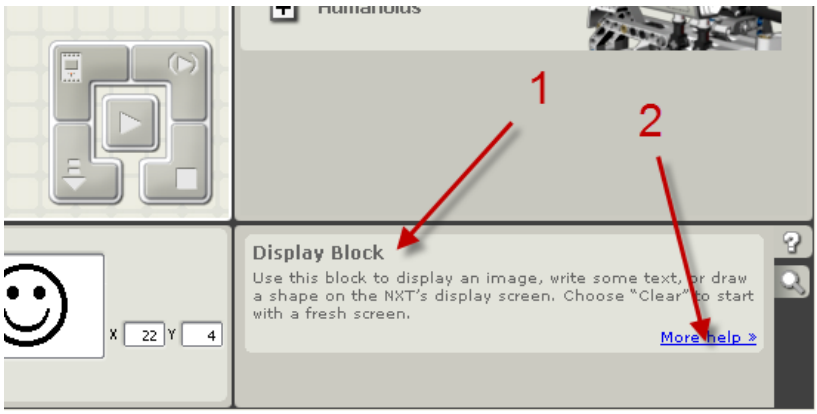


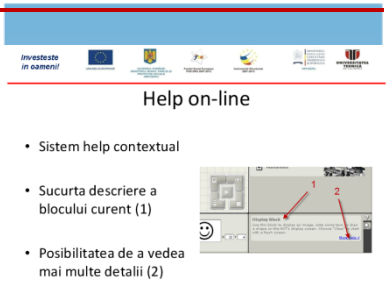
Help on-line

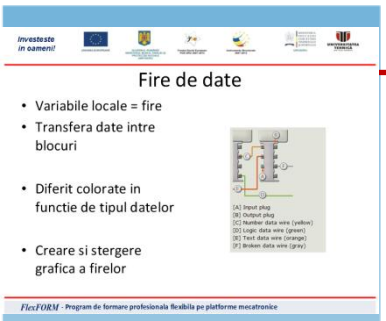
Pentru mai multe detalii despre program și blocurile aferente, în colțul dreapta jos al programului apare fereastra de ajutor.

În această fereastră există două opțiuni:

- o scurtă descriere a blocului selectat (indcația 1)
- posibilitatea de a vedea mai multe detalii (indcația 2), aceasta facând trimitere la suportul on-line







Fire de date

Firele de date transferă informații între blocurile de programare. Anumite blocuri necesită fire de date pentru ca acestea sa funcționeze.

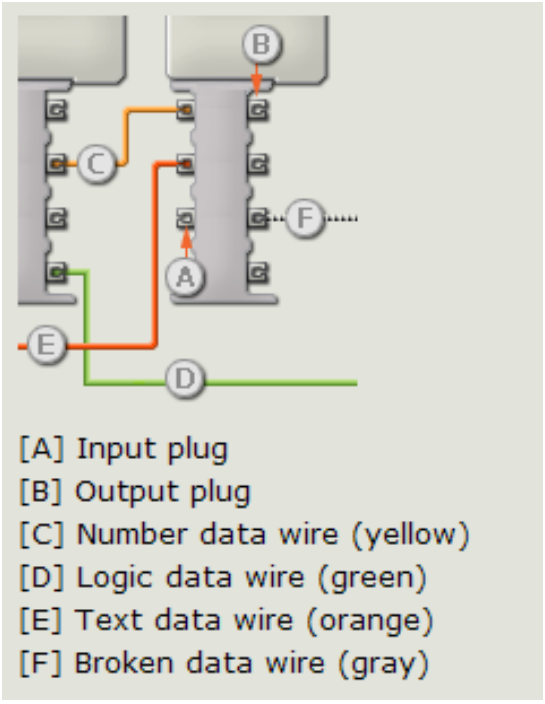
În funcție de tipul datelor, firele de date au diferite culori:

- galben – pentru date numerice
- verde – pentru date logice (true/false)
- portocaliu – pentru date text

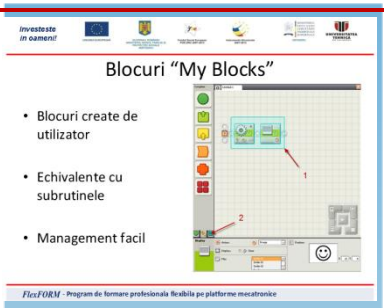
Dacă firul de date are culoarea gri, înseamnă că firul nu este complet sau este legat la o intrare greșită.

Pentru a crea un fir de date, se selectează ieșirea blocului a cărui informație vrem să o folosim și se trage firul de date până la intrarea blocului de execuție dorit.

Pentru a șterge un fir pus greșit sau pe care nu mai vrem să-l folosim, se apasă pe locul de intrare a firului în blocul de execuție.



Blocuri “My Blocks”

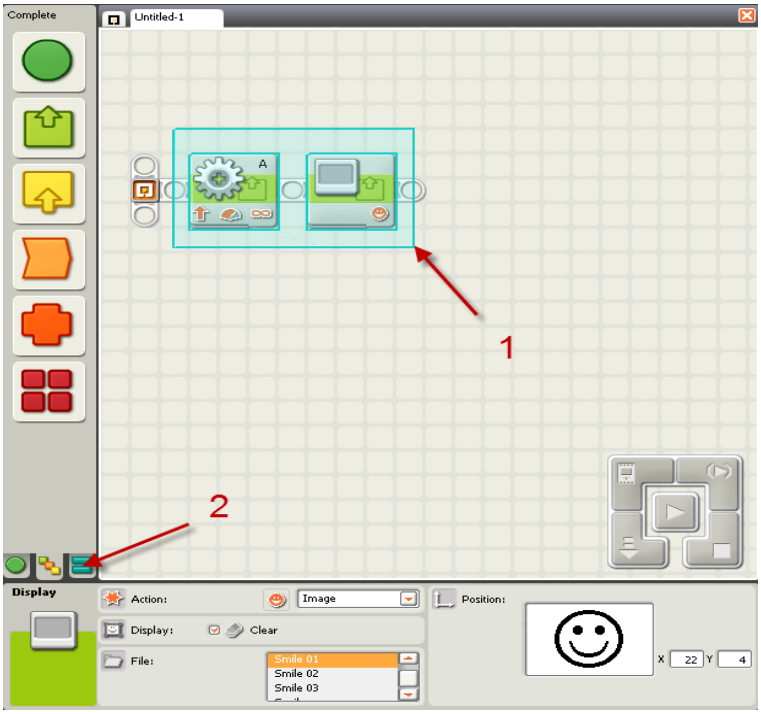


Cu ajutorul blocurilor “My blocks” se pot personaliza grupe de blocuri. De exemplu se creează un “My block” format dintr-un bloc de mișcare și un bloc de sunet. Acest bloc se poate utiliza mai apoi în diferite aplicații.

Crearea unui “My block”:

1. se selectează un număr de blocuri care se potrivesc din punct de vedere logic (indicația 1)
2. se selectează toate blocurile de mai sus
3. din meniu se alege comanda “choose a new my block”; se deschide o fereastră pentru crearea unui nou “my block”
4. se denumește acest nou bloc și i se scrie o scurtă descriere; se apasă next
5. se alege icoana pentru noul bloc; se apasă finish

Noul “my block” va apărea în paleta denumită “Custom palette” (indicația 2).



Interfața cu utilizatorul



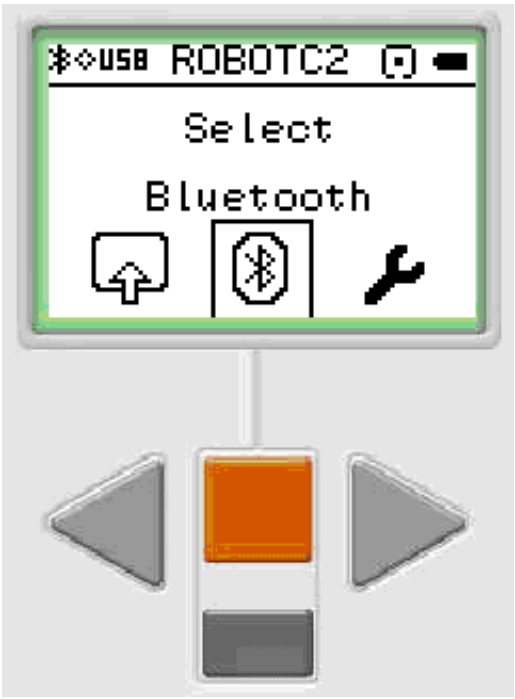
Controlerul NXT oferă mai multe modalităti de comunicare cu utilizatorul. Acestea sunt:

- Afişajul
- Butoanele
- Difuzorul

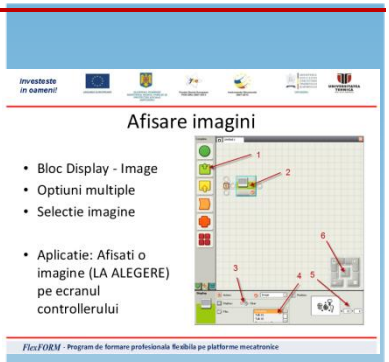
Aceste modalităţi ar putea fi încadrate în două categorii: intrări si ieşiri.

Principala modalitate de introducere a comenzilor se realizează folosind butoanele de pe controller. Acestea se folosesc atât la navigarea în meniul NXT cât şi în timpul rulării unui soft dacă programul realizat acceseaza această funcție.

Principalele modalităţi de ieşire a informaţiilor sunt afişajul şi difuzorul. Afişajul poate fi folosit pentru readarea informaţiilor în formă vizuală (poze, text, grafice) iar difuzorul în formă audio.



Afișare imagini



Pentru afișarea de imagini este nevoie de un bloc Display (indicația 2). Acest bloc are mai multe opțiuni, cum ar fi:

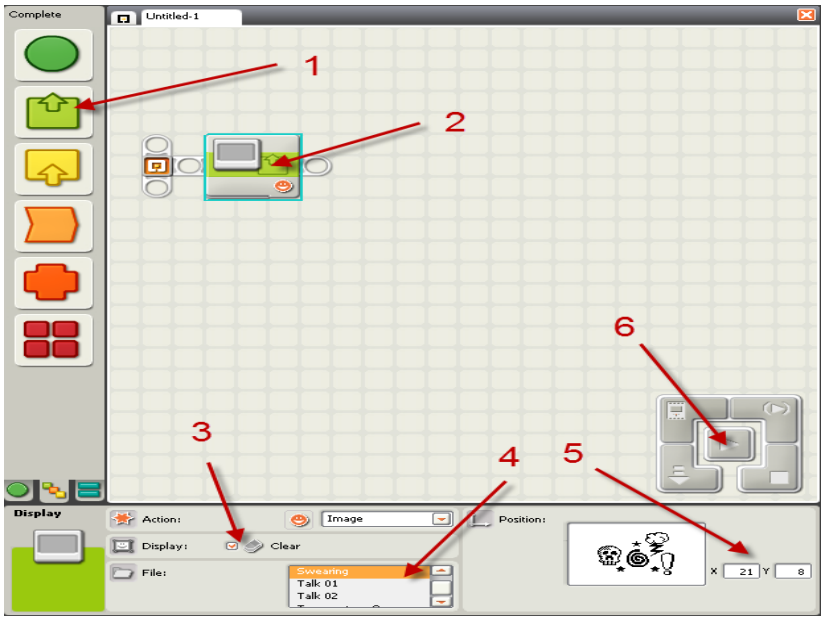
- image – afișează imagini
- text
- drawing
- reset

Opțiunile acestui bloc se pot schimba din panoul de comandă.

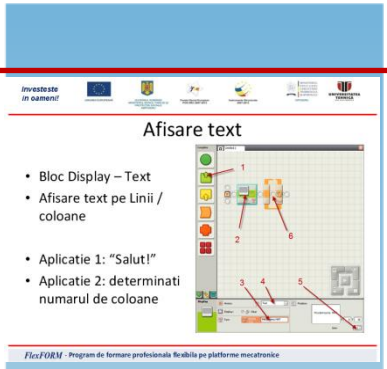
Aplicație:

Să se afișeze pe display o imagine la alegere

1. se alege din meniul din stânga blocul „display” (indicația 1)
2. se pune blocul în program (indicația 2)
3. pentru a șterge imaginea precedentă de pe ecaran se selectează căsuța „clear” (indicația 3)
4. din bara de comandă se alege imaginea dorită (indicația 4)
5. se alege poziția imaginii (indicația 5)
6. programul se descarcă în controller



Afișare text



O altă aplicație a blocului "display" este scriere text.

După ce este selectată opțiunea pentru text, se poate scrie textul dorit în căsuța desemnată (indicația 3).

Textul se poate afișa oriunde pe ecran, în funcție de coordonatele selectate.

Aplicație 1:

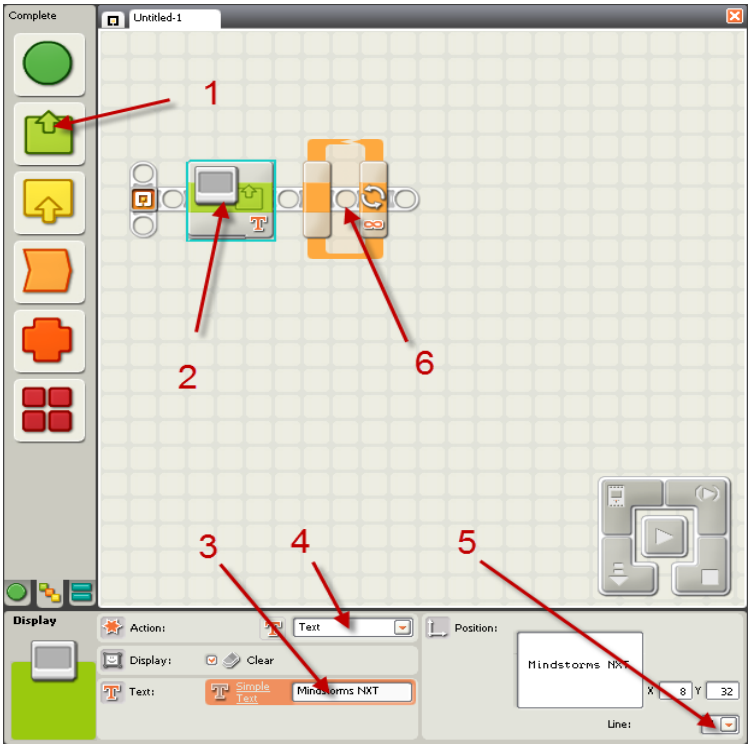
Să se afișeze pe ecran mesajul "Salut".

1. Se selectează blocul dorit din paleta "Action"
2. Se selectează "Text" din panoul de comandă (indicația 4).
3. În căsuța desemnată se scrie mesajul "Salut".

Aplicație 2:

Să se determine numărul de coloane care pot fi afișate.

Numărul de linii se poate observa în meniul din indicația 5 (maxim 8).

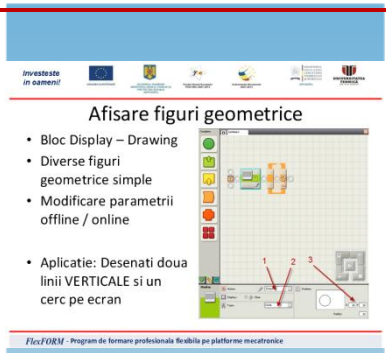


Afișare figuri geometrice

Pe lângă opțiunile de text și imagine, blocul display mai are și opțiunea de afișare figuri.

Această opțiune permite alegerea unei figuri din meniu, cum ar fi un punct, o linie sau un cerc.

Din coordonatele x și y (indicația 3) se poate alege poziția figurilor pe ecran.

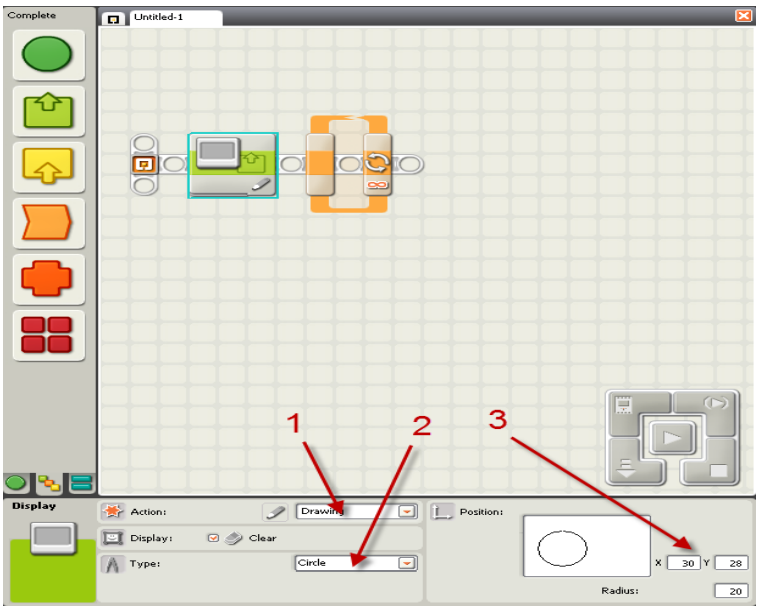


Aplicație:

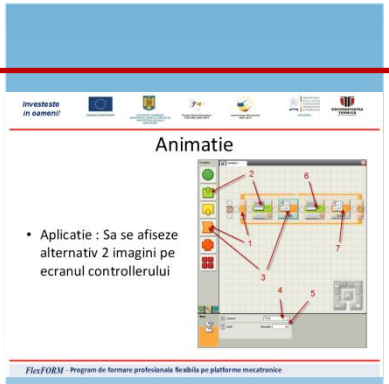
Să se deseneze două linii verticale și un cerc pe ecranul controllerului.

1. După ce s-a selectat blocul “Display”, se selectează funcția “Drawing” din panoul de comandă (indicația 1).
2. Se alege din panoul de comandă figura geometrică dorită, în acest caz, o linie. (indicația 2)
3. Pentru ca linia să fie verticală, se modifică valorile pentru coordonatele X, astfel încât să fie egale. Lungimea liniei este dată de coordonatele Y (indicația 3).
4. Pentru a doua linie verticală se efectuează aceleași operații ca și mai sus într-un alt bloc “Display”.
5. Pentru a desena un cerc, se selectează opțiunea pentru cerc într-un alt bloc Display. Se trec coordonatele dorite, după care se scrie raza dorită a cercului.

Pentru ca toate imaginile să rămână pe ecranul controlerului, se deselectează opțiunea “Clear”. Tot programul se pune într-o buclă.



Animație



Pentru a realiza o animație, este nevoie de două imagini afișate alternativ pe ecran, după cum se vede în figura de mai jos.

Pentru a afișa alternativ două imagini pe ecranul NXT-ului se parcurg pașii următori:

1 – se selectează blocul loop (buclă) și se pune în program

2 – se ia un bloc display, se pune în buclă și i se alege imaginea dorită.

3 – se selectează blocul wait și se pune în program

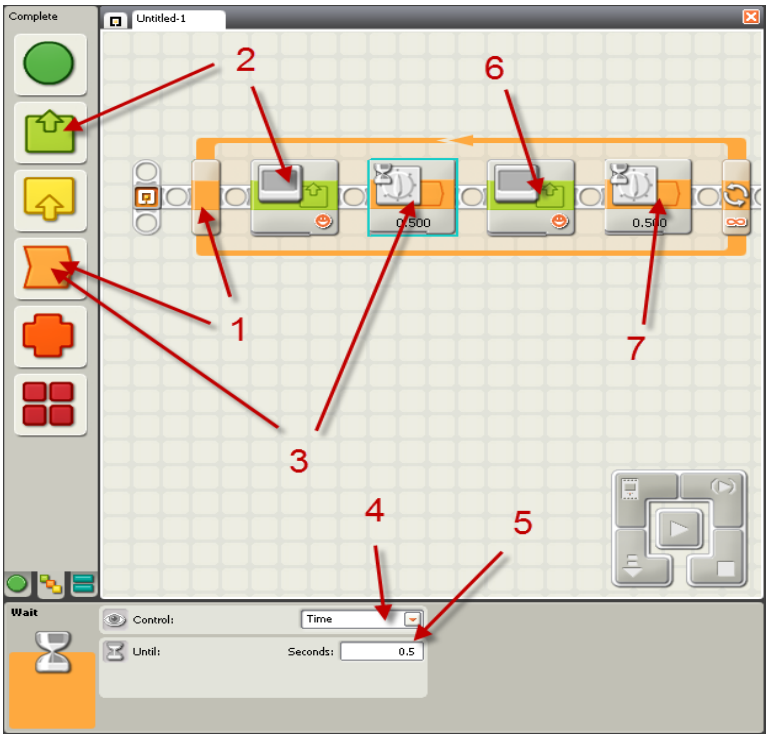
4 – din panoul de comandă se schimbă opțiunea blocului, selectându-se instrucțiunea „Time” pentru ca blocul anterior să aștepte un anumit interval de timp dorit.

5 – se selectează intervalul de timp la 0.5 secunde.

6 – pentru a doua imagine care trebuie afișată pe ecran se ia un alt bloc display în care se alege imaginea dorită.

7 – ultimul bloc din program este „wait” pentru ca imaginea să rămână un anumit timp pe ecran.

La ambele blocuri dispaly trebuie selectată optiunea clear” pentru ca imaginea precedentă să nu ramană afișată pe ecranul controllerului.



Text Vertical

Pentru a scrie un text pe verticală, e nevoie de câte un bloc pentru fiecare literă. Textul poate fi compus din maxim 8 litere. De asemenea, opțiunea “Clear” trebuie deselectată pentru a rămâne pe ecran litera din blocul anterior.



Aplicație

Să se scrie cuvântul “Salut” pe verticală.

1. Se selectează, în blocul Display, opțiunea pentru scriere text.
2. Se scrie prima literă a cuvântului, pe prima linie a blocului. Liniile se selectează din panoul de comandă. De pe coordonata X se alege locul pe orizontală a literei.
3. Se repetă aceleași operații pentru literele ramase din cuvânt, schimbând doar linia pe care trebuie să se afle următoarea literă. Coordonata X trebuie să rămână aceeași pentru fiecare literă.

Toate blocurile se pun într-o buclă, pentru a rămâne cuvântul pe ecranul controlerului. De asemenea, pentru fiecare bloc se deselectează opțiunea “Clear”.

Creare imagini



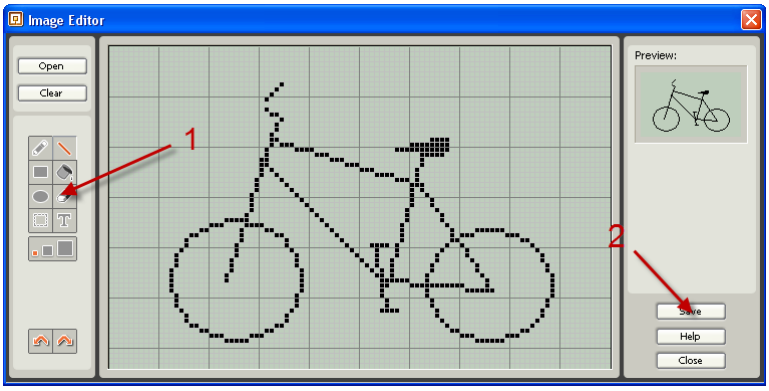
Pentru a crea imagini mai complexe, se poate folosi editorul de imagini încorporat în programul Mindstroms NXT, care poate fi accesat din meniul "tools".

Se va deschide o fereastră similară cu programul „Paint" din Windows, unde se pot desena imaginile dorite.

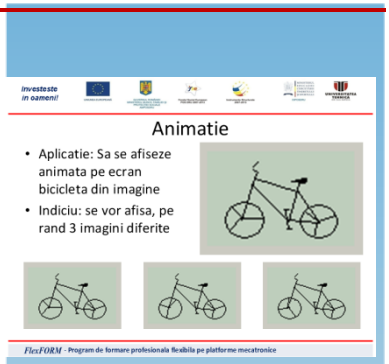
Aplicație:

Se va crea o imagine care să semene cu o bicicletă, imagine pentru care va fi realizat un program care să o afişeze pe ecranul NXT-ului.





Animație



Folosind programul ”image editor” se vor crea mai multe imagini în aşa fel încât la afişarea lor în mod consecutiv să dea impresia că roţile unei biciclete se rotesc.

Programul pentru această animaţie nu este dificil, acesta presupunând afişarea fiecărei poze pentru un anumit interval de timp şi repetarea acestei rutine.



Histograma



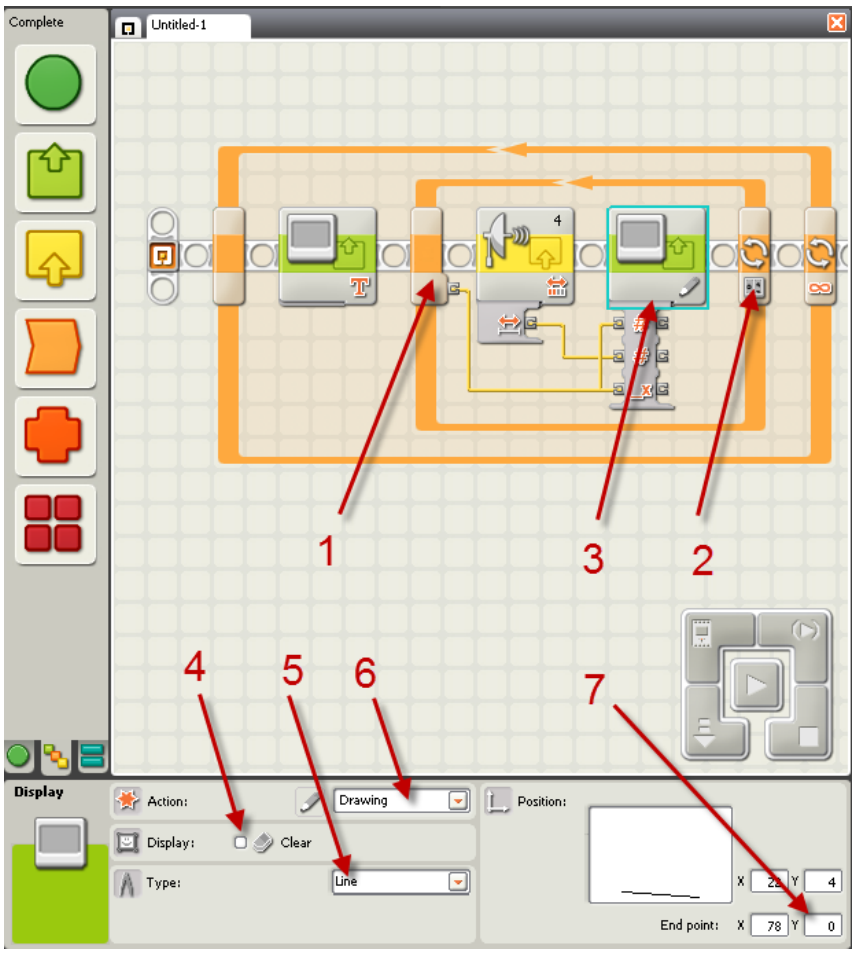
O histogramă este o reprezentare grafică a distribuţiei unor date.

În general această reprezentare se face din coloane verticale alăturate, de diferite înălţimi, înălţimea acestora reprezentând diferitele mărimi.

Histograma realizată în programul Mindstorms va afişa datele citite de pe senzorul ultrasonic în timp real.

Acest lucru se va face adăugând pentru fiecare citire a senzorului o linie verticală pe ecranul NXT-ului, cu lungimea proporțională cu valoarea citită de la senzor.

Liniile se vor adăuga consecutiv din stânga spre dreapta ecranului, iar dacă desenul a ajuns la capătul ecranului se șterge imaginea (clear din blocul display) și se reia algoritmul de mai sus.



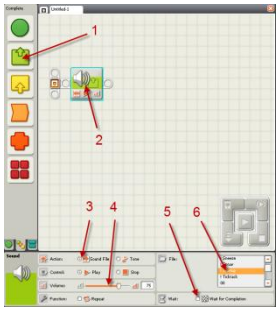
Sunet

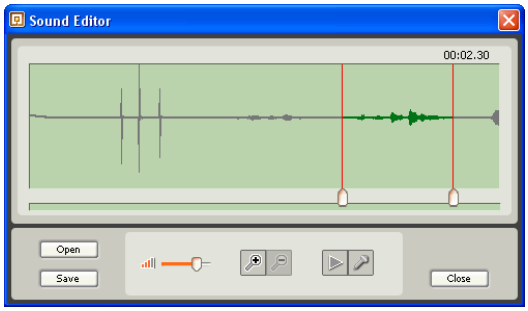


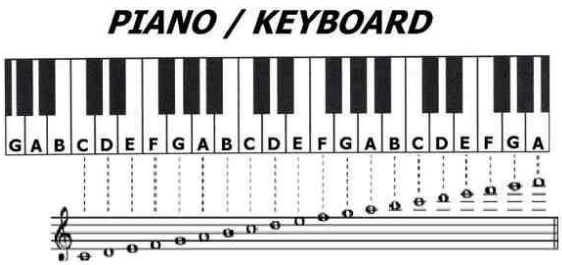
Semnalele sonore sunt un mijloc important de redare a informaţiilor şi de atenţionare.

Programul Mindstorms NXT dă posibilitatea utilizatorului de a reda atât înregistrări cât şi de a produce anumite sunete.

La fel ca la histogramă, semnalele sonore pot fi folosite pentru a cuantifica valorile citite de la senzorii Mindstorms prin modificarea tonalității sunetelor redate.

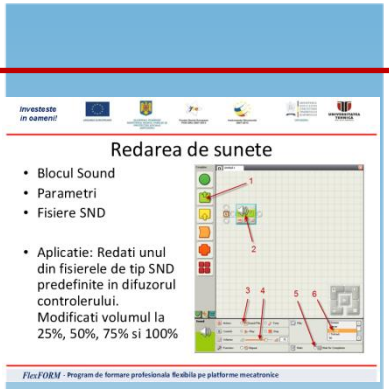








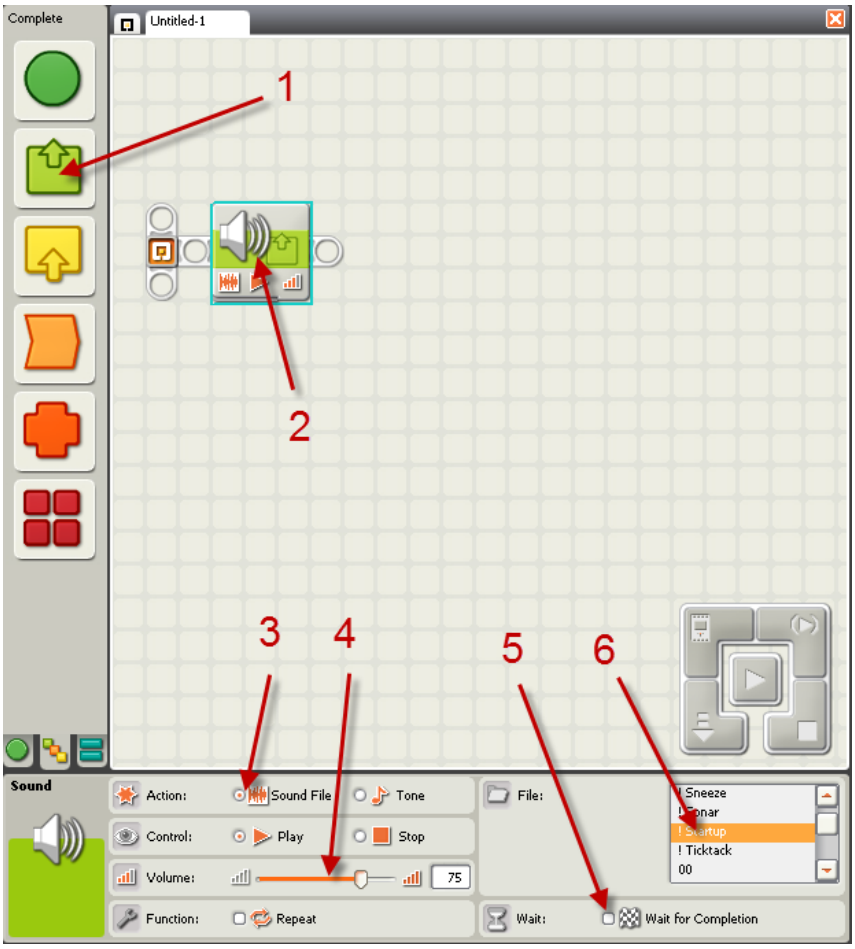
Redarea de sunete



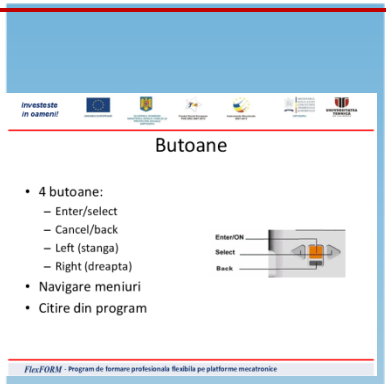
Redarea sunetelor se face cu ajutorul blocului ”Sound” (indicația 1 și 2).

Acest bloc se setează în felul următor pentru a reda sunete:

- opţiunea ”Sound File” la ”Action” (indicația 3)
- ”Play” la ”Control”
- ”Volume” în funcție de nivelul necesar, în mod normal fiind lăsat la 75% (indicația 4)
- la ”File” se va alege fişierul dorit (indicația 6)
- pentru a nu suprapune 2 sunete, se bifează opțiunea „wait for completion” (indicația 5)



Butoane



Controlerul NXT dispune de patru butoane care pot fi folosite atât pentru navigarea în meniul său cât şi folosirea lor în programe propri.

Cele patru butoane au următoarele funcţii atunci când sunt folosite în meniu:

- Enter/select
- Cancel/back
- Left (stânga)
- Right (dreapta)

Atunci când sunt folosite în programe, fiecărui buton îi poate fi atribuită o funcţie individuală, în funcție de dorințele utilizatorului.



Butoane – Wait for



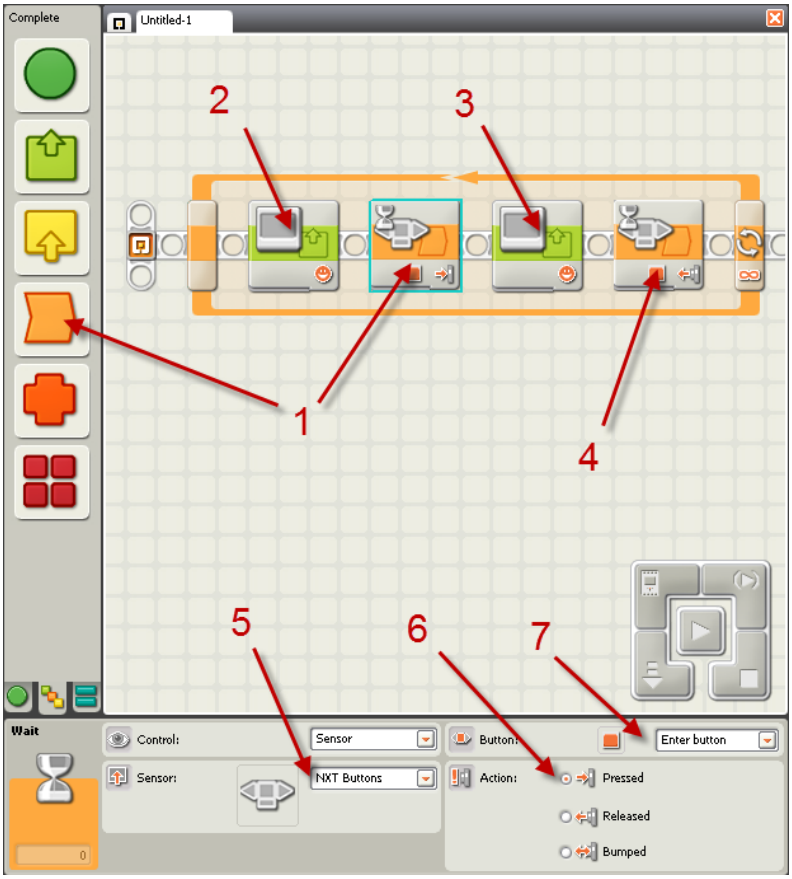
Blocul ”Wait for” permite introducerea unei proceduri de aşteptare a unei comenzi. Acest bloc poate fi setat să aştepte pentru diferite condiţii cum ar fi:

- Timp
- Butone NXT
- Comunicare
- Senzori
- Timer

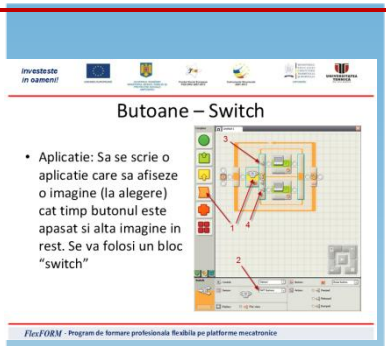
Pentru a da comanda de la butoanele NXT-ului, blocul ”Wait” trebuie setat să fie de tip ”Sensor” şi la Sensor trebuie aleasă opţiunea ”NXT Buttons”, după care se poate alege butonul care va fi folosit.

Aplicație:

Să se afișeze o imagine (la alegere) cât timp butonul este apăsat și altă imagine în rest. Se vor folosi blocuri de tip “wait for”, după exemplul de mai jos:

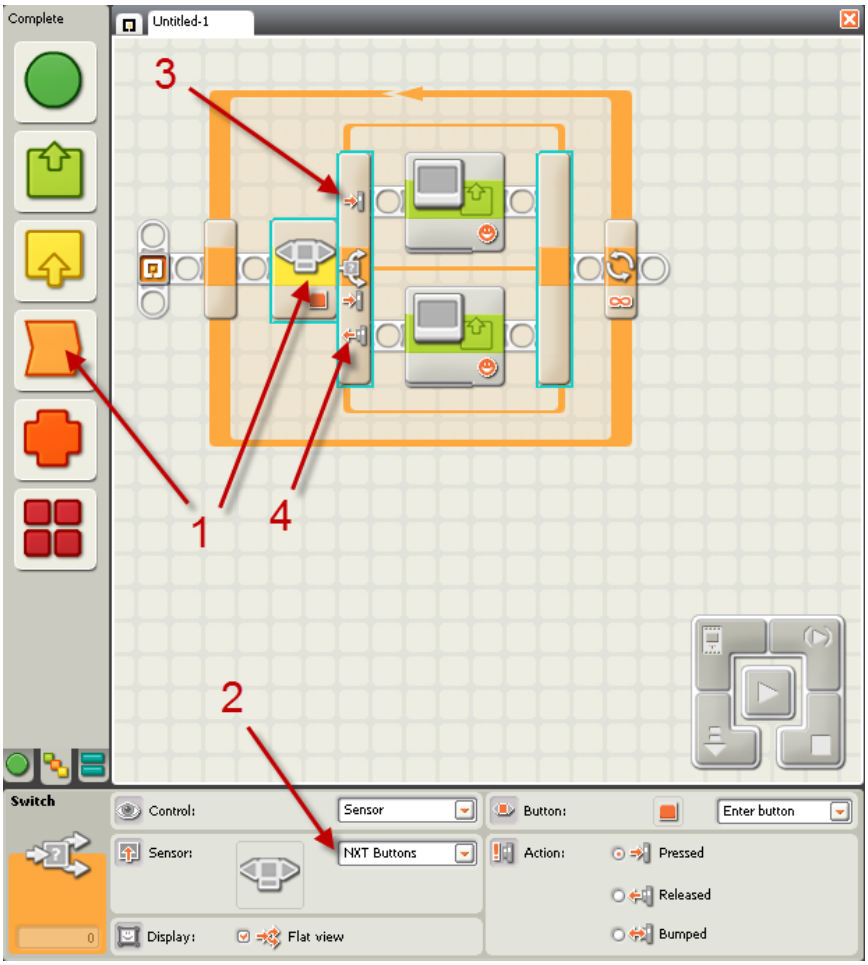


Butoane – Switch

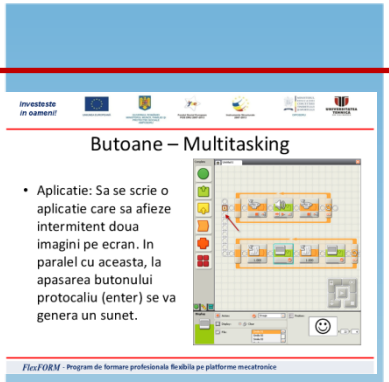


Aplicație:

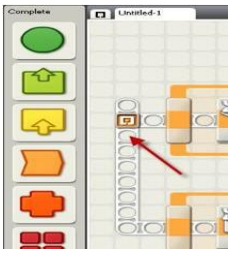
Să se creeze o aplicație care să afișeze o imagine (la alegere) cât timp unul dintre butoane este apăsat și altă imagine în rest. Se va folosi un bloc de tip switch, după cum este exemplificat mai jos:



Butoane – multitasking



Controllerul Mindstorms NXT permite executarea a 3 secvențe de cod simultan. Acest lucru face posibilă rezolvarea unor aplicații mai complexe.

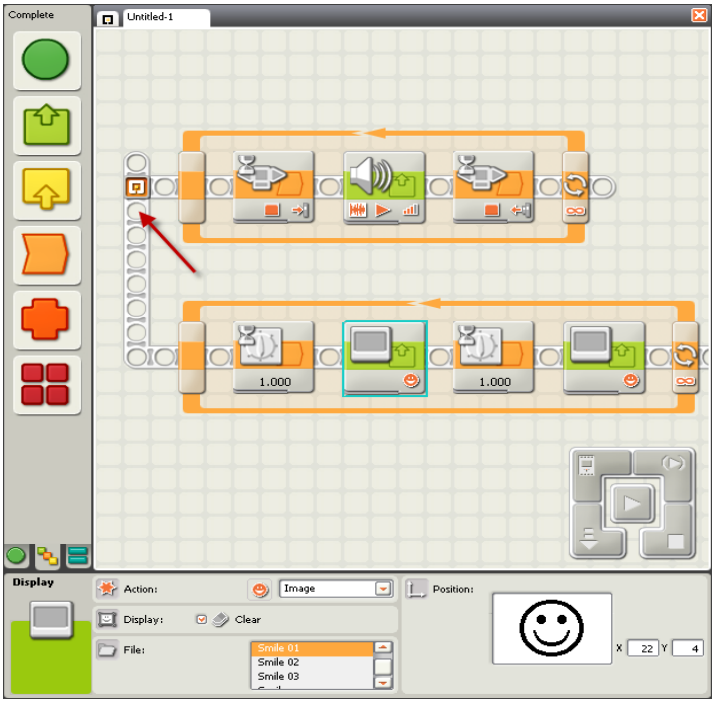


Aplicație:

Să se scrie o aplicație care afișează intermitent două imagini pe ecran. În paralel cu acestea, la apăsarea butonului enter (portocaliu) se va genera un sunet.

În prima fază se creează blocul de afișare de imagini. Se ia un bloc de tip loop, în care se introduc 2 blocuri de afișare și 2 blocuri wait.

În a doua fază, conform figurii de mai sus, se trage o ramură. În noua ramură creată se așează un bloc de tip loop în care se introduc 2 butoane de tip „wait for” și între ele un bloc de tip „sound”. Condițiile blocurilor „wait for” vor fi „on button pressed” și „on button release” a butonului „Enter”.



Actuatori

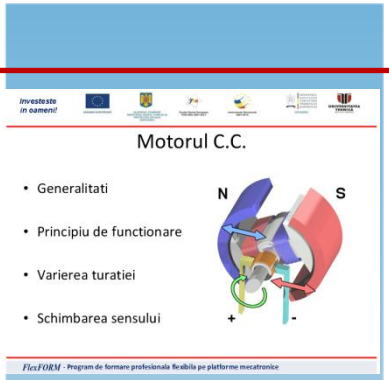


Controllerele Mindstorms NXT dispun de 3 intrări pentru motoarele oferite de LEGO. Conexiunea acestora se face prin cabluri de date. Aceste cabluri de date nu doar alimentează actuatorii ci fac și conexiunea dintre senzorul de rotație din componenta acuatorului și interfața integrată în controller pentru acesta.

LEGO Mindstorms oferă suport și pentru versiunile mai vechi de echipamente LEGO, punând la dispoziție și adaptoare pentru actuatorii mai vechi. De asemenea, controllerele suportă orice motoare de curent continuu ce pot fi alimentate cu 9V, 1 A.



Motorul de curent continuu (Motorul C.C.)



Motorul de curent continuu este un dispozitiv electromecanic ce transformă energia electrică în energie mecanică. Este format din două părți: rotor și stator.

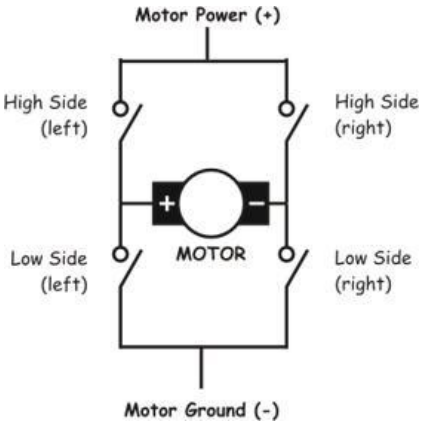
Acesta funcționează datorită forțelor electromagnetice ce acționează asupra unui conductor alimentat la curent aflat în câmp magnetic. Statorul este format din unul sau mai mulți poli magnetici, iar rotorul dintr-o bobină.

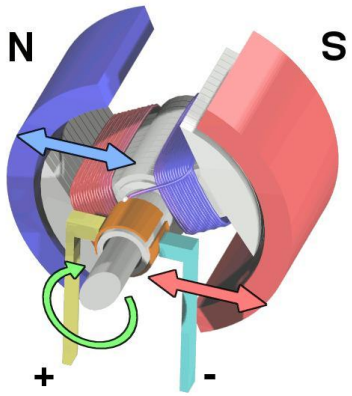
Rotorul de deplasează în câmpul magnetic până când polii rotorici se alinează cu polii magnetici opuși din stator. În acel moment, polaritatea rotorului se inversează și acesta va continua deplasarea până la următoarea aliniere a polilor magnetici.

Pentru a putea modifica turația motorului, trebuie modificată tensiunea aplicată infășurării rotorice. Acest lucru se poate realizeaza prin mai multe metode: introducerea în circuit a unei rezistențe variabile, a unor generatoare sau prin modularea în lățime a impulsului (PWM).

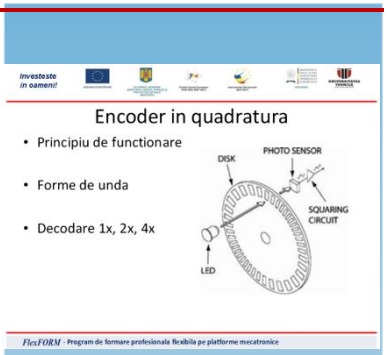
Schimbarea sensului de rotație se poate realiza fie prin schimbarea polarității tensiunii de alimentare, fie prin schimbarea sensului câmpului magnetic de excitație. Pentru a modifica polaritatea tensiunii se folosesc punțile H.

Puntea H este un circuit electronic format din 4 „întrerupătoare” (de obicei tranzistoare) dispuse ca în figura de mai jos, care modifică sensul de trecere a curentului prin motor.



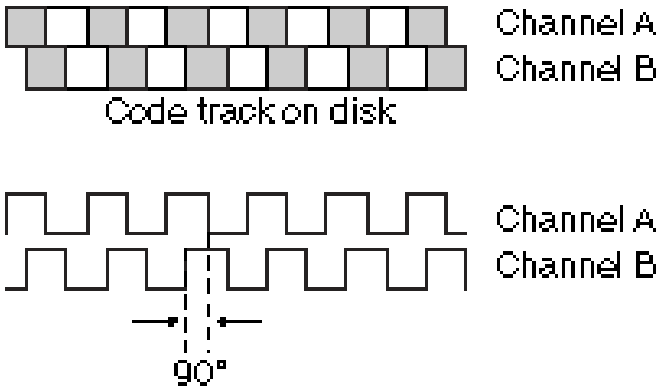


Encoder în cuadratură

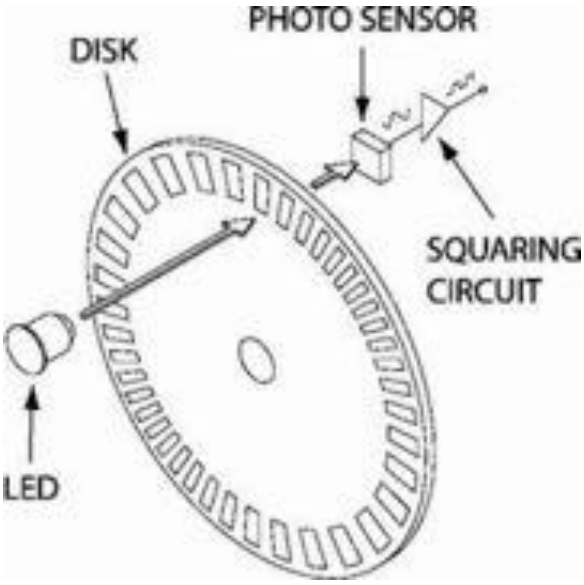


Encoderul în cuadratură este un senzor ce masoară poziția și viteza unui actuator rotativ. Poate fi de două tipuri, optic și mecanic. Cel optic folosește un disc cu fante, un LED și un senzor de lumină, cel mecanic folosește două contacte acționate de camele de pe rotor.

Amandouă folosesc 2 canale de ieșire (A și B) pentru a detecta poziția, forma de undă fiind defazată la 90 de grade.



Monitorizând pulsurile furnizate de cele două ieșiri se determină viteza, respectiv sensul de rotație. Dacă trecerea se face de la A la B, atunci sensul de rotație este în sens orar, iar dacă trecerea este de la B la A atunci sensul este trigonometric.



Servomotorul LEGO



Servomotorul LEGO este un motor de curent continuu de 3 W, compus din următoarele:

- Corpul motorului propriu-zis
- Angrenaj cu roți dințate (reductor)
- Senzor de rotație (Encoder în cuadratură)
- Corp de prindere ale altor piese LEGO (de exemplu axuri)

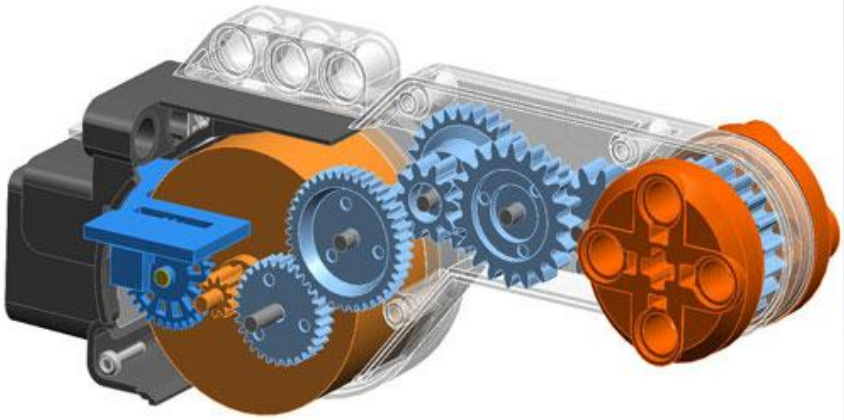
Precizia senzorului de rotație este de 1 grad. Acesta poate măsura rotațiile motorului în grade sau în rotații complete (1 rotație completa = 360 grade).

Avantaje ale folosirii motoarelor LEGO:

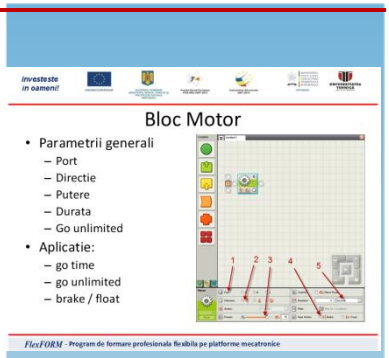
- Conector de ieșire compatibil cu mai multe aplicații LEGO
- Prezența senzorului de rotație
- Protecție la supracurent și supratensiune
- Prezența reductorului

Dezavantaje ale folosirii motoarelor LEGO:

- Design-ul nu e compact
- Gama mică de puteri disponibilă
- Turație scăzută la ieșire
- Roțile dințate sunt din plastic, ceea ce duce la frecări mari, randament scăzut și posibilitate de uzură la sarcini mai mari



Blocul „Motor”



Blocul motor reprezintă interfața cu motoarele a utilizatorului.

Parametrii generali ce pot fi modificați din acest bloc sunt:

- Portul pe care este legat motorul la controller (indicația 1)

- Direcția de rotație a motorului (indiacția 2)

- Puterea de rotație a motorului (indicația 3)

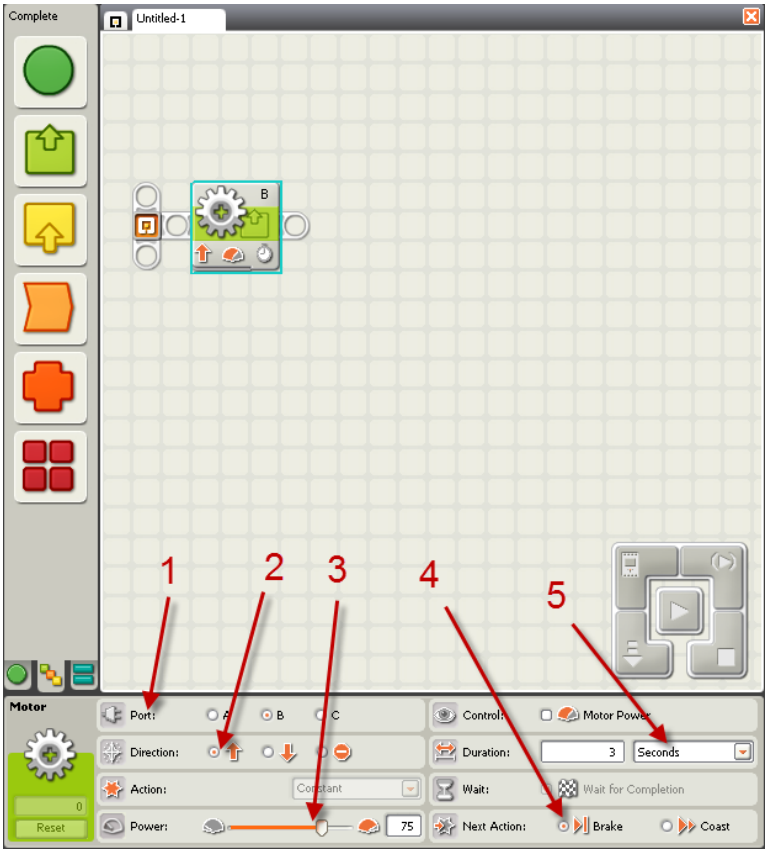
- Durata de rotație a motorului (indicația 5)

Aplicații:

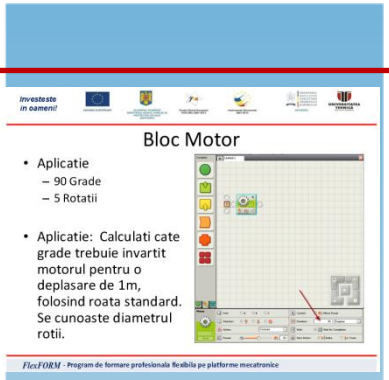
Să se comande motorului să meargă un anumit interval de timp.

Să se comande motorului să meargă nelimitat.

Să se comande motorului ca la terminarea acțiunii să frâneze liber sau forțat (coast/break) (indicația 4).



Blocul „Motor”

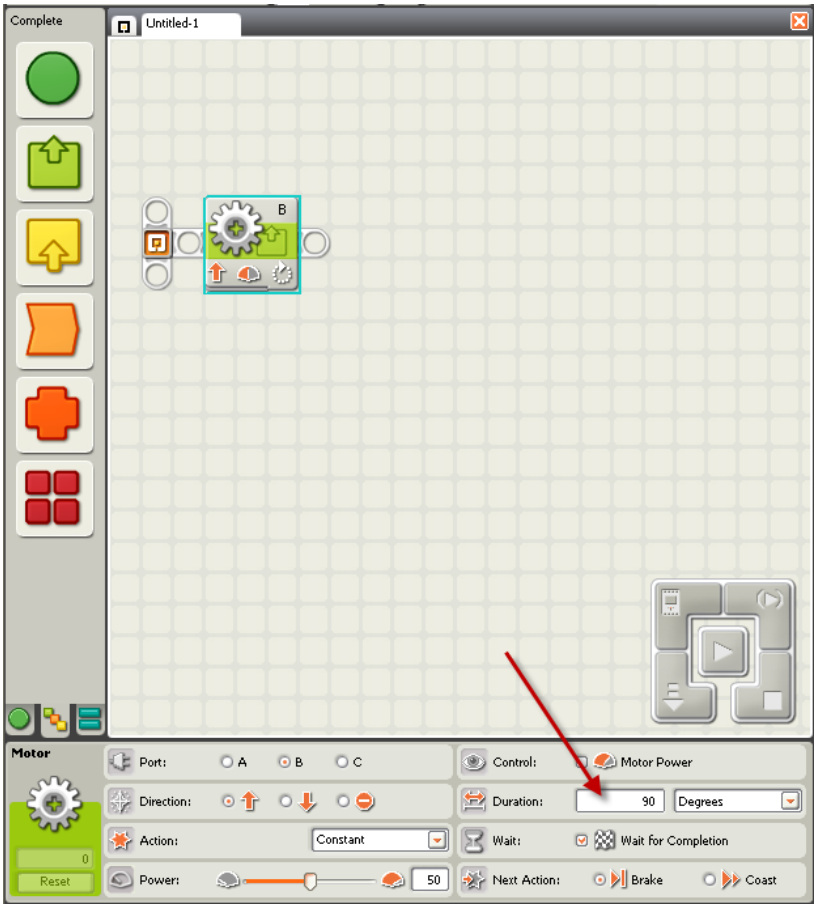


Aplicații (indicația 1):

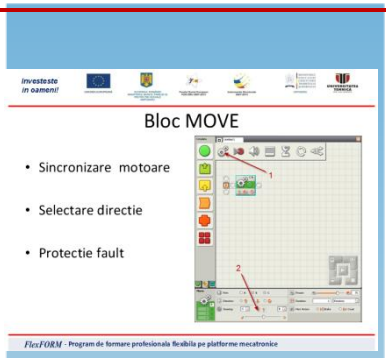
Să se comande motorului să se miște 90 de grade

Să se comande motorului să execute 5 rotații

Să se calculeze numărul de grade care tebuie executate de motor pentru a parcurge 1 metru liniar, folosindu-se roata din setul pus la dispoziție.



Blocul „move”

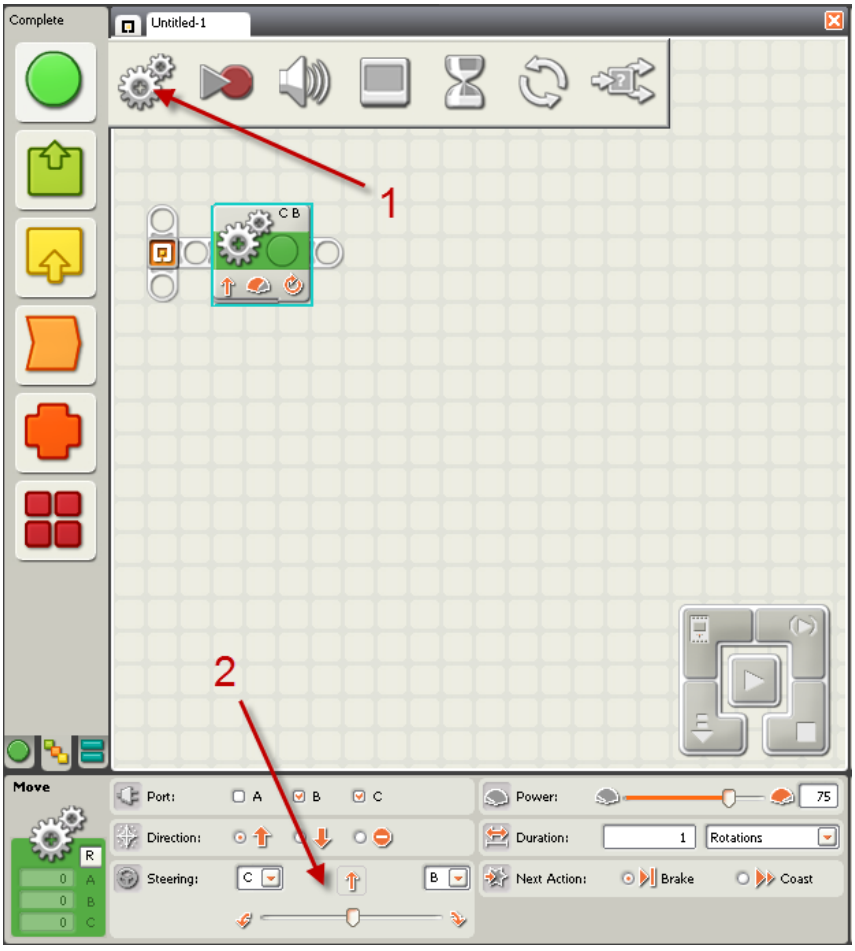


Blocul move se diferențiază de blocul motor prin posibilitatea de acționare a 2 motoare simultan.

Din meniul de blocuri se alege blocul move (indicația 1).

Se selectează porturile pe care sunt conectate motoarele (pot fi bifate mai multe în același timp)

Se poate seta direcția de deplasare relativă a celor două motoare, conform indicației 2.



Senzori



Controllerul NXT are în componență 4 porturi care permit conectarea la acesta a diverselor tipuri de senzori.

Senzorii sunt folosiți pentru a capta informații despre mediul înconjurător. Pe lângă senzorii „tipici” ai setului Mindstorms NXT (senzor de culoare, senzor de atingere și senzor ultrasonic) există o varietate mare de senzori compatibili dintre care amintim:

- Optici: senzor de lumină, senzor de culoare
- Ambientali: senzor de temperatură, senzor de sunet
- Mecanici: senzor de accelerație, senzor giroscopic
- Orientare: senzor tip busolă, senzor de distanță



Senzor atingere



Senzorul tactil este unul dintre cei mai simpli și cel mai intuitiv dintre toți senzorii NXT. Funcţionează aproximativ ca un buton de comandă. În momentul în care este acţionat, circuitul este închis şi curentul electric trece prin el.

Comanda cea mai utilizată a senzorului este de amortizare. Amortizoarele sunt cele mai simple modalităţi de a interacţiona cu mediul înconjurător; ele permit robotului să detecteze obstacolele în momentul în care le ating, aceştia schimbându-şi orientarea corespunzător.

Senzorul tactil poate avea trei aplicaţii: apăsat, lăsat sau de amortizare. Aceseta sunt importante deoarece obligă robotul să reacţioneze diferit, în funcţie de situaţie.

Una din variantele de programare a senzoului este din softul NXT-G si este următoarea: în tabloul de comandă există două subdiviziuni şi anume: selectarea portului pe care este conectat senzorul şi modul de acţionare al acestuia. În cea de-a doua subdiviziune se pot selecta funcţiile: apăsat, lăsat şi amortizare. Amortizarea înseamnă apăsarea rapidă a butonului şi eliberarea acestuia în mai puţin de 5 secunde.

A doua modalitate de programare este din meniul controllerului NXT. Această metodă este foarte utilă mai ales în momentul în care se dorește verificarea funcționării senzorului. De exemplu, dacă în program există o problemă în funcționarea senzorului, din meniul controllerului se poate verifica dacă senzorul are vreo defecțiune sau este o problemă de programare.



Senzor tactil



Aplicație:

Să se scrie un program care va afișa o imagine la apăsarea senzorului de atingere și alta în rest.

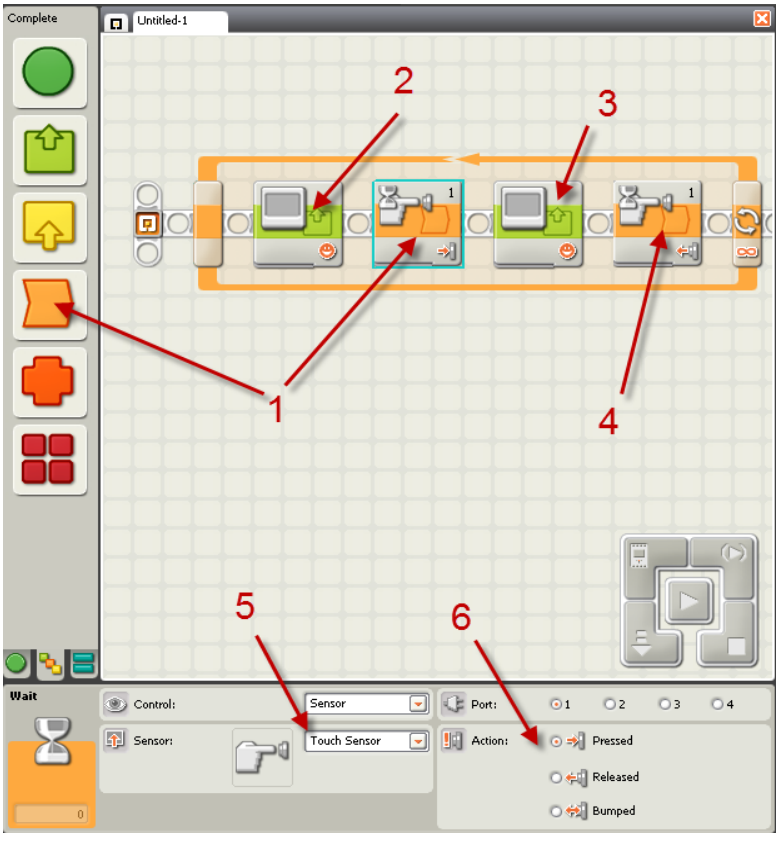
În prima fază se alege imaginea care apare inițial pe ecranul controllerului, dintr-un bloc de tip display (indicația 2).

Se alege din paleta “flow” blocul “wait” (indicația 1) căruia i se selectează senzorul de atingere (indicația 5) din panoul de comandă. Tot din panoul de comandă, din tipul acțiunii la care trebuie sa reacționeze senzorul (indicația 6), se alege ca acesta să reacționeze în momentul în care este apăsat.

Următorul bloc din program este tot unul de tip display (indicația 3) pe care este selectată o altă imagine, cea dorită în momentul apăsării butonului.

Ultimul bloc este unul de tip wait pe care este selectat un sensor tactil (indicația 4), astfel încât ultima imagine să fie reținută pe ecran până în momentul altei comenzi primită de senzor. Tot programul este pus într-un bloc de tip loop.

Cu ajutorul acestui program se pot testa cele trei faze ale senzorului.



Senzor tactil



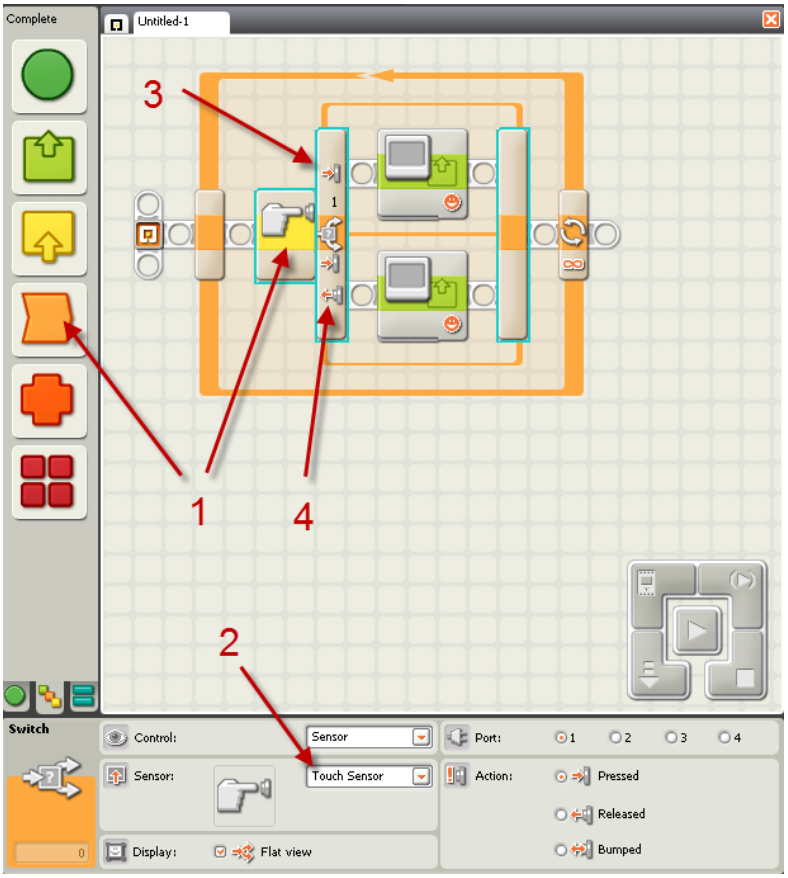
Aplicație:

Să se scrie un program care va afișa o imagine la apăsarea senzorului de atingere și alta în rest.

Pentru acest program este nevoie de un bloc de tip swith care are ca intrare un sensor tactil (indicația 1). Tipul senzorului se alege din panoul de comandă (indicația 2).

Se selectează acțiunea dorită, în acest caz se doreste ca butonul senzorului să fie apasat. În momentul în care senzorul este apăsat, se afișează o anumită imagine. Imaginea este selectată dintr-un bloc de tip display.

Blocul este amplasat pe ramura "true" a blocului, după cum se vede în indicația 3. În caz contrar, se afișează o imagine diferită tot cu ajutorul unui bloc display care este amplasat pe ramura de „false" a blocului, după cum se poate observa în indicația 4.



Senzor ultrasonic



Senzorul ultrasonic permite robotului să detecteze obiecte. El mai poate fi folosit pentru a evita obstacole, să măsoare distanţe şi să detecteze mişcarea.

Senzorul ultrasonic măsoară distanţa în centimetri sau inchi. Este capabil să măsoare distanţe de la 0 la 255 centimetrii cu o precizie de +/- 3 cm.

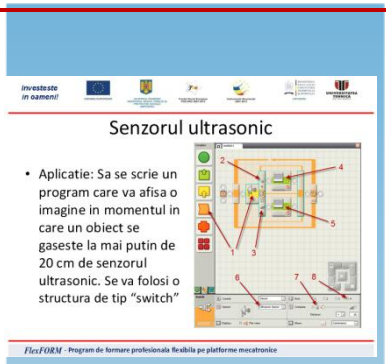
Principiul detectării ultrasonice este faptul că senzorul eminte ultrasunete de înaltă frecvență (peste limita auzului uman), care ricoșează de pe obiecte și este citită de senzor.

Timpul necesar întoarcerii undei sonore determină distanța la care se află obiectul. Cu cât intervalul de timp este mai mare, cu atât distanța față de obiect este mai mare. Liliecii folosesc același principiu pentru a se deplasa și a localiza prada.



Un mod comun de a programa senzorul este de a-l seta să verifice constant mediul înconjurător iar apoi să folosească valorile citite pentru a comanda robotul să reacționeze. (ex. Să ocolească sau să dea înapoi) Citirile sunt normalizate, primind valori de la 0 la 100.

Senzor ultrasonic



Aplicație:

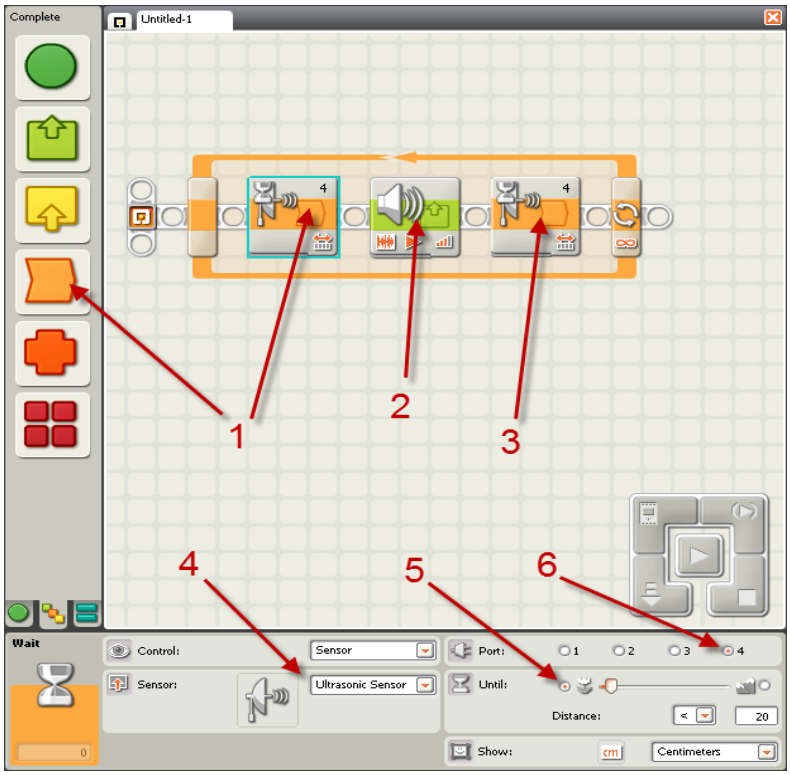
Să se scrie un program care va genera un sunet în momentul în care un obiect se apropie la mai puțin de 20 cm de senzorul ultrasonic.

Pentru rezolvarea aplicației, se va folosi un bloc wait. Acesta se alege din paleta „flow”, după cum se vede în indicația 1 din figura alăturată.

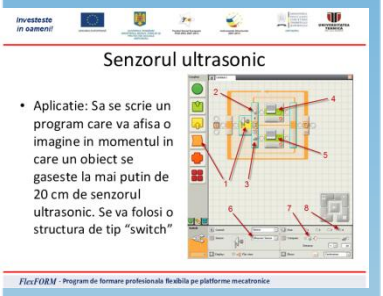
Din panoul de comandă se alege funcția dorită pentru blocul wait, adică senzorul ultrasonic căruia i se setează parametri doriți, adică pentru ce distanță să funcționeze (indicația 5) și portul pe care este pus senzorul (indicația 6).

Următoul bloc din program este blocul de sunet (indicația 2), deoarece robotul, în momentul în care ajunge la o distanță mai mică sau egală cu cea selectată la blocul anterior, să genereze un sunet selectat din meniul blocului.

Ultimul bloc este tot unul de tip wait căruia i se selectează tot un senzor ultrasonic (indicația 6). Toate trei blocurile sunt puse într-un bloc loop.



Senzor ultrasonic



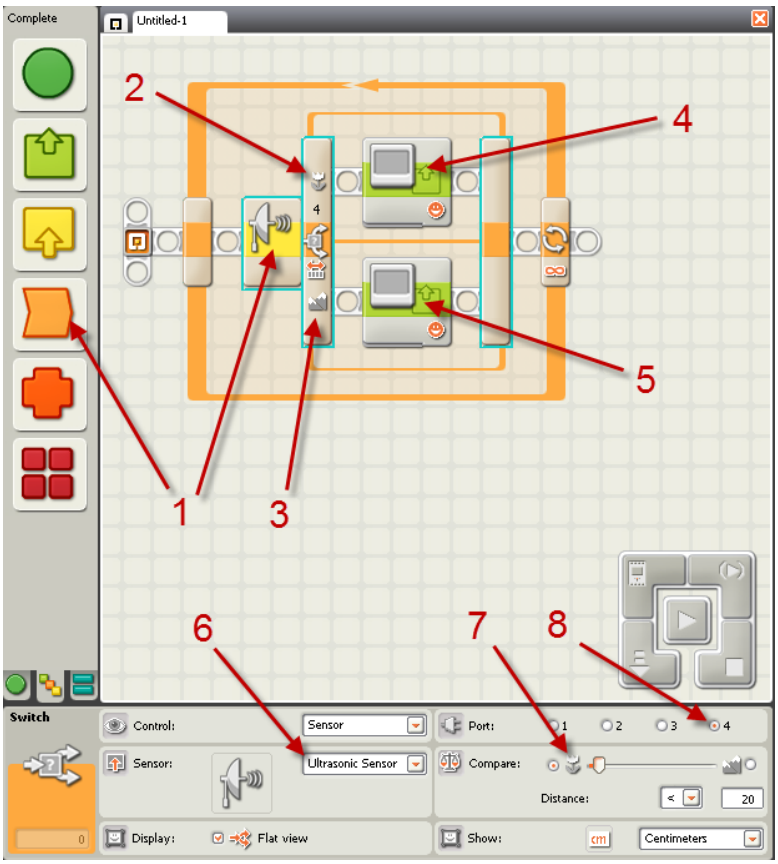
Aplicație

Să se scrie un program care va afișa o imagine în momentul în care un obiect se găsește la mai puțin de 20 cm de senzorul ultrasonic.

Pentru această aplicație se va folosi un bloc de tip switch (indicația 1), căruia i se alege un senzor ultrasonic din panoul de comandă (indicația 6).

Cu cursorul din indicația 7 se alege distanța pe care să o „vadă” senzorul, adică dacă este selectat 20 de centimetri, în momentul în care distanța e mai mică se execută blocurile din partea de „true” a blocului (ndicația 2), iar dacă valoarea e mai mare, programul din partea de „false” a blocului switch (indicația 3).

Când senzorul detectează un obiect la mai puțin de 20 de centimetri, va afișa o imagine (indicația 4), iar dacă obiectul este la o distanță mai mare, pe ecranul controllerului va fi afișată o altăţț imagine (indicația 5). Senzorul este conectat la portul 4 al controlerului (indicația 8).



Senzor ultrasonic



Senzorul ultrasonic are o rază mare de acțiune și nu depinde de intensitatea luminoasă din cameră (pentru că folosește sunetul). Pe de altă parte, emite și receptionează unde la unghi aproximativ 30° , chiar și un obiect punctiform apare detaliat după o scanare.

Senzorul are probleme cu detectarea anumitor materiale, de exemplu nu poate percepe materiale moi deoarece acestea absorb ultrasunetele, în timp ce o oglindă poate să reflecte ultrasunetele.

Senzor busolă

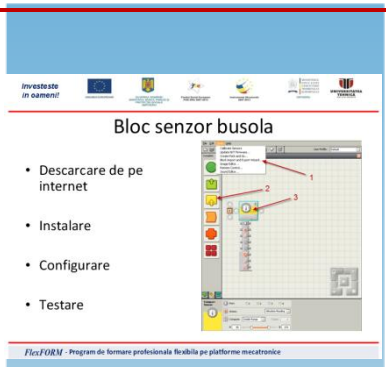


Busola este un instrument de navigare care se foloseşte pentru determinarea direcţiei relative faţă de polii magnetici ai pământului. Senzorii au în construcție un circuit electronic care detectează direcția liniilor câmpului magnetic terestru, aflând astfel orientarea polului Nord.

Principala problemă şi dezavantaj a acestor senzori constă în faptul că sunt uşor influenţaţi de orice câmp magnetic exterior.

Poziţionarea lor trebuie făcută atent pe robot pentru că orice sistem electric va genera câmp magnetic care are potenţialul de a induce în eroare senzorul.





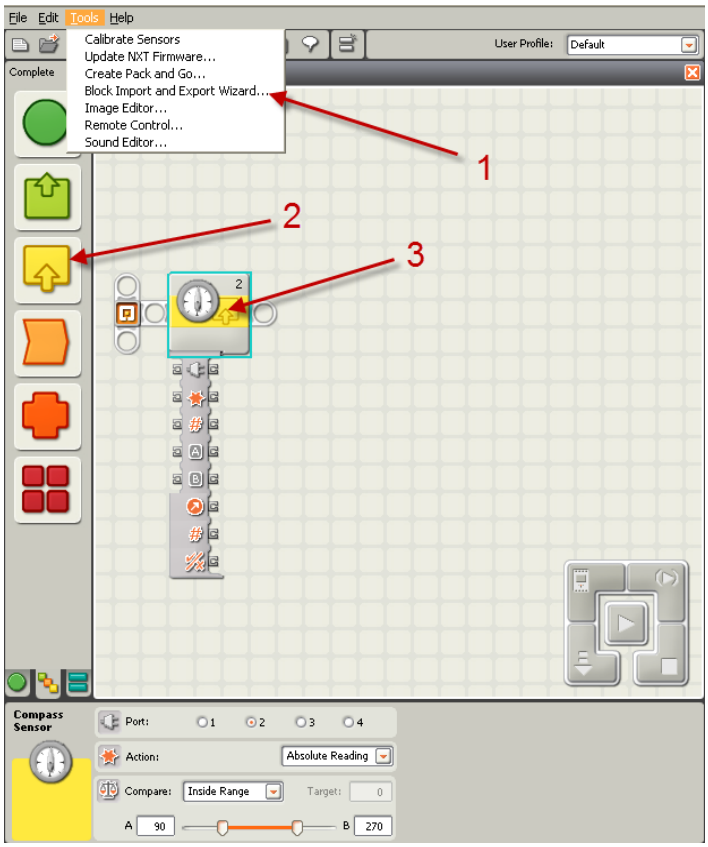
Bloc senzor busolă

Blocul ”Compass Sensor” este blocul care permite folosirea senzorului busolă.

Acest bloc nu se găseşte în mod normal în librăria standard de blocuri a programului Lego Mindstorm NXT şi pentru a putea fi folosit trebuie instalat.

Acest lucru se realizează folosind opţiunea ”Block Import Export Wizard” din meniul ”Tools”.

Valoarea citită de pe acest bloc este diferenţa între poziţia actuală şi direcția nord.



Folosind meniul View se poate vizualiza rapid valoarea citită de pe senzor şi astfel verifica buna lui funcţionare.

Folosind acest senzor se pot stabili punctele cardinale. În mod normal, senzorul are stabilită ca referinţă orientarea spre polul nord.

Introducând diferite defazaje se poate modifica punctul de referință spre care busola indică, adică pentru un defazaj de -90 de grade va indica direcția vest, pentru +90 de grade va indica direcția est iar pentru defazaj de 180 de grade direcția sud.

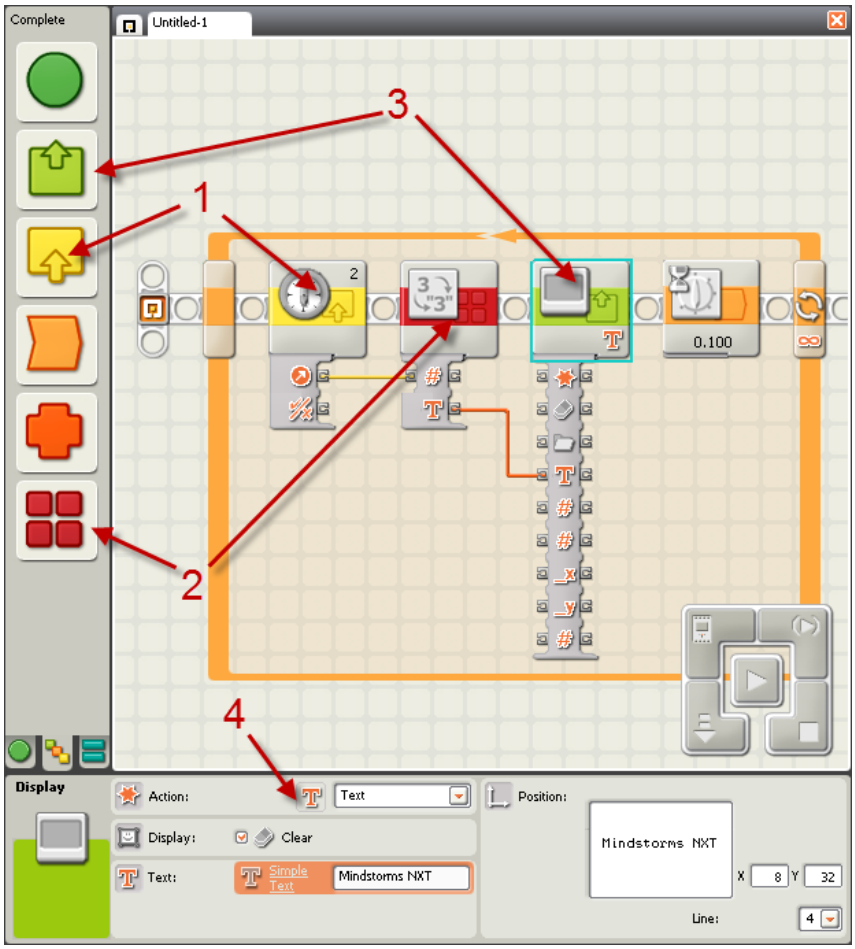
Senzor busolă



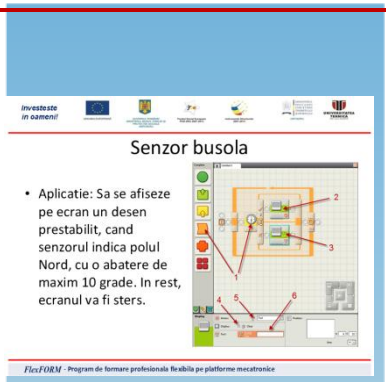
Aplicație:

Prima aplicaţie creată cu ajutorul senzorului ”Compas” va fi una care va permite vizualizarea valorii citită de pe senzor pe ecranul controlerului NXT. Pentru realizarea acestei aplicaţii vor fi necesare următoarele blocuri:

- Blocul ” Compass Sensor” cu ajutorul căruia se va citi de pe senzor diferenţa între pozitia curentă şi poziţia reală (indicația 1)
- Un bloc ”Number to text” care permite transformarea unei valori din format numeric în text (indicația 2)
- Blocul ”Display” care permite afișarea informaţiilor sub formă de text pe ecran (indicația 3)
- Un bloc ”Loop” pentru repetarea acţiuni
- Un bloc de wait pentru introducerea unei înârzieri



Senzor busolă

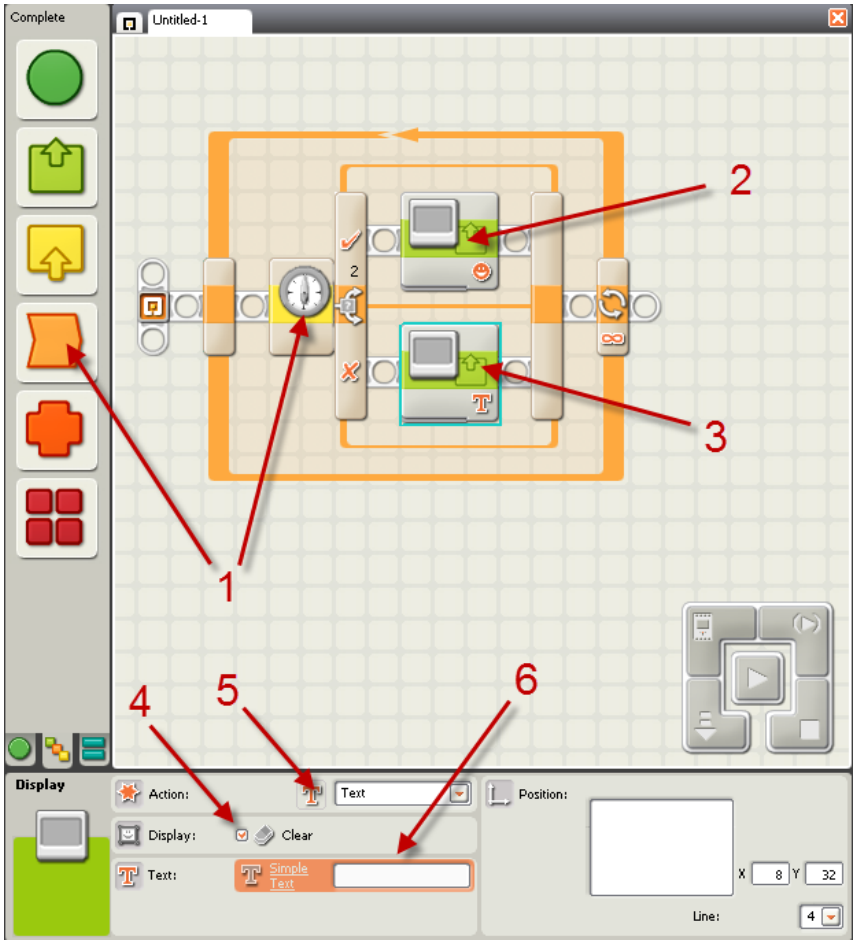


Aplicație:

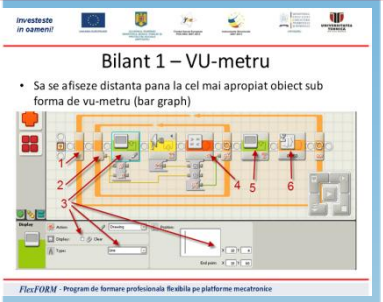
Această aplicaţie afişează un desen prestabilit atunci când valoarea citită de pe senzor este într-un anumit interval adică eroarea faţă de poziţia dorită este considerată acceptabilă pentru această aplicaţie .

Aplicaţia este realizată folosind următoarele blocuri:

- Un bloc ”Loop” pentru repetarea acţiuni
- Un block ”Swich” pentru determinarea imaginii de pe ecran, cu opțiunea „sensor – compass sensor” (indicația 1)
- Două blocuri ”Display” pentru afişarea sau ștergerea imaginii (indicațiile 2 și 3)



Bilanț 1 – VU-metru



Să se afișeze distanța până la cel mai apropiat obiect sub forma de vu metru (bar graph).

Se iau două blocuri loop (buclă) din paleta de blocuri și se pun conform indicației 1 și 2.

Se selectează în al doilea bloc loop opțiunea “show counter” și la control opțiunea “logic”.

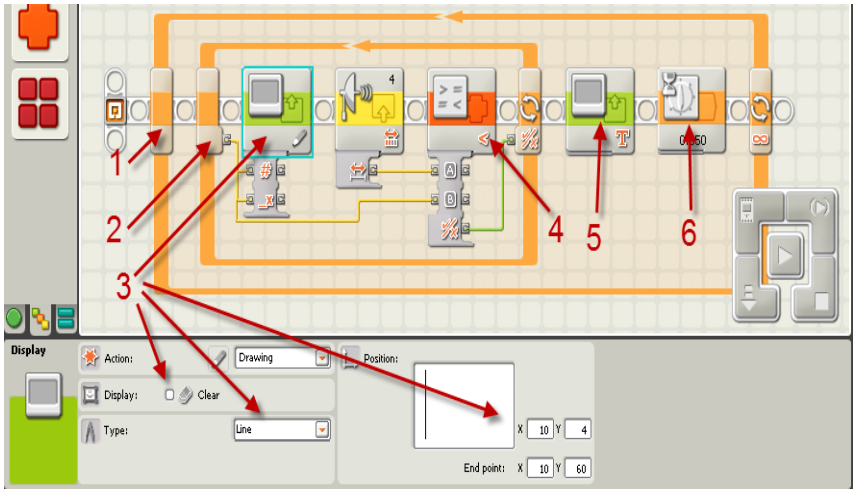
Se ia un bloc display și se pune conform indicației 3. Se selectează funcția drawing iar tipul de desen să fie linie (indicația 3). Din portul de counter se duce un fir de date la proprietățile “#” si “_X” ale blocului display.

Se introduce un bloc “ultrasonic sensor” și “logic block” cu proprietatea “<” (indicația 4).

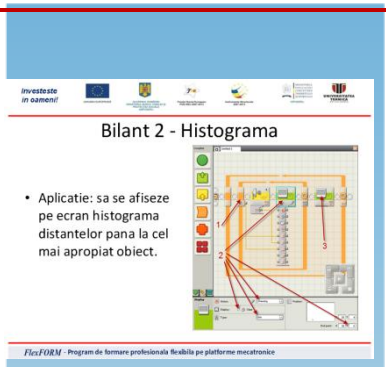
La intrarea A a blocului comparator se duce un fir de date de la senzorul ultrasonic, iar la intrarea B se ia un fir de date de la portul de counter al blocului loop. Rezultatul blocului logic se conectează cu un fir de date la valoarea de adevăr al blocului “loop” (portul din partea dreaptă).

Se mai introduc un bloc de tip display, cu proprietatea clear (indicația 5) și un bloc de tip “wait” setat la 0.05 secunde (indicația 6).

Se descarcă programul în controller.



Bilanț 2 – Histograma



Să se afișeze pe ecranul controllerului NXT histograma distanțelor până la cel mai apropiat obiect.

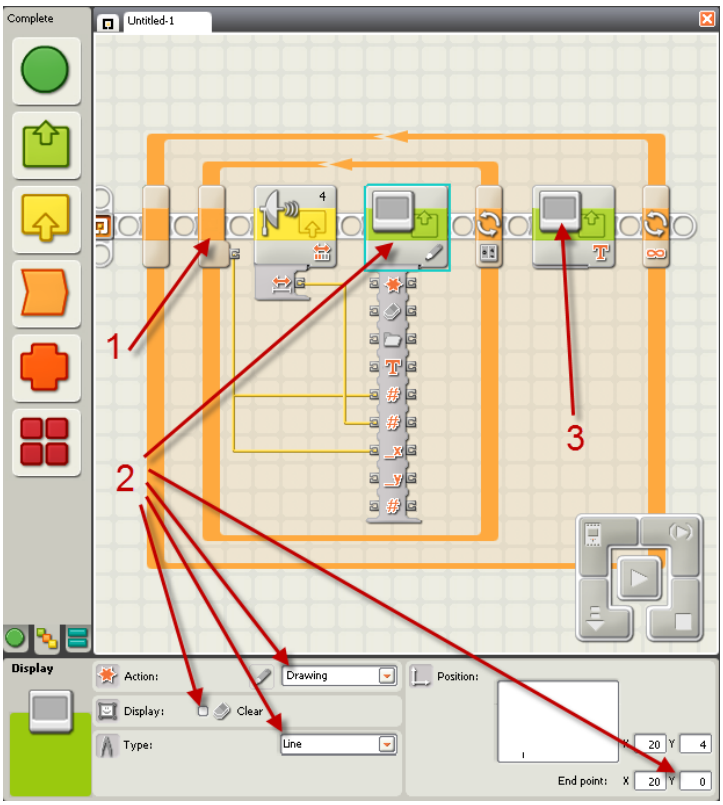
Se iau două blocuri “loop” (buclă) din paleta de blocuri și se pun conform indicației 1.

Se selectează în al doilea bloc loop opțiunea “show counter” și la opțiunea control tipul “counter”.

Se introduce un bloc “ultrasonic sensor” și un bloc “display” (indicația 2). Se selectează funcția drawing iar tipul de desen să fie linie (indicația 2). Se debifează optiunea clear.

Din portul de counter al buclei se duce un fir de date la primul port “#” al blocului display și la portul “_X” (indicația 2). Din blocul senzorului ultrasonic se duce un fir de date la al doilea port “#” al blocului display. Se introduce un alt bloc display cu optiunea clear conform indicației 3.

Se descarcă programul în controller.



FlexFORM – PROGRAM DE FORMARE PROFESIONALA
FLEXIBILA PE PLATFORME MECATRONICE

3. Creativitate

3. Creativitate. Learning by doing

Scopul activităţilor acestei zile este de a evidenţia importanţa cursurilor de tip „*hands-on*”. Este pus accentul pe beneficiile aduse de către acestea faţă de cursurile clasice. Se propun cursanților diverse metode de stimulare a creativităţii. Se explică în detaliu metoda „learning by doing”, metoda care va fi pusă în practică pe tot parcursul aplicaţiei practice.

Aplicaţia dezvoltată pe parcursul acestei zile se referă la realizarea unui sistem de orientare, care va fi echipat cu un senzor magnetic de tip busolă. Se propune realizarea unei busole mecanice şi programarea intuitivă a unui algoritm de control pentru aceasta. Se discută importanţa existenţei lanţului cinematic informaţional.

La finalul celei de-a treia zi, cursanţii vor fi capabili să realizeze un sistem mecatronic de orientare şi să-i aplice un algoritm de control. Vor fi pe deplin înţelese metodele de sporire a creativităţii şi conceptele de „learning by doing”, „trial and error” şi „hands-on” precum şi posibilităţile de a aplica aceste metode la clasă. (A4)

Noţiuni teoretice:

Este explicat conceptul de curs „hands-on”. Aceste cursuri (ore) oferă cursanţilor (elevilor) posibilitatea de a experimenta noţiunile teoretice învăţate pe măsura apariţiei acestora. Beneficiile majore constau în optimizarea timpului de învăţare (cursantul nu mai trebuie să recitească noţiunile teoretice înainte să le pună în practică) şi în imposibilitatea unor cursanţi (elevi) de a „rămâne în urmă”. În acelaşi timp, dascălul are un feed-back constant atât despre măsura în care au fost înţelese noţiunile teoretice cât şi despre capacitatea fiecărui cursant de a pune în practică noţiunile teoretice învăţate.

Se prezintă în continuare importanţa comunicării între membri echipei pentru sporirea creativităţii şi implicit a randamentului muncii depuse. Este introdus conceptul de „learning by doing”. Se pune accent pe ideea de a învăţa din greselile făcute, deziderat care stă la baza conceptului „trial and error”. Noţiunile teoretice sunt fixate pe parcursul aplicaţiei.

Aplicaţia:

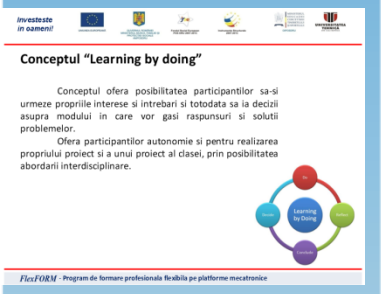
Fiecare echipă va trebui să realizeze un sistem minimalist de orientare. Nu vor fi impuse nici un fel de constrângeri constructive asupra structurii mecanice. Aceasta va fi echipată cu un senzor de tip busolă. Este pus accentul pe conceptul de „learning by doing”.(B9)

După finalizarea structurilor mecanice, fiecare echipă va justifica alegerile făcute pe parcursul construcţiei.

Echipele vor trebui să afişeze pe ecranul controlerului orientarea platformei faţă de polul nord şi unghiul cu care va trebui acţionat actuatorul (în cazul existenţei unui reductor mecanic) pentru a putea orienta platforma către polul nord.

Se vor programa platformele în aşa fel încât să se orienteze întotdeauna către polul nord, indiferent de poziţia de start. Se vor folosi metode intuitive de control. Se discută rolul tehnicii „trial and error”.

Conceptul “Learning by doing”



Conceptul “learning by doing” este o combinaţie între mediul de învățare şi mediul de lucru, de unde se nasc experienţe de învățare reale şi relevante. Acesta oferă posibilitatea participanților să-şi urmeze propriile interese şi întrebări şi totodată să ia decizii asupra modului în care vor găsi răspunsuri şi soluții problemelor.

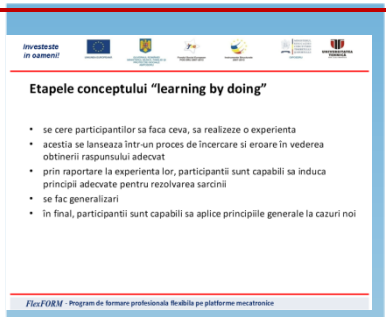
Oferă participanților autonomie şi pentru realizarea propriului proiect şi a unui proiect al grupului, prin posibilitatea abordării interdisciplinare. Participanții aplică şi integrează conținutul diferitelor discipline în momente autentice de acțiune, în loc de situații de izolare sau artificiale şi sprijină integrarea teoriei cu practica.

În comparație cu metoda tradițională de învățare, care se bazează pe construcție și reproductivitate, metoda „learning by doing” se bazează pe analiză, inițiativă și dedicare.

Exista anumite abilități necesare pentru ca participantul să poată câștiga anumite cunoștințe pe baza procesului:

- trebuie să vrea să se implice activ în experimente;
- trebuie să poată să reflecte asupra experimentului;
- trebuie să posede şi să utilizeze abilitați analitice să conceptualizeze experimentul;
- trebuie să știe cum poate folosi rezultatele obținute în alte exerciții.

Etapele conceptului „learning by doing”



Etapele învățării prin descoperire se ordonează oarecum invers faţă de învățarea tradițională.

Prima etapă în desfășurarea cursurilor de tip „learning by doing” este punerea problemei. Aceasta trebuie să fie destul de interesantă, pentru a capta atenția participanților, să stimuleze imaginația și să-i țină pe aceștia prinși în rezolvarea problemei.

A doua etapă este antrenarea participanților într-un proces de încercare şi eroare (trial and error) în vederea obținerii unui rezultat adecvat. Avantajele acestui proces sunt următoarele:

- Orientat pe soluție: procesul nu încearcă să vadă de ce funcționează soluția, ci doar dacă există o soluție;
- Specific la problema: procesul nu încearcă să generalizeze soluția la alte probleme;
- Non-optimal: procesul încearcă să găsească o soluție, nu toate și nu cea mai bună;
- Cunoștințe puține: procesul poate avea loc chiar și atunci când există cunoștințe puține sau deloc despre subiect;

În etapa a treia, participanții, prin raportare la experiența lor, sunt capabili să inducă principii adecvate pentru rezolvarea sarcinii, în urma cunoștințelor acumulate anterior.

În următoarea etapă se fac generalizări, prin referire la alte situații în care se pot aplica principiile găsite.

În final, participanții vor putea aplica noile cunoștințe acumulate la cazuri noi (să aplice regulile învățate în alte contexte)

Avantaje şi inconveniente ale conceptului „learning by doing”



Ca și avantaje ale acestui concept de învățare pot fi menționate următoarele:

Motivant – participanții au libertatea de a alege modul de abordare a problemei, pot folosi cunoștințele anterior obținute în vederea soluționării problemei, reprezintă o modalitate de a-și expune punctele de vedere prin experimente.

Memorare mai buna – Informația care ne vine pe cale senzorială vizuală nu este izolată de alte procese psihice. Alături de ea, apar anumite emoții și gânduri. În acel moment, informația se înmagazinează nu doar pe cale vizuală, ci corelata cu acele gânduri si emoții simțite.

Favorizează relațiile interpersonale

Creste încrederea în sine şi în ceilalți

Ca și inconveniente ale acestui concept sunt menționate:

Consumă timp – există riscul ca timpul necesar de a ajunge la o soluție prin metoda „trial and error” să fie insuficient.

Generalizarea este uneori dificilă – există posibilitatea ca participanții să nu poată face generalizarea soluției găsite fie din lipsa de informație fie din cauza informațiilor prea puține sau prea particulare.

Valoarea educativă a lecției scade datorită acțiunii de a descoperi în sine.

Stimularea creativității. Munca în echipă



Munca în echipă este definită ca „*o activitate de învățare, limitată în timp, prin care două sau mai multe persoane învață să execute, în ansamblu şi într-un mod interactiv, una sau mai multe sarcini mai mult sau mai puțin structurate, în vederea atingerii unor obiective determinate”,* conform cercetătorului în domeniu Jean Prolux.

Roger Mucchielli a identificat 7 caracteristici ale echipei ca grup primar:

a) Un număr redus de membri, condiție a creșterii eficacității;

b) Calitatea relațiilor interpersonale, exprimată printr-o rețea de legături vii ce se formează în timpul acțiunii echipei. În echipă, relațiile interumane joacă un rol esențial, membrii au conștiința apartenenţei la o formă de cultură comună;

c) Angajamentul personal are o semnificație aparte. Echipa nu este o colecție, o adunare de indivizi ci o totalitate, un grup psihosocial vivant şi evolutiv, o interdependență conștientă unde fiecare vine cu componenta sa, o unitate de acțiune;

d) Echipa este o unitate particulară în desfășurare, un organism în evoluție;

e) Echipa se prezintă ca o intenționalitate către un scop comun, acceptat şi dorit de către toți membrii;

f) Constrângerile în interiorul echipei sunt rezultatul orientării spre obiectivul comun;

g) Structura organizatorică a echipei variază în funcție de tipul de acțiune, de obiective şi de contextul specific;

Elemente constitutive ale echipei



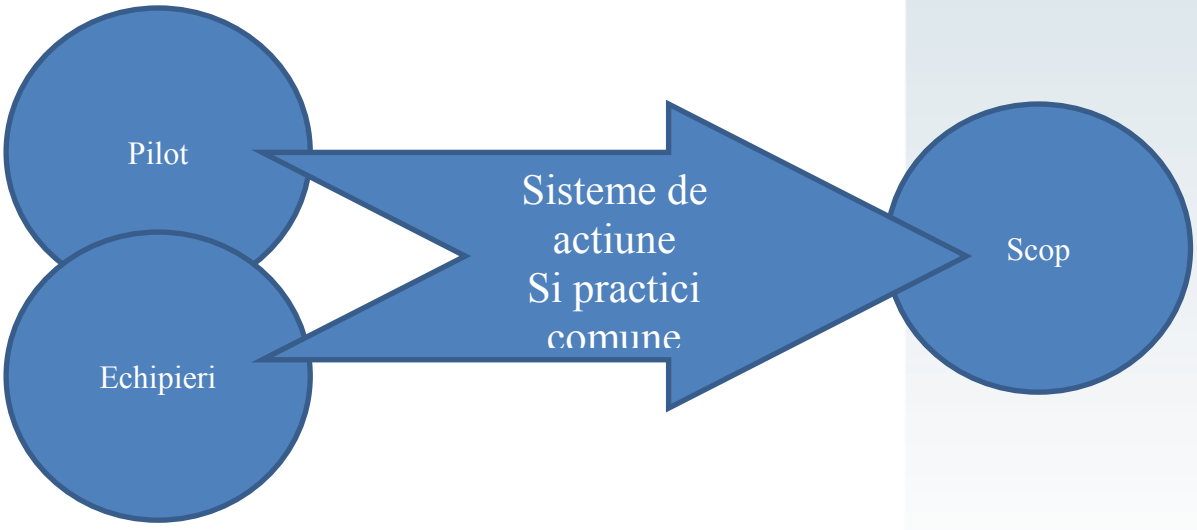
O echipă se compune din câteva „elemente constitutive”, după Devillard și anume:

Elementul pilot semnifică conducerea echipei, luarea deciziilor, asumarea responsabilităților în vederea funcționării și realizării obiectivului dorit. El poate fi reprezentat de un lider care-și asumă funcția de organizare si căruia i se recunoaște experiența și capacitatea de a se centra asupra anumitor obiective.

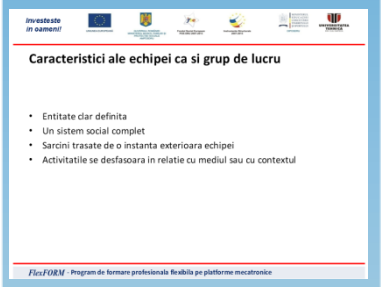
Grupul de echipieri reprezintă elementul rațional şi funcțional al echipei, aspectul uman, dimensiunea psihologică din care rezultă entitatea echipei, dimensiunea afectivă care asigură împrăștierea sentimentelor comune.

Scopul constituie ținta şi cadrul general de acțiune, fiind la rândul sau compus din mai multe elemente: strategia şi valorile entității, misiunea și obiectivele echipei, nevoile şi dorințele personale ale echipierilor, așteptările contextului.

Sistemul de acțiune reprezintă ansamblul proceselor, procedurilor şi modurilor de operare necesare în vederea realizării obiectivelor. El reprezintă câteva subsisteme care dau calitate acțiunii: de informații, de operare (normele de acțiune), de regularizare (utilizarea resurselor şi capacitatea de regenerare) şi cel de decizie.



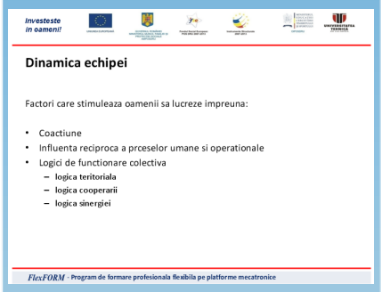
Caracteristici ale echipei ca şi grup de lucru



Claude Louche a identificat câteva caracteristici specifice ale echipei ca şi grup de lucru:

- Echipa constituie o **entitate clar definită**, în comparație cu grupul restrâns care poate fi mai puțin conturat, mai puțin net;
- Echipa constituie un **sistem social complet**: are obiective net definite, rolurile sunt diferențiate, căile de interdependență dintre participanți sunt precis trasate;
- **Sarcinile** sunt trasate de o instanță exterioară echipei, dar echipa este responsabilă pentru realizarea lor și poate fi evaluată;
- **Activitățile echipei** se desfășoară în relație cu mediul sau cu contextul în care echipa activează: alți participanți, moderator, juriu, care sunt atenți la răspunsurile echipei. Relațiile cu mediu se înscriu între obiectivele echipei, ca sarcini de realizat.

Dinamica echipei



Dinamica echipei se sprijină pe fundamente variate: motivațiile echipierilor, miza acțiunii, dozajul unității și diversității, puterea.

Trei factori stimulează oamenii sa lucreze împreună:

a) Stimularea provocată prin coacțiune, atracția fiecărui individ pentru ceilalți şi către o echipă ca model de identificare. Prezenţa altora stimulează, mobilizează, augumentează acțiunea de învățare;

b) Influența reciprocă a proceselor umane şi operaționale. Primul nivel este un proces obiectiv, constatabil, măsurabil: acţiunea în sine şi produsul acesteia. Al doilea nivel este acela care vizează grupul şi persoanele ce-l compun, este un proces subiectiv, implicând aspecte afective ale interacţiunii;

c) Conjugarea a trei logici de funcţionare colectivă presupune ca echipa să funcţioneze riguros şi flexibil în acelaşi timp, împărţind sarcinile şi rolurile, acoperind întreaga problematică, definind reguli şi respectându-le.

Cele 3 logici sunt logica teritorială, logica cooperării şi logica sinergiei. Prima se referă la definirea sarcinilor în „teritoriu” pentru fiecare membru; a doua presupune ca fiecare sarcină să fie abordată ca o operaţie colectivă, ceea ce-i va responsabiliza pe toţi să se implice în funcţie de sarcină şi situaţie; a treia semnifică dependenţa fiecărui echipier de ceilalţi, interacţiunea, competenţa de articulare reciprocă, capacitatea de a „face împreună” ceva, dezvoltând inteligenţa colectivă.

Cursuri de tip „hands-on”



Literatura de specialitate spune că „experienţele servesc ca un punct de start eficient în orice tip de învăţare”, conform cercetatorului Stefan Brall.

Modelul de învăţare „hands-on” presupune un ciclu de învăţare bazat pe experiment, repetat de două ori, în care fazele de experimentare, predare şi dezvoltare de noi strategii alternează. Pornind de la premisa că fiecare acţiune e încorporată în discuţii ce au avut sau au loc, experienţa imediată constituie punctul de start pentru învăţare. Modelele de învăţare bazate pe experiment, unde noutatea comportamentală este derivată din experienţă, descriu cum conceptele abstracte reies din observaţia şi reflecţia asupra experienţelor trăite.

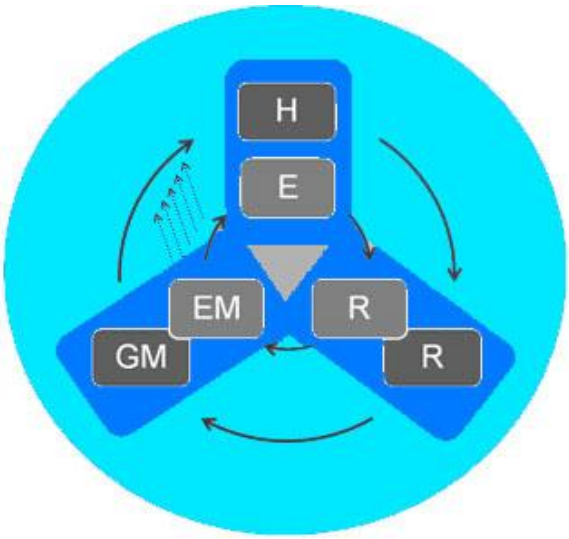
Toate modelele de învăţare bazate pe experiment identifică reflecţia ca fiind factorul decisiv al procesului de învăţare şi o consideră elementul principal în acesta, mai ales la adulţi.

Ciclul dual al învăţării bazate pe experiment



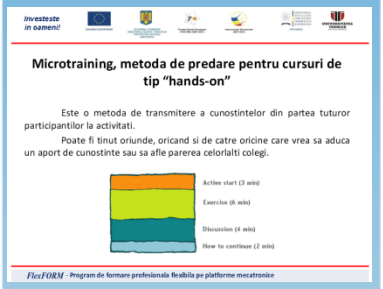
Conform acestui model, învăţarea bazată pe reflecţia asupra experientelor este conectată cu învăţarea care produce acţiuni: în primul ciclu, experienţa episodică (E) este transformată într-o experienţă a realităţii bazată pe model (EM) prin procesul de reflecţie (R). Punctul de start pentru acest proces de învăţare bazat pe experiment este diferenţa dintre cunoştinţe şi aptitudini experimentate fie prin experienţe anterioare sau prin acţiuni desfăşurate anterior.

Al doilea ciclu descrie aplicarea conştientă a unui asemenea model bazat pe realitate în procesul de învăţare, generand acţiuni (H). Reflecţia ce a creat noi experienţe dezvoltă modele generalizate de acţiune (GM) relativ stabile.



O dezvoltare ulterioară a cunoştinţelor şi acţiunilor cu acest procedeu este posibilă dacă se repetă ciclii de acţiune şi îmbunătăţire şi nu se bazează doar pe experienţele anterioare. Pe lângă observarea activă şi reflectivă a experienţelor anterioare din care decurg modele de acţiune viitoare, acest tip de învăţare permite participanţilor să gândească noi metode de abordare a problemelor.

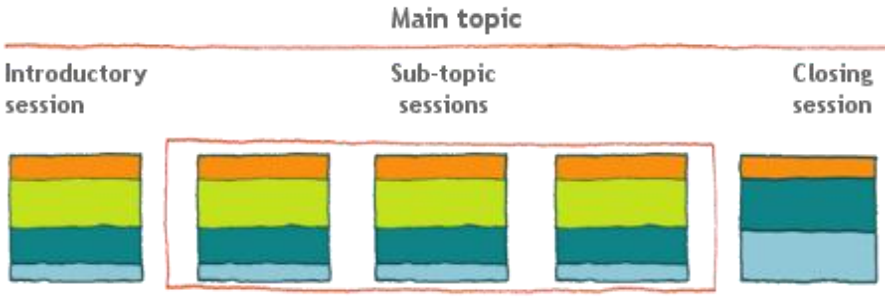
Microtraining, metodă de predare pentru cursuri de tip “hands-on”



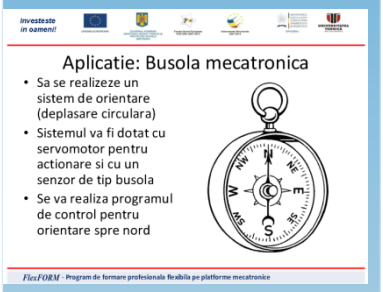
Microtrainingul este o metodă de transmitere a cunoştinţelor din partea tuturor participanţilor la activităţi. Poate fi ţinut oriunde, oricând şi de către oricine care vrea să aducă un aport de cunoştinţe sau să afle părerea celorlalţi colegi.

O sesiune de microtraining este o întâlnire scurtă, de aproximativ 15 minute. Se începe cu prezentarea, punerea în discuţie a problemei, urmat de un exerciţiu sau o demonstraţie a problemei realizată de catre participant, urmând ca la final să aibă loc discuţii asupra soluţiilor găsite, feedback şi o discuţie asupra cum se pot reţine cunoştinţele acumulate.

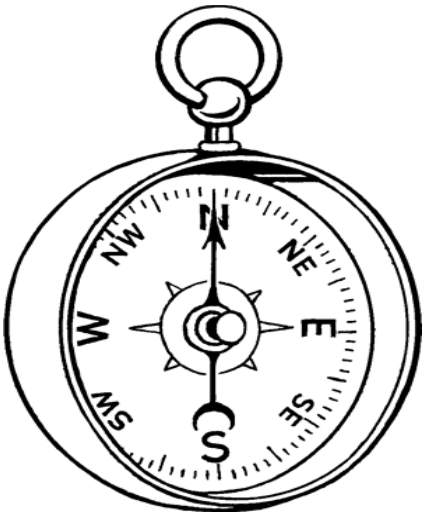
Câteodată este nevoie de mai mult de o sesiune de microtraining iar atunci se recurge la mai multe cicluri de microtraininguri. În sesiunea introductivă se defineşte problema iar în sesiunea de final se leagă sesiunile subtopicului respectiv. Se face legătura între acest topic şi următoarele cicluri de discuţii.



Aplicaţie: Busolă



Se propune o aplicaţie practică – sistem de poziţionare circular. Aplicaţia are în vedere punerea în practică a conceptelor prezentate pe parcursul acestei zile.



Se cere realizarea unei “busole mecatronice”, adică un sistem de poziţionare acţionat cu servomotor şi dotat cu un senzor de tip busolă.

Obiectivul principal al structurii este indicarea de către aceasta a polului nord terestru.

Ca şi obiective secundare putem aminti:

- crearea unei structuri mecanice viabile
- rezolvarea unor probleme de conectare
- conceperea unui algoritm de control
- optimizarea structurii

Structura mecanică – obiective



La construcția structurii mecanice trebuie urmărite urmatoarele obiective:

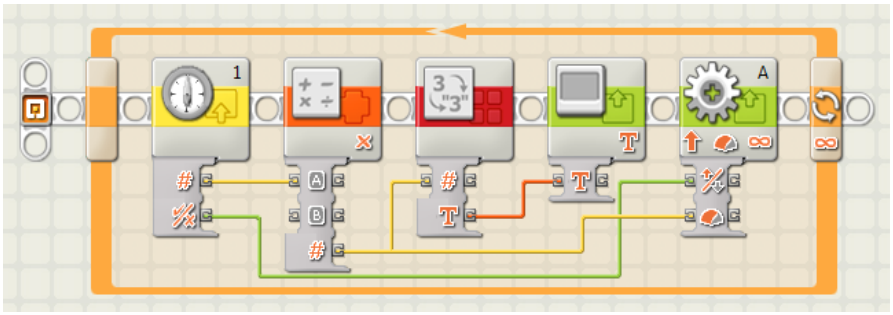
- Construcție compactă, ușoară – realizarea construcției dintr-un număr cât mai mic de piese. Trebuie pus accent pe rezistenţa la dezmembrare. Nici un element al structurii nu trebuie să fie încovoiat sau torsionat.
- Posibilitate de rotire 360 de grade – este necesar ca structura mecanică să permită orientarea busolei la orice unghi. În plus, ar fi bine ca structura să se poată roti oricât într-o direcţie sau cealaltă fără a se produce torsionarea sau întinderea firelor de legătură.
- Poziționarea senzorului departe de motor – pentru a elimina interferențele electromagnetice produse de bobina din interiorul motorului



Algoritm de control – obiective



- Să se citească valoarea de la senzor pentru determinarea orientării - acest lucru se realizează folosind blocul ”Compas Sensor”
- Să se acţioneze motorul - orientare spre polul Nord – această sarcină presupune menținerea motorului la o anumită poziție, controlul poziției se face folosind un control proporțional, determinarea abaterii fiind simplificată de faptul că informația citită de pe senzor este de fapt unghiul acestuia față de nord
- Optimizarea sistemului - timp minim, distanţa cea mai scurtă – principala modalitate de optimizare este calibrarea corectă a controlului P, adică alegerea unei valori cat mai potrivite pentru coeficientul Kp. Va fi necesară introducerea unui bloc de tip “math” în interiorul algoritmului pentru realizarea înmulțirii cu coeficienul Kp



Optimizarea structurii mecanice



- Eliminarea interferenţelor – interferențe electromagnetice pot fi introduse și de alte componente cum ar fi controlerul NXT; este necesară studierea posibilităţilor de poziţionare a senzorului pentru a minimaliza erorile
- Determinarea distanţei minime între senzor şi motor – determinarea poziției în care structura să fie cât mai compactă și în același timp senzorul să aibă o eroare minimă
- Reducerea frecărilor și a jocurilor mecanice – realizarea acestui obiectiv crește precizia și randamentul sistemului

Concluzii



Se pun în balanţă avantajele şi dezavantajele muncii în echipă.

Se va aprecia tipul de activităţi la care se pretează cursurile de tip "*hands-on*" şi se caută exemple din aria curiculară specifică fiecărui profesor în care se pot aplica concepte de predare precum "*learning by doing*" şi "*trial and error*".

Se propune un brainstorming (anticipare a zilei următoare) în legătură cu metodele de control ale structurilor mecatronice.

Investeste in oameni!

FlexFORM – PROGRAM DE FORMARE PROFESIONALA FLEXIBILA PE PLATFORME MECATRONICE

4. Metode interactive de lucru in grup

www.flexform.ro



OBIECTIV

- Prezentarea metodelor interactive de lucru în grup, astfel încât acestea să poată fi aplicate cu succes de către fiecare cursant la clasă.

COMPETENŢE

- Construcția unui sistem de poziționare liniară cu un număr minim de componente şi implementarea strategiei optime de control.
- Dobândirea unor cunoştinţe referitoare la două metode importante de lucru în grup („brainstorming” şi „thinking hats”) precum şi o metoda de evaluare (Răspunde, Aruncă, Întreabă)

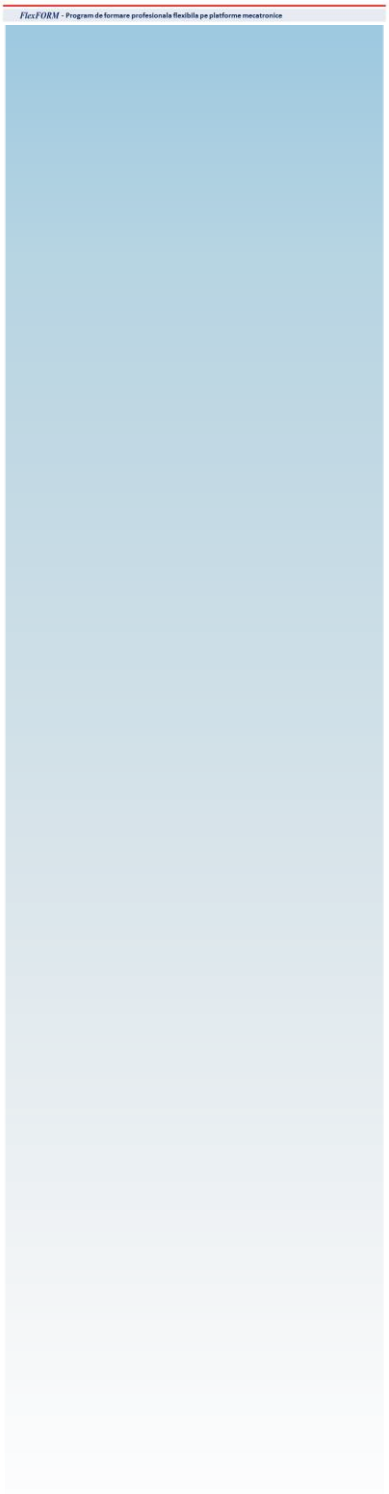
În a doua parte a zilei va fi conceput şi realizat un sistem mecatronic liniar de poziţionare. Sunt readuse în prim plan conceptele „learning by doing” introduse în ziua anterioară şi sunt reamintite strategiile de control învăţate în cadrul primului modul al acestui curs.

Noţiuni teoretice:

Se pune accent atât pe importanţa metodelor interactive moderne de predare cât şi pe importanţa aplicării acestora în cazul lucrului în echipă (grup). Sunt prezentate pe larg metodele „brainstorming” şi „thinking hats” împreună cu situaţiile concrete în care aceste metode pot fi aplicate cu succes. Este prezentată şi metoda de evluare şi fixare a cunostinţelor R.A.I. (Răspunde, Aruncă, Întreabă). Aceste noţiuni teoretice vor fi fixate în cadrul aplicaţiei propuse.(A5)

Noţiunile teoretice de natură tehnică, legate de construcţia şi controlul axei cinematice propuse vor fi prezentate în momentul în care este nevoie de acestea, în timpul aplicaţiei.





Aplicaţie:

Fiecărei echipe i se va acorda o oră pentru a concepe şi realiza prin metode de tip „learning by doing” un sistem liniar de poziţionare, bazat pe elemente LEGO. Singurele restricţii impuse sunt legate de lungimea utilă minimă şi rigiditatea mecanismului creat. (B10)

După expirarea timpului alocat, vor fi prezentate soluţiile constructive alese. Se acordă un punctaj fiecărei soluţii construite pe baza unor criterii şi al unui barem prestabilit. Se stabilesc cele mai bune trei variante propuse.

Se propune o discuţie de tip „brainstorming”, în urma căreia se decide care variantă constructivă este cea optimă şi ce îmbunătăţiri i s-ar putea aduce. Se acorda timpul necesar pentru ca fiecare echipă sa construiască câte o copie după design-ul declarat câştigător. Se testează buna funcţionare a sistemelor de poziţionare.

Sunt prezentate succint strategiile de control ON-OFF, P şi PID. Prin tragere la sorţi, fiecare echipă va trebui să implementeze una dintre cele trei strategii.

După expirarea timpului alocat, se vor observa rezultatele obţinute. Se propune o discuţie de tip „thinking hats” pentru a sublinia avantajele/dezavantajele fiecărei strategii de control în parte. Se trag concluziile.

Se propune o discuţie de tip „brainstorming” despre posibilităţile cursanţilor de a implementa la clasă o asemenea aplicaţie în lipsa componentelor de tip LEGO.

Ziua se încheie prin punerea în practică a metodei de evaluare RAI pentru a fixa cunostinţele acumulate în această zi.

Analiza metodelor de lucru în grup



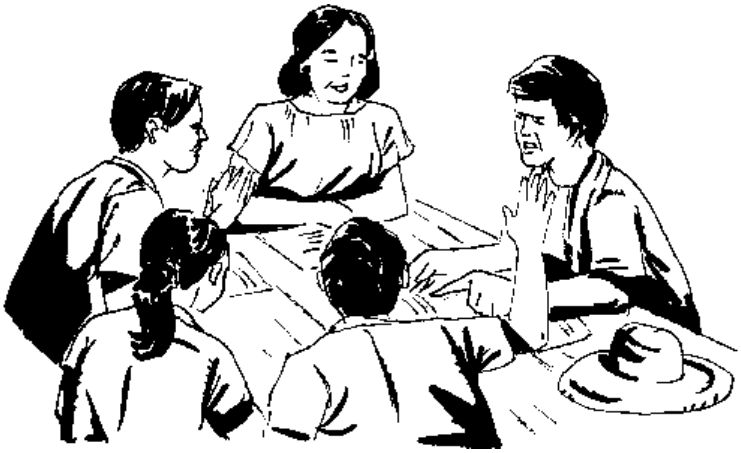
„Munca în echipă este o activitate de învățare, limitată în timp, prin care două sau mai multe persoane învață să execute, în ansamblu şi intr-un mod interactiv, una sau mai multe sarcini mai mult sau mai puțin structurate, în vederea atingerii unor obiective determinate”, conform cercetătorului în munca în echipă, Jean Proulx.

Metodele interactive de lucru în grup sunt mijloace didactice care favorizează schimbul de idei, de experiențe, de cunoştințe, ce ajută pe planul învățării şi dezvoltării personale.

Învățarea prin colaborare produce o confruntare de idei, de opinii, de influențare reciprocă prin implicarea directă şi activă a participanților.

Munca în echipă este un aspect al învățării prin cooperare, stârnind curiozitatea, ajutând la dezvoltarea gândirii critice, la conştientizarea informațiilor pe care participantul le ştie despre subiectul discuției, la aflarea de noi informații legate de subiect.

Experiența dobândită prin cooperare dezvoltă un anumit tip de competență, antrenând atât latura profesională, cât şi pe cea socio-afectivă şi anume competența colectivă.

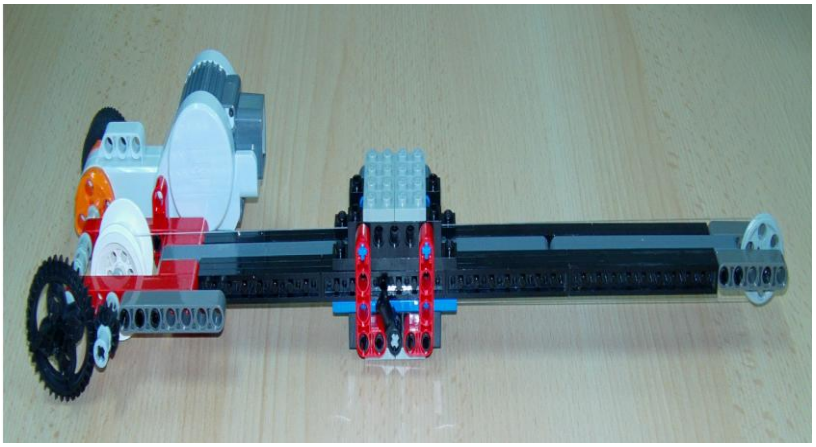


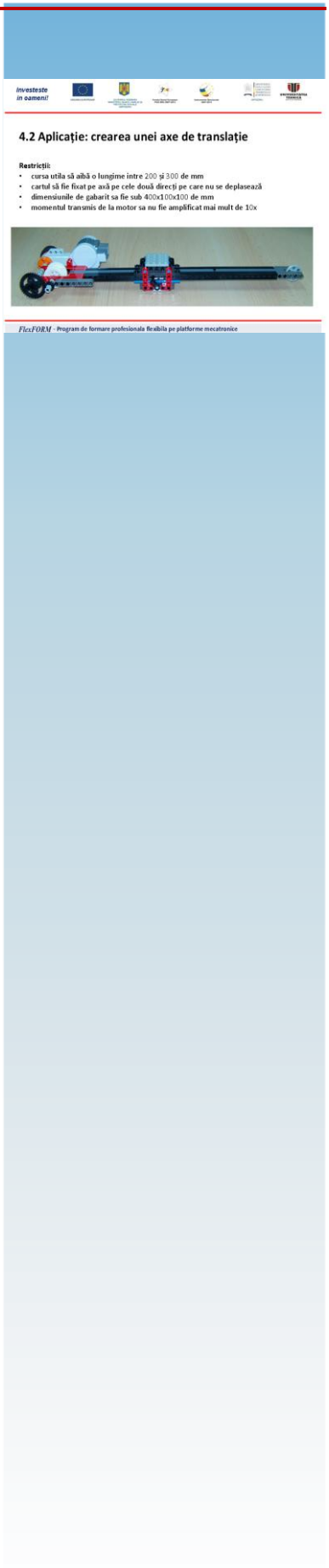
Aplicaţie: Crearea unei axe de translaţie

Se va construi o structură mecanică, utilizând componente LEGO, care va permite transformarea mişcării de rotație în mişcare de translație.

Restricții:

- cursa utilă să aibă o lungime între 200 și 300 de mm
- cartul să fie fixat pe axă pe cele două direcții pe care nu se deplasează
- dimensiunile de gabarit să fie sub 400x100x100 de mm
- momentul transmis de la motor să nu fie amplificat mai mult de 10x





Metoda brainstorming



Brainstorming-ul este un proces prin care membrii unui grup emit cât mai multe idei spontane legate de un subiect anume, pentru găsi rezolvări ale unor probleme.

În şedințele de brainstorming, provocarea o constituie ideile altora. Prin faptul că ideile provin de la altcineva, acestea pot servi ca stimuli pentru a genera alte idei. Chiar înțelegerea greşită a unei idei anterioare poate duce la o nouă idee bună. Deşi ideile nu sunt noi şi unei persoane i se par comune, în mintea altei persoane asocierea de mai multe astfel de idei poate stimula apariția unei idei cu totul originale.

Nici o idee nu este ridicolă. De obicei, ideile năstruşnice sunt binevente deoarece acestea ar putea inspira soluții mai bune decât cele obişnuite, convenționale, pot crea noi perspective sau pot diminua prejudecățile.

O abordare care este încurajată în procesul de brainstorming este cea de a combina şi îmbunătăți ideile existente, această abordare conducând la idei mai bune şi mai complete.

Este important ca pe parcursul şedințelor de brainstorming să nu se încerce evaluarea ideilor propuse. Cel care moderează şedința este dator să contracareze orice tendințe de evaluare, specificând acest lucru la început.



Definirea obiectivelor

Orice problemă poate constitui subiectul unei sedinte de brainstorming. Modul de formulare al problemei influențează major succesul abordării.

Dacă enunțul problemei este prea general, ideile generate sunt atât de disparate încât nu reuşesc să interacționeze pentru a provoca o reacție stimulativă în lanț, care constituie esența unei şedințe de brainstorming. Un enunț de genul „un control mai bun al traficului” este prea general.

Dacă enunțul problemei este prea restrâns, există riscul de a termina sedința din cateva idei referitoare nu la problema în sine, cât la un anumit mod de gestionare a sa. O formulare de genul „îmbunătățirea semaforizării” nu va genera idei referitoare la controlul traficului ci se va axa doar pe mijloacele de semaforizare luminoasă, atenția fiind îndepărtată de scopul discuției.

Datoria moderatorului este de a gasi cea mai bună exprimare a scopului în sine, iar dacă unul din membrii discuției consideră că acesta este prea funcțional, poate propune unul nou.

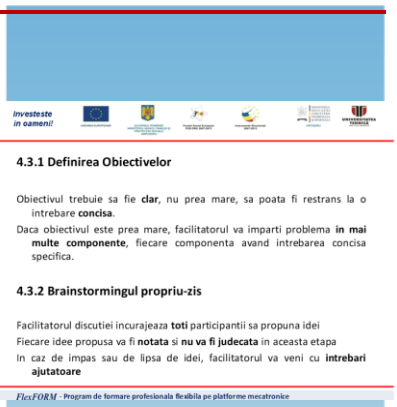
Brainstorming-ul propriu-zis

Principii generale pentru desfăşurarea sedinței de brainstorming:

- Nu trebuie să se emită critici sau evaluări
- Se poate spune orice, oricât de deplasat sau de ridicol
- Ideile nu trebuie elaborate cu detalii exagerate, sunt de ajuns câteva cuvinte
- Dați răgaz celui care ia notițe să scrie
- Moderatorul trebuie ascultat

Există în grup o persoană care ia notițe, rolul acesteia fiind de a converti într-o listă fluidă ideile formulate pe parcursul sedinței. Notițele trebuie să aibă sens şi după mai multă vreme, când contextul nu mai e aşa de limpede.

Persoana care ia notițele trebuie să atragă atenția dacă o idee similară a mai fost vehiculată, iar dacă are îndoieli despre acest lucru, să-l întrebe pe moderator.


Investeste in oameni!

4.3.1 Definirea Obiectivelor

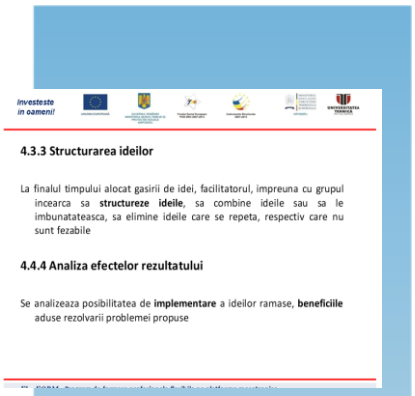
Obiectivul trebuie sa fie **clar**, nu prea mare, sa poata fi restrans la o intrebare **concisa**.
Daca obiectivul este prea mare, facilitatorul va imparti problema **in mai multe componente**, fiecare componenta avand intrebarea concisa specifica.

4.3.2 Brainstormingul propriu-zis

Facilitatorul discutiei incurajeaza **toti** participantii sa propuna idei
Fiecare idee propusa va fi **notata** si **nu va fi judecata** in aceasta etapa
In caz de impas sau de lipsa de idei, facilitatorul va veni cu **intrebari ajutatoare**

FlexFORM - Program de formare profesionala flexibila pe platforme mecatronice

Structurarea ideilor



La finalul discuțiilor, moderatorul împreună cu persoana care ia notițe afişează ideile prezentate în etapa anterioară, şi împreună cu participanții încep să discute pe marginea lor.

În această etapă se cere eliminarea ideilor care se repetă, eliminarea ideilor care nu sunt fezabile sau combinarea unor idei.

Analiza efectelor rezultatului

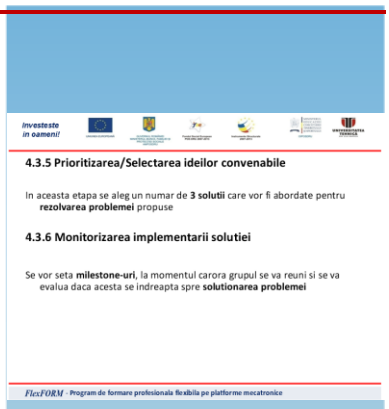
În aceasta etapă se iau pe rând fiecare idee rămasă în urma eliminării de la punctul anterior şi se discută. Se are în vedere posibilitatea de implementare a ideii, beneficiile care le aduce în soluționarea problemei, cât de bine rezolvă problema.

Ideile pot fi împărțite în mai multe categorii:

- Utile în mod direct
- Abordare interesantă
- Necesită o examinare ulterioară

După ce ideile au fost supuse dezbaterii, se face o rundă de vot, unde toți participanții aleg ideile care li se par a fi cele mai bune. În urma votului se elimină jumatate. Se reiau dezbaterile asupra ideilor rămase, apoi urmează încă o rundă de votare. După această rundă de vot se aleg primele 3 idei, care vor fi abordate pentru soluționarea problemei.

Monitorizarea implementării soluţiei



La o perioadă bine stabilită la finalul sedinței de brainstorming, echipa se va reuni pentru a urmări şi evalua implementarea soluțiilor alese în cadrul şedinței de brainstorming. Aceste puncte din proiect poartă numele de milestone-uri.

Milestone-urile sunt puncte identificabile de-a lungul unui proiect sau al unui set de activități care pot măsura progresul unui proiect şi prezintă sfârşitul sau începutul unei activități importante. Pot reprezenta şi finalitatea unor activități dintr-o categorie bine definită.

O metodă foarte des utilizată de a urmări mersul unui proiect este diagrama Gantt. Aceasta este un grafic care ilustrează etapele proiectului. Prezintă datele de început şi sfârşit ale activităților importante şi un sumar al elementelor proiectului. De obicei aceste grafice arată şi interdependenţa între etapele proiectului, astfel încât conducătorul de proiect să-şi poată da seama de necesarul de resurse umane, de logistică şi de timpul care poate fi acordat fiecărei etape.

Acest tip de grafic rezolvă problemele de gestiune a timpului unui proiect. Este uşor de citit şi de urmărit, poate fi asociat cu alte metode de vizualizare a mersului proiectului (de exemplu matricea sarcinilor) şi ajută la o gestiune mai uşoară a proiectului.

Concluzii



Brainstormingul este o metodă populară de interacțiune în grup, unde membrii unui grup propun idei legate de o problemă emisă.

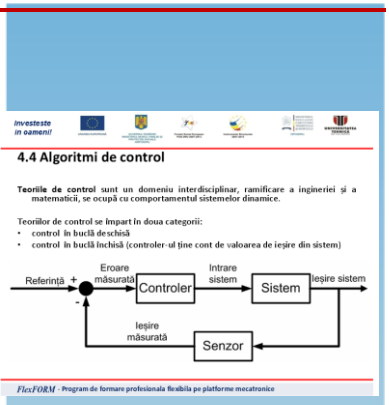
Poate fi propusă orice idee, oricât de deplasată sau ridicolă pare, deoarece aceasta poate duce la o abordare diferită a problemei şi la soluții mai bune, mai complexe.

La metoda brainstorming accentul se pune în primă fază pe cantitatea de idei emise, calitatea acestora urmând a fi supusă dezbaterii într-o etapă ulterioară.

Moderatorul poartă un rol important în discutie. Acesta emite problema astfel încât să fie înțeleasă de fiecare participant la discuție, să ajute la atingerea scopului propriu-zis îndrumând discuțiile.

Etapa de monitorizare a implementării soluției joacă un rol important în activitatea de brainstorming, aceasta urmărind bunul mers al activităților care duc la soluționarea problemei.

Algoritmi de control



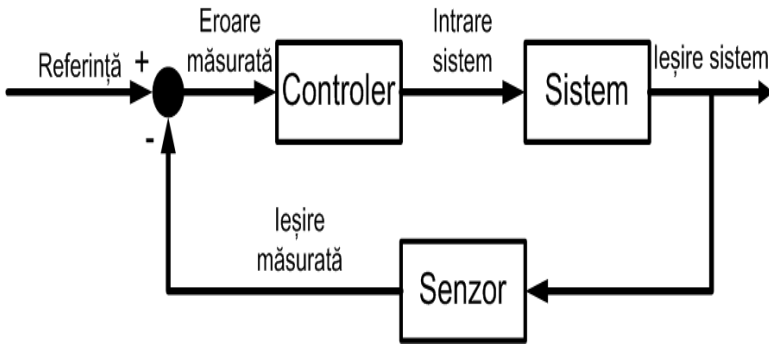
Algoritmii de control reprezintă un domeniu interdisciplinar, o ramificare a ingineriei și a matematicii, care se ocupă cu comportamentul sistemelor dinamice.

Teoriile de control se împart în două categorii:

- control în buclă deschisă
- control în buclă închisă

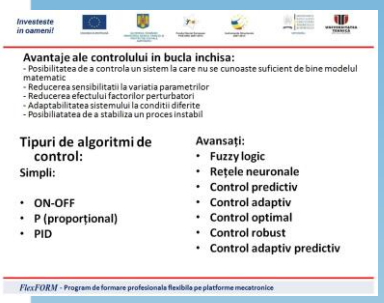
Controlul în buclă deschisă cuprinde toate formele de control care nu folosesc informații citite în timp real din sistem pentru controlul acestuia. Acest gen de control nu poate să-și autocorecteze valoarea controlată pentru a compensa o eroare apărută datorită unor perturbații asupra sistemului. Controlul în buclă deschisă este util pentru sistemele la care între intrare și ieșire există o relație bine definită și la care este destul de improbabil să apară factori perturbatori sau aceștia pot fi ignorați deoarece nu conduc la depăşirea toleranțelor admise.

Controlului în buclă închisă cuprinde toate formele de control care pentru a controla un sistem dinamic folosesc informații citite în timp real de la ieșirile sistemului. Citirea acestor informații se face folosind diverse categorii de senzori.



Schemă de control în buclă închisă

Avantaje ale controlului în buclă închisă



Principalele avantaje ale unui sistem în buclă închisă comparativ cu unul în buclă deschisă sunt:

- posibilitatea de a controla un sistem la care nu se cunoaște suficient de bine modelul matematic
- reducerea sensibilității la variația parametrilor
- reducerea efectului factorilor perturbatori
- adaptabilitatea sistemului la condiții diferite
- posibilitatea de a stabiliza un proces instabil

Tipuri de algoritmi de control

Câteva exemple de algoritmi de control in buclă închisă ar fi:

- ON-OFF
- P (proporțional)
- PID
- Fuzzy logic
- Rețele neuronale
- Control predictiv
- Control adaptiv
- Control optimal
- Control robust
- Control adaptiv predictiv

Controlul ON OFF



Controlul on-off (totul sau nimic) este cea mai simplă formă de control în buclă închisă. Acest tip de control foloseşte uzual două valori pentru comanda sistemului: o valoare pentru pentru pornirea sistemului (atunci când valoarea citită de pe senzor este mai mică decat valoarea referintei) şi una pentru oprirea sistemului (atunci când valoarea citită de pe senzor depășește această valoare). Diferența dintre cele două valori depinde de histerezisul dorit, adică de intervalul în care ieșirea sistemului poate varia.

Acest gen de control se folosește în domeniile unde sistemul are inerții mari și precizia nu are o mare importanță, principalul domeniu de utilizare fiind cel termic.

Unul din cele mai bune exemple pentru acest gen de control este termostatului pentru centralele termice de apartament. Acesta trebuie să mențină temperatura la o anumită valoare, să zicem de 21 de grade cu o variație admisă de plus minus un grad. Asta înseamnă că centrala porrneşte atunci când temperatura măsurată pe senzor coboară sub 20 de grade și se oprește atunci când temperatura trece de 22 de grade. Histerezisul în acest caz este de 2 grade.

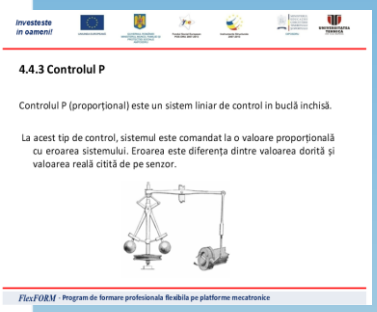
Avantaje:

- Ușor de implementat

Dezavantaje:

- Precizie mică
- Supracreştere – trecerea peste semnalul de comandă („overshoot”)
- Histerezis

Controlul P

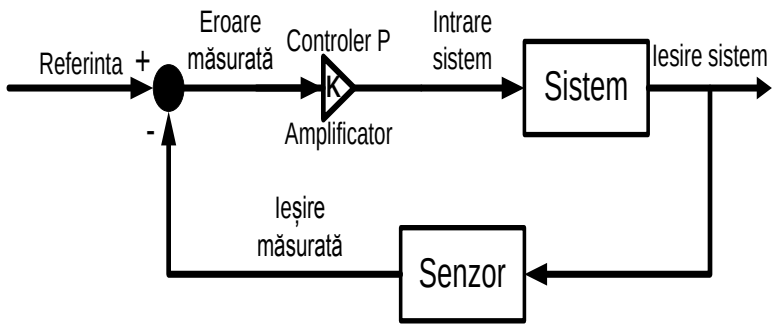


Controlul P (proporțional) este un tip liniar de control în buclă închisă.

La acest tip de control, sistemul propus menține semnalul de comandă la o valoare proporțională cu diferența (eroarea) între valoarea de referință şi valoarea semnalului de ieşire. Eroarea este diferența dintre valoarea setată ca referință, adică valoarea la care vrem să stea sistemul, și valoare citită de pe senzor.

Practic, un controler cu algoritm P este un amplificator care păstrează relația de proporționalitate între ieșirea măsurată și intrarea în sistem, amplificare realizată cu coeficientul K.

Un bun exemplu pentru implementarea unui astfel de controler este un robot mobil care trebuie să navigheze într-un labirint folosind regula mâinii drepte. Robotul trebuie să urmărească întotdeauna peretele care se află în dreapta lui păstrând o distanță constantă între el și perete. Această sarcină se realizează măsurând continuu distanța dintre perete și robot și ajustând direcția de deplasare a robotului în funcție de această distanță. În cazul în care pentru ajustarea direcției se folosește un control P, referința devine distanța la care robotul trebuie să se afle față de perete, ieșirea măsurată devine distanța reală între robot și perete, măsurată de senzor(de exemplu ultrasonic) iar eroarea devine diferența între acestea, adică abaterea robotului de la distanța stabilită. Intrarea în sistem (adică rezultatul algoritmului P) este o valoare proporţională cu abaterea robotului, valoare ce se calculează înmulțind eroarea cu coeficientul K.



Schemă implementare control P

Coeficientul K se alege în funcție de viteza cu care se dorește ca robotul să-și corecteze eroarea. De exemplu, un K prea mare poate face robotul să fie instabil sau prea lent, în schimb un K prea mic poate îl poate face să nu ia destul de strâns o curbă, ceea ce conduce la erori precum abaterea semnificativă de la o traiectorie: robotul poate sări peste o curbă sau se poate lovi de perete.

Un alt exemplu des întâlnit de control P este folosit la servomotoare, pentru controlul poziției. În acest caz intrarea în sistem este tensiunea de alimentare a motorului care este ținută la o valoare proporțională cu diferența între poziția dorită și poziția masurată.

Avantaje/dezavantaje ON-OFF și P



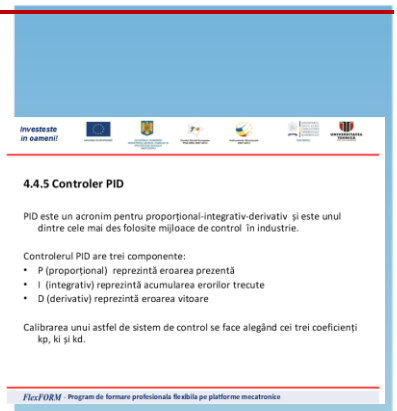
Principalul avantaj al controlului ON-OFF este reprezentat de simplitatea lui, ceea ce îl face ușor de implementat și mai puțin costisitor în comparație cu alte mijloace de control.

Controlul P, comparativ cu cel de tip ON-OFF, permite eliminarea oscilațiilor pe iesire, iar semnalul de control nu suferă salturi, ceea ce în multe situații reprezintă un avantaj.

Depaşirea sistematică a valorii de referinţă este o altă problemă care în cazul controlului ON-OFF este accentuată de modul specific de lucru şi de inerțiile din sistem. Comparativ, în cazul controlului P, aproprierea de valoarea de referintă implică şi micşorarea semnalului de control la valori care pot evita producerea suprareglajului (overshoot). Cu cât semnalul se apropie mai mult de referință cu atât mai mult se micșorează valoarea de comandă. În momentul când sistemul trece peste referință valoarea de comandă este zero, iar acest fapt face ca sistemul sa aibă o supracreştere mai mica sau chiar deloc în anumite cazuri. Pe de altă parte, în cazul controlerului de tip P, sistemul nu mai ajunge la referință şi ramâne în anumite cazuri sub valoarea semnalului dorit. Acest gen de eroare se numește eroare staționară.

Ca exemplu, în cazul unui sistem de poziționare, poziția finală poate fi diferită de poziția dorită (referință). Poziția finală este cea în care sarcina sistemului este egală cu puterea comandată de controler. Pentru că puterea este proporțională cu eroarea sistemului, sistemul are nevoie de o eroare pentru a se stabiliza.

Control PID



PID este un acronim pentru proporțional–integrativ–derivativ și este unul dintre cele mai des folosite mijloace de control în industrie.

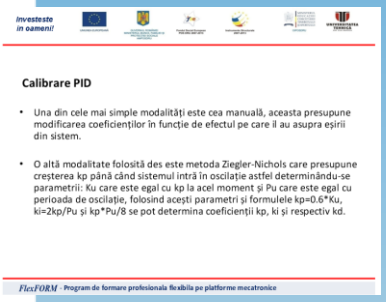
Controlerul PID are trei elemente:

- P (proporțional); componenta proporțională se calculează prin inmulțirea erorii cu un coeficient proporțional (k_p);
- I (integrativ); componenta integrală se calculează prin acumularea (integrarea) erorilor anterioare şi înmultirea cu un coeficient integral (k_i). Efectul important al acestei componente este eliminarea erorii staţionare specifice controlerului P.
- D (derivativ); componenta derivativă se calculează prin derivarea erorii şi înmultirea cu un coeficient derivativ (k_d). Efectul important al acestei componente este comportarea de tip predictiv. De exemplu, în cazul modificării bruşte a erorii (datorită modificarii bruşte a semnalului de referință sau datorită perturbațiilor), efectul componentei D va fi modificarea pentru un timp relativ scurt a semnalului de control astfel incât să se grăbească atingerea valorii de referință.

Determinarea PID-ului presupune calculul celor trei componente și însumarea lor. Este foarte importantă corelarea acestor trei componente astfel încât sistemul de control să fie stabil. Procesul de corelare se numeşte acordarea controlerului.

Un exemplu de sistem uzual care folosește control PID este liftul; în acest caz controlul P nu mai este suficient deoarece acesta va avea întotdeauna o eroare staționară care depinde de sarcina cu care lucrează. Acest gen de eroare poate fi eliminat doar de componenta I din PID.

Calibrarea parametrilor PID



Una din cele mai simple modalități de calibrare este cea manuală, aceasta presupune modificarea coeficienților în funcție de efectul pe care îl au asupra ieșirii din sistem.

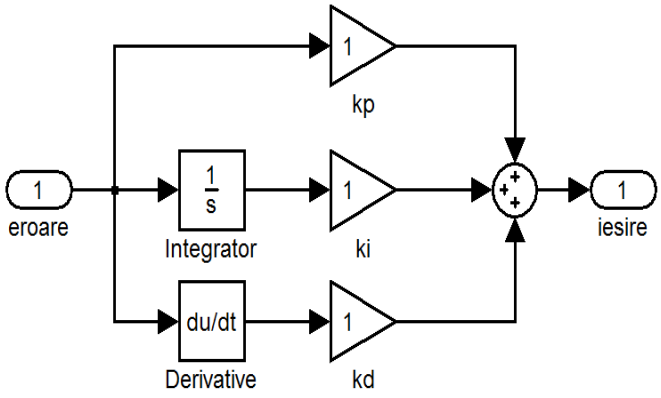
O altă modalitate folosită des este metoda Ziegler-Nichols care presupune creșterea kp până când sistemul intră în oscilație. Se determină astfel parametrii: Ku care este valoarea lui k_p în acel moment, Pu care este egal cu perioada de oscilație. Folosind acești parametri și formulele kp=0.6*Ku, ki=2kp/Pu și kp*Pu/8 se pot determina coeficienții k_p, k_i și respectiv k_d.

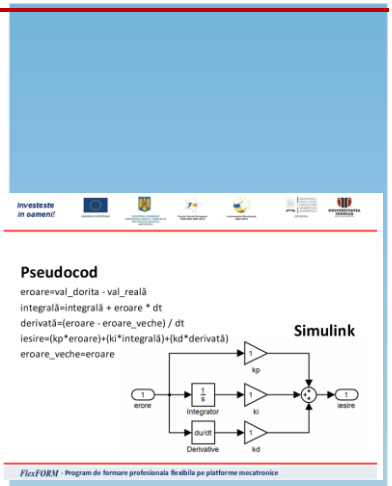
Variante de implementare PID

Pseudocod

- eroare=referinţa - valoare_reală
- integrală=integrală + eroare * dt
- derivată=(eroare - eroare_veche) / dt
- ieşire=(kp*eroare)+(ki*integrală)+(kd*derivată)
- eroare_veche=eroare

Simulink





Aplicație



Să se implementeze pe controlerul NXT al axei de translaţie un algoritm de control ON/OFF, P sau PID.

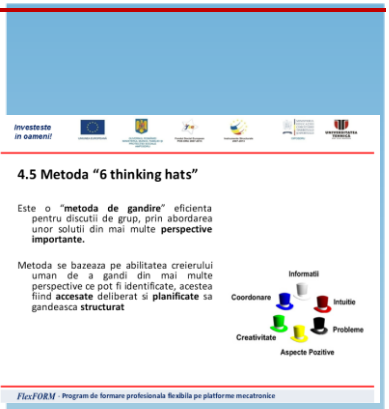
În funcție de algoritmul de control ales, se vor avea în vedere detaliile specifice de implementare.

Se va încerca o implementare de tip minimalist, folosind un număr minim posibil de blocuri şi o structurare cât mai concisă a acestora.

Se va avea în vedere calibrarea algoritmului de control propus.

Se va stabili eficiența metodei de control în funcție de viteza de reacție a sistemului, de eroarea staționară aparută şi de absența oscilațiilor.

Metoda „6 thinking hats” (“6 pălarii gânditoare”)



Aceasta este o metodă de gandire eficientă pentru discuții de grup, ce are ca scop abordarea unor soluții din diferite pespective importante.

Premisa acestei metode este abilitatea creierului uman de a gândi din mai multe perspective distincte care pot fi identificate, accesate deliberat şi astfel planificate să dezvolte strategii de gândire despre o problemă particulară. „6 thinking hats” te face să gândesti altfel decat modul obişnuit şi ajută la mai buna întelegere a situației.

S-au identificat 6 perspective, cărora li s-a asociat o culoare:

Informații (alb) – dă informațile disponibile;

Emoții (roşu) – se bazează pe primul sentiment sau pe prima reacție instinctuală la auzul problemei;

Judecata pesimistă (negru) – se bazează pe identificarea problemelor, barierelor care pot apărea în soluția propusă;

Judecata optimistă (galben) – se bazează pe identificarea beneficiilor soluției alese ;

Creativitate (verde) – se bazează pe provocarea şi investigarea unor noi posibilități;

Coordonare (albastru) – se bazează pe urmarirea mersului discuției, pe ansamblul soluției, clarifică informația.

Pentru a exemplifica modul de gândire al fiecărei pălării, vom analiza o problemă simplă:

„Elevii povestesc în timp ce profesorul predă”.

Pălăria albă



Cel care poartă pălăria albă trebuie să se imagineze ca fiind un computer care oferă imagini şi informații despre subiect. El trebuie să fie obiectiv şi neutru, trebuie să se concentreze strict pe problema discutată şi să relateze exact datele.

De obicei, pălăria albă prezintă atât informațiile cunoscute despre subiect, cât şi posibile necunoscute ale acestuia.

Ca şi exemplu a comportamentului pălăriei albe la problema propusă:

„Elevii vorbesc în timp ce profesorul predă";

„Este zgomot şi deci alți elevi nu pot fi atenți la ceea ce spune profesorul".

Pălăria roşie

Purtând pălăria roşie, gânditorul poate spune aşa: *"Aşa simt eu în legătură cu…"* Această pălărie legitimează emoţiile şi sentimentele ca parte integrantă a gândirii. Ea face posibilă vizualizarea exprimarii lor.

Pălăria roşie permite ganditorului să exploreze sentimentele celorlalţi participanţi la discuţie, întrebându-i care este părerea lor "din perspectiva pălăriei roşii", adică din punct de vedere emoţional şi afectiv. Cel ce priveşte din această perspectivă nu trebuie să-şi justifice sentimentele şi nici să găsească explicaţii logice pentru acestea.

Ca si exemplu a comportamentului pălăriei roşii la problema propusă:

„Profesorul se simte ofensat"

„Elevii sunt frustrați că nu aud instrucțiunile profesorului"

Pălăria neagră

Este pălăria avertisment, concentrată în special pe aprecierea negativă a lucrurilor. Gânditorul pălăriei negre punctează ce este rău, incorect şi care sunt erorile.

Explică ce nu se potriveşte şi de ce ceva nu merge; care sunt riscurile, pericolele, greşelile demersurilor propuse. Nu este o argumentare ci o încercare obiectivă de a evidenţia elementele negative. Se pot folosi formulări negative, de genul: *“Dar dacă nu se potriveşte cu…” “Nu numai că nu merge, dar nici nu…”*.

Gânditorul nu exprimă sentimente negative, acestea aparţinând pălăriei roşii, după cum aprecierile pozitive sunt lăsate pălăriei galbene. În cazul unor idei noi, pălăria galbenă trebuie folosită înaintea pălăriei negre.

Ca şi exemplu a comportamentului pălăriei negre la problema propusă:

„Învățatul este compromis”;

„Elevii prind doar fragmente din lecție”.

Pălăria galbenă

Este simbolul gândirii pozitive şi constructive, al optimismului. Se concentrează asupra aprecierilor pozitive, aşa cum pentru pălăria neagră erau specifice cele negative. Exprimă speranţa; are în vedere beneficiile, valoarea informaţiilor şi a faptelor date.

Gânditorul pălăriei galbene luptă pentru a găsi suporturi logice şi practice pentru aceste beneficii şi valori. Oferă sugestii, propuneri concrete şi clare. Cere un efort de gândire mai mare, beneficiile nu sunt sesizate întotdeauna rapid şi trebuie căutate. Nu se referă la crearea de noi idei sau soluţii, acestea fiind domeniul pălăriei verzi.

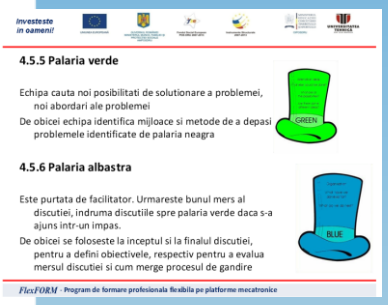
Ca şi exemplu a comportamentului pălăriei galbene la problema propusă:

„Toata lumea poate spune ce gândeşte”;

„Un elev poate spune o idee fără să aştepte, astfel micşorându-se riscul de a o uita”.



Pălăria verde



Căutarea alternativelor este aspectul fundamental al gândirii sub pălăria verde. Este folosită pentru a ajunge la noi concepte şi noi percepţii, noi variante, noi posibilităţi. Gândirea laterală este specifică acestui tip de pălărie. Cere un efort de creaţie.

De obicei echipa identifică mijloace şi metode de a depăşi problemele identificate de pălăria neagră .

Ca şi exemplu a comportamentului pălăriei verzi la problema propusă:

„Profesorul trebuie să încerce să interacționeze cu toată clasa”;

„Elevii vor putea să dezvolte idei ca urmare a creativității la clasă”;

„Elevii vor ține cont în momentul în care vorbesc dacă remarcile lor afectează învățatul altor elevi”.

Pălăria albastră

Pălăria albastră este coordonatoarea discutiei şi cere ajutorul celorlalte pălării. Defineşte problema şi conduce întrebările, reconcentrează informaţiile pe parcursul activităţii şi formulează ideile principale şi concluziile la sfârşit. Monitorizează mersul discutiei şi are în vedere respectarea regulilor.

Rezolvă conflictele şi insistă pe construirea demersului gândirii. Chiar dacă are rolul conducător, este permis oricărei pălării să-i adreseze comentarii şi sugestii.

Ca şi exemplu al comportamentului pălăriei albastre la problema propusă:

„Elevii realizează că îl fac pe profesor să se simtă nerespectat şi neapreciat”;

„Profesorul trebuie sa implice toţi studenţii în discuţie”

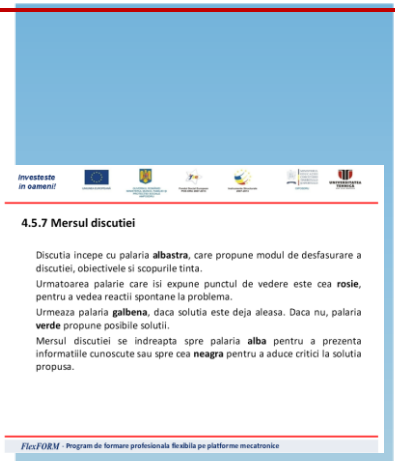
Mersul discuției

Discuția începe cu pălăria **albastră**, care propune modul de desfăşurare a discuției, obiectivele şi scopurile țintă.

Urmatoarea pălărie care îşi expune punctul de vedere este cea **roşie**, pentru a vedea reacții spontane la problemă.

Urmează pălăria **galbenă**, dacă soluția este deja aleasă. Dacă nu, pălăria **verde** propune posibile soluții.

Mersul discuției se îndreaptă spre pălăria **albă** pentru a prezenta informațiile cunoscute sau spre cea **neagră** pentru a aduce critici la soluția propusă.



Concluzii



Metoda „6 thinking hats” propune abordarea unei situații sau probleme din mai multe puncte de vedere, bine structurate, în vederea îmbunătățirii unei soluţii, a identificării unor probleme posibile în soluția aleasă.

Schimbarea perspectivei ajută la o înțelegere mai bună a problemei şi deci la găsirea unei soluții mai eficiente pentru rezolvarea acesteia.

S-au identificat 6 perspective din care trebuie urmărită problema: a informației cunoscute, a sentimentelor, a optimismului, a pesimismului, a creației şi a clarificării problemei.

Aplicație “6 thinking hats”

Să se determine, uitlizând metoda “6 thinking hats”, metoda de control cea mai eficientă pentru sistemul de pozitionare cu o axă construit.

Se va avea în vedere eficiența metodei, timpul de implementare şi dificultatea implementării metodei alese.



Aplicație brainstorming:



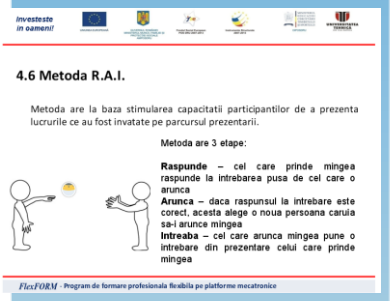
Posibilitățile de predare a noțiunilor acumulate la clasă, fără suport LEGO.

Se vor lua în considerare posibilitățile de a utiliza metodele şi tehnicile de predare învățate, în diverse situații apărute pe parcursul orelor cu elevii.

Se discută problemele care ar putea apărea în urma utilizării metodelor moderne de predare.

Se vor exemplifica aplicații din diverse arii curiculare, specifice disciplinelor predate la clasă, care se pretează la utilizarea acestor metode.

Metoda R.A.I



Metoda R. A. I. are la bază stimularea şi dezvoltarea capacităţilor elevilor de a comunica (prin întrebări şi răspunsuri) ceea ce tocmai au învăţat.

Denumirea provine de la iniţialele cuvintelor *Răspunde – Aruncă - Întreabă* şi se desfăşoară astfel: la sfârşitul unei lecţii sau a unei părţi de lecţie, profesorul, împreună cu elevii săi, evaluează cunoştinţele obţinute în urma predării-învăţării, printr-un joc de aruncare a unei mingi mici şi uşoare de la un elev la altul.

Cel care aruncă mingea trebuie să pună o întrebare din lecţia predată celui care o prinde. Cel care prinde mingea răspunde la întrebare şi apoi aruncă mai departe altui coleg, punând o nouă întrebare. Evident, interogatorul trebuie să cunoască şi răspunsul întrebării adresate.

Elevul care nu cunoaşte răspunsul iese din joc, iar răspunsul va veni din partea celui care a pus întrebarea. Acesta are ocazia de a mai arunca încă o dată mingea, şi, deci, de a mai pune o întrebare.

În cazul în care cel care interoghează este descoperit că nu cunoaşte răspunsul la propria întrebare este scos din joc în favoarea celui căruia i s-a adresat întrebarea. Eliminarea celor care nu au răspuns corect sau a celor care nu au dat nici un răspuns conduce treptat la rămânerea în grup a celor mai bine pregătiţi.

Este o metodă de a realiza o evaluare rapidă, într-un mod plăcut, energizant şi mai puţin stresant decât metodele clasice de evaluare. Se desfăşoară în scopuri constatativ-ameliorative şi nu în vederea sancţionării prin notă sau calificativ. Permite reactualizarea şi fixarea cunoştinţelor dintr-un domeniu, pe o temă dată.

FlexFORM – PROGRAM DE FORMARE PROFESIONALA
FLEXIBILA PE PLATFORME MECATRONICE

5. Dezvoltarea gandirii integratoare

5. Dezvoltarea gândirii integratoare

Noţiuni teoretice:

Se pune accent pe formarea unei gândiri sistemice, de ansamblu asupra unui sistem mecatronic. Este evidenţiată necesitatea unei gândiri integratoare. Sunt date câteva exemple de produse mecatronice din viaţa de zi cu zi care nu ar fi existat în absenţa acestui tip de gândire. (A6)

Sunt introduse noţiunile de gândire „top-down” a unui sistem, de la general la particular. Este amintită şi gândirea axată pe detaliu, de tip „bottom-up”.

Noţiunile teoretice de natură tehnică vor fi introduse pe parcursul aplicaţiei.

Aplicaţia:

Vor fi expuse conceptele legate de roboții paraleli, subliniindu-se diferenţele şi avantajele aduse de către aceştia faţă de roboţii seriali. (C1)

Se va prezenta în particular structura robotului pentalater. Cursanţii vor trebui să conceapă şi să realizeze un asemenea robot, utilizând conceptul de proiectare „top-down”. Proiectarea acestuia va face apel la toate noţiunile teoretice acumulate pe parcursul acestui curs.

Se va prezenta cinematica inversă şi modalitatea de implementare practică a acesteia. Se discută metodele de implementare a formulelor matematice complexe în mediile grafice de programare.

Se alege o strategie de control a structurii, bazată pe experienţa din ziua anterioară privind strategiile de control. În cazul apariţiei divergenţelor de opinie intre cursanţi, se poate aplica metoda „thinking hats” pentru a găsi o rezolvare la această problemă.

Se implementează algoritmul ales şi se recurge la acordarea acestuia. Se foloseste o metoda de tip „bottom-up” pentru acordare.

Robotul va fi controlat de la distanţă prin intermediul comunicaţiei radio.

Se discută posibilitatea implementării unui sistem master-slave utilizând câte doi roboţi conectaţi prin legătura radio.

Integrarea în natură şi în tehnologie

Dacă ne imaginăm o axă a materiei, a energiei şi a informaţiei, cu origine comună (deoarece aparţin aceluiaşi sistem), pe care înscriem vectori cu valoarea elementului respectiv, reprezentarea unui sistem va fi de forma unei elipse în spaţiu (figura 1):

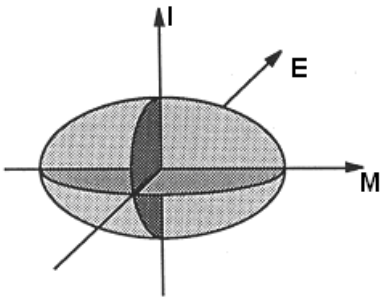


Fig. 1 Aspectul general al sistemului:

I- axa informaţiei, E- axa energiei, M- axa materiei

Dacă sistemul transferă o parte a elementelor (material, energetic sau informaţional), unui alt sistem, ambele sisteme se transformă, generând două noi sisteme cu proporţii diferite între cele trei elemente. Cum în natură există o infinitate de sisteme, axele lor având aceaşi orientare, vor genera un câmp material, un câmp energetic şi respectiv un câmp informaţional.

Într-un sistem, cu cât valoarea unui vector este mai mare, cu atât va scădea valoarea relativă a celorlalţi doi vectori. Astfel, dacă un vector tinde către infinit, ceilalţi doi tind către zero, şi se poate ajunge la o situaţie de “puritate relativă” a unui sistem (figura 2):

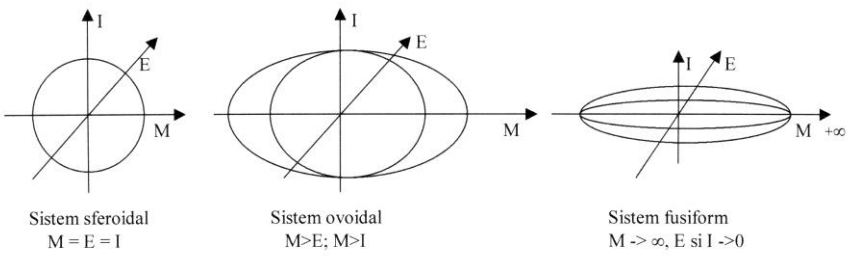


Fig. 2 Situaţia de “puritate relativă” a unui sistem

Prezentat în dinamică, în cazul unui sistem în care vectorul material tinde ca valoare către infinit, aspectul grafic va fi ca în fig. 3.

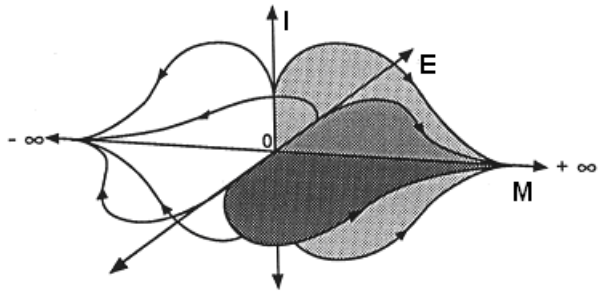


Fig. 3 Sistem în dinamică, cu M tinzând spre infinit

Pentru fiecare sistem, pe cele trei axe există un punct 0 (zero), *punctul de echilibru*, de la care pornesc în direcţii diferite vectori pozitivi şi negativi. Vectorii pozitivi reprezintă orientarea forţelor proprii ale sistemului în scopul supravieţuirii, iar vectorii negativi acele forţe ce au tendinţa de a distruge sistemul, stricându-i echilibrul.

Orice agresiune, reprezintă dezvoltarea unui vector negativ şi va genera simultan o imagine corespondentă pe aceeaşi axă, dar în direcţie opusă, cu scopul de a se menţine integritatea sistemului.

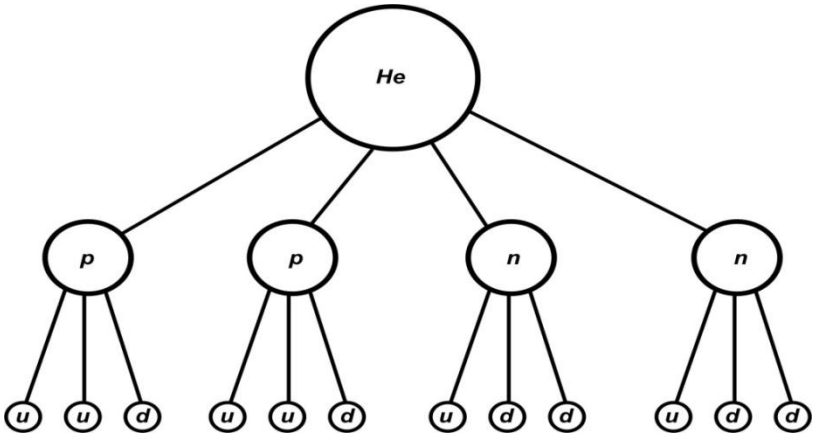
Un sistem este în echilibru atunci când valoarea vectorilor pozitivi şi negativi corespondenţi este egală, iar agresiunea corespunde cu modificarea valorii unuia sau mai multor vectori ai sistemului. Dacă prin propriile forţe sistemul nu reuşeşte să se echilibreze prin realizarea unei valori echivalente vectorului negativ, în cazul sistemelor închise se produce distrugerea sistemului, iar pentru sistemele biologice (deschise) apare starea de boală, ce poate culmina cu moartea.

Echilibrul universal îşi găseşte exemplificarea în religie prin unitatea contrariilor “rai - iad”, în mitologie prin “bine- rău”, în fizică prin “materie - antimaterie”, iar în biologie prin sistem “excitator – inhibitor”, “sistem coagulant – anticoagulant”, “aparat flexor - extensor”, etc. La nivel Universal, totul poate fi rezumat la echilibrul “+” şi ” -”.

Principiul superizării

Din pricina efectului de sinergie, sistemul înseamnă mult mai mult decât suma părţilor.

Schema procesului de superizare:



Integrarea este însoţită de *procesul de superizare*, deoarece întregul sistem are proprietăţi emergente pe care nu le au componentele (exemplu: celulele se înmulţesc, macromoleculele nu au aceasta proprietate). Pentru a pune în evidenţă acest proces, exemplificăm atomul de He.

Problema integrării este esenţială în mecatronică. În realizarea diferitelor produse şi sisteme, trebuie găsite soluţii specifice pentru integrarea componentelor mecanică-electronică-informatică. Până în prezent sunt validate două componente: integrarea în modul hardware şi integrare în modul software.

Principiile ordinii şi organizării

Acest principiu ar putea rezolva multe paradoxuri ale principiului al doilea al termodinamcii. Ordinea nu poate fi distrusă dacă cineva sau ceva nu o generează.

Nu putem vorbi de un principiu al dezordinii (entropiei) în absenţa principiului ordinii şi organizării.

Din jocul principiilor celor două contrarii rezultă în realitate întreaga evoluţie a universului. Organizarea reprezintă o formă de manifestare a materiei, un atribut inseparabil al acesteia. Aceasta înseamnă că, pe lângă cel de al doilea principiu al termodinamicii, în univers trebuie să mai existe şi un principiu care să tindă spre creşterea ordinii şi a organizării (Restian, A., 1987, Principiul organizării integrării, complexificării şi diversificării sistemelor, Studii şi cercetării de biotehnologie).

Extinderea acestui principiu la sistemele deschise, adică la lumea reală, a dus la anumite paradoxuri, aşa cum este moartea termică a universului sau paradoxul biologic şi al existenţei sistemelor în general. Cel de al doilea principiu al termodinamicii tinde să crească dezordinea, prin scăderea energiei interne a sistemului, iar principiul ordinii şi al organizării tinde, dimpotrivă, să crească ordinea şi organizarea folosind, cu ajutorul informaţiei, posibilităţile pe care al doilea principiu, care nu apelează la informaţie, nu le exclude cu desăvârşire.

Cu ajutorul informaţiei, principiul ordinii şi organizării, caută să găsească de fiecare dată posibilităţile pe care cel de al doilea principiu, care acţionează neselectiv, nu le exclude cu desăvârşire.

Principiul integrării, complexităţii şi diversificării sistemelor

Acest principiu explică existenţa a mai bine de 500 mii de specii de plante şi mai mult de 1,5 milioane specii de animale.

Integronica ştiinţa proceselor de integrare

Integronica este ştiinţa proceselor de integrare şi a sistemelor hiperintegrate, aşa cum este organismul uman. Ea ţine seama de unitatea indisolubilă a lumii în care trăim şi de necesitatea unei perspective unice asupra acestei lumi. Conceptul este ilustrat în figura 4.

Unitatea: ştiinţă, literatură şi artă, tehnologie, se realizează în cadrul definit de matematică, cibernetică şi filosofie.



Fig. 4 Conceptul de integronică

La baza integronicii se află nu numai unitatea lumii înconjurătoare ci şi unitatea gnoseologică, a cunoaşterii de către subiect a acestei lumi. Pentru că nu se poate vorbi de o cunoaştere fizică, chimică şi nici chiar despre o

cunoaştere ştiinţifică sau artistică, cunoaşterea umană fiind unitară. Integrarea este un proces firesc în natură, care a creat forme şi structuri ce favorizează evoluţia în acest sens.

Procesul de integrare are la bază trei principii: principiul superizării, principiul ordinii şi organizării sistemice şi principiul integrării, complexificării şi diversificării sistemice. În baza principiului superizării, întregul (sistemul) are proprietăţi emergente, datorate efectului de sinergie.

În societatea bazată pe cunoaştere, demersurile pentru promovarea conceptului de integrare în educaţie, cercetare şi tehnologie reprezintă o nevoie majoră. Cunoaşterea în sine este rezultatul structurării şi integrării informaţiilor. Tehnologia informaţiei şi comunicaţiilor facilitează aceste demersuri.

Integrarea reprezintă un principiu de funcţionare a psihicului uman, iar acesta este integrat în sistemul nervos. În literatura de specialitate se aduc în atenţie abordări privind filosofia integrării şi logica integrării, de asemenea se definesc şi mesagerii integrării.

În natură, integrarea poate fi: genetică, prin constrângere, prin dependenţă, la alegere, întâmplătoare etc. Sistemele de integrare pot fi: dominant material-energetice sau dominant funcţional-informaţionale.

În plan socio-economic, trebuie să avem în vedere diferite trepte de integrare precum: integrarea instituţională, integrarea interinstituţională şi integrarea la nivel naţional, ca paşi intermediari pentru integrarea reală în Uniunea Europeană.

În educaţie şi cercetare, integrarea cunoştinţelor şi a resurselor constituie baza pentru a stimula iniţiativa şi creativitatea. Se cunoaşte că, personalitatea unui individ nu depinde atât de lărgimea orizontului şi bogăţia cunoştinţelor, cât de capacitatea de organizare şi integrare a acestora.

Vectorizarea inovării prin stimularea transdisciplinarităţii, integrarea cunoştinţelor şi resurselor în educaţie, cercetare

şi tehnologie constituie baza creşterii productivităţii muncii în producţia de cunoaştere. Mecatronica a deschis orizonturi nebănuite în toate domeniile, datorită stimulării efectului de sinergie.

Studiind legăturile indisolubile dintre diferitele obiecte şi fenomene, integronica încearcă să depăşească limitele extrem de strâmte ale celorlalte ştiinţe particulare, pe care nu le poate înlocui însă. Ştiinţele particulare s-au dezvoltat ca urmare a posibilităţilor limitate ale omului de a cuprinde realităţile lumii înconjurătoare. Nevoia de progres a impus desfiinţarea graniţelor dintre ştiinţe şi evoluţia spre interdisciplinaritate. Aşa au apărut chimia-fizică, biofizica, biochimia etc.

Subliniind limitele abordărilor fragmentare şi necesitatea unei viziuni globale, integronica încearcă să evite astfel de situaţii, subliniind şi mai pregnant faptul că trebuie să se ţină seama nu numai de subsistemul asupra căruia trebuie să acţionăm, ci şi de legăturile lui cu celelalte subsisteme şi, de fapt, de suprasistemul din care ele fac parte. Ea se înscrie astfel în contextul gândirii moderne care este globală, probabilistă, modelatoare, operatoare, pluridisciplinară şi prospectivă.

Concepţia integronică este unul din marile câştiguri ale omenirii datorate revoluţiei informatice. Însăşi principiul de bază al integronicii: principiul ordinii şi organizării sistemice care contravine principiului al doilea al termodinamicii, a putut fi formulat datorită luării în considerare a informaţiei. În formularea principiului al doilea al termodinamicii nu se ţine seama de informaţie.

Extrem de util, acest proces de apariţie a unor ştiinţe interdisciplinare nu s-a dovedit nici el suficient pentru a putea rezolva complicatele probleme ale lumii atât de unitare. Este firesc deoarece, reprezentând mai mult decât suma părţilor sale, unitatea organismului de exemplu, nu poate fi refăcută prin simpla unificare a neurologiei cu endocrinologia sau a psihologiei cu imunologia. Cu atât mai puţin unitatea lumii nu poate fi refăcută prin simpla unificare a astronomiei cu fizica, cu chimia şi biologia.

Prin faptul că informaţia este componentă dătătoare de ton în mecatronică, impactul tehnologiei depăşeşte sfera economicului, fiind esenţială în domeniile social, cultural etc.

Aceasta explică interesul deosebit la nivelul UE şi a ţărilor comunitare de a lansa iniţiative şi a dezvolta programe speciale pentru acest domeniu. Demersurile întăresc convingerea că, în societatea bazată pe cunoaştere, relevanţa culturală depinde de performanţele tehnice, tehnologice.

Bazele concepţiei integronice

Concepţia integronică este unul din marile câştiguri ale omenirii datorate revoluţiei informatice. Însăşi principiul de bază al integronicii: principiul ordinii şi organizării sistemice care contravine principiului al doilea al termodinamicii, a putut fi formulat datorită luării în considerare a informaţiei.

Progresele în domeniul tehnologiei electronice, apariţia circuitelor integrate, mici ca dimensiuni, ieftine şi fiabile, au permis includerea electronicii în structurile mecanice. Se realizează astfel primul pas către integrare: integrarea electromecanică. Structurile electromecanice astfel obţinute nu dispun de inteligenţă proprie.

Următorul pas în integrare a fost determinat de apariţia microprocesoarelor. Cu aceleaşi caracteristici constructive, ca şi circuitele integrate, adică mici ca dimensiuni, ieftine şi fiabile, microprocesoarele au putut fi integrate în structurile electromecanice realizate anterior. Astfel, acestea devin inteligente. Aceasta înseamnă că pot preleva informaţii privind starea internă, starea mediului, pot prelucra aceste informaţii şi pot lua decizii privind comportarea sistemului.

Tehnologia mecatronică aduce în centrul atenţiei problema informaţiei care este componenta dătătoare de ton în raport cu materialul şi energia. Această poziţie a informaţiei este motivată de către japonezi prin următoarele argumente:

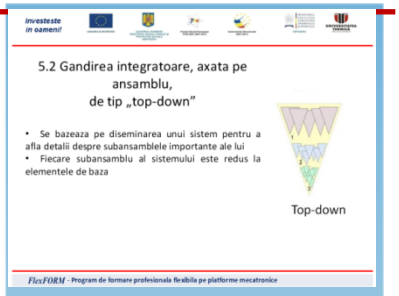
- informaţia asigură satisfacerea nevoilor spirituale ale omului;

- numai informaţia creşte valoarea nou adaugată a tuturor lucrurilor;

- informaţia înseamnă cultură.

Promovarea legăturilor informaţionale în structura sistemelor tehnice le asigură flexibilitate şi reconfigurabilitate. Evaluarea cantitativă şi calitativă a informaţiei constituie o problemă esenţială în educaţie, cercetare şi în activităţile de producţie. Informaţia este deopotrivă importantă în medicină, literatură, artă, sport etc.

Gandirea integratoare, axata pe ansamblu, de tip “top-down”



O abordare de tip “top-down” se bazează pe diseminarea unui sistem pentru a afla detalii despre subsistemele componente.

Într-o abordare „top-down” se defineşte o entitate care să aibă un overview asupra întregului sistem, care specifică dar nu detaliază primele elemente din sistem.

Fiecare subsistem este apoi despărțit în părțile componente, procesul fiind repetat până când se ajunge la elementele de bază.

Există riscul ca această abordare să nu reducă întotdeauna sistemul la elementele de bază sau ca elementele componente să nu fie destul de detaliate astfel încât validarea modelului să nu fie realistică.

Avantajele gandirii de tip “top-down”



Separarea între funcționalitatea elementelor de bază şi funcționalitatea ansamblului duce la un design modular. Utilizând această metodă nu se pierde timp cu detaliile fiecărui subansamblu, până când nu s-a ajuns la elememtele de bază.

Evidențierea integrării modulelor de bază în scheletul ansamblului duce la o bună ierarhizare a întregului sistem, accentuând relațiile între elementele de bază, subansamble şi rolul acestora în ansamblul final. Se poate urmări mai uşor rolul fiecărei componente, astfel managerul de proiect poate să delege persoanele competente pe acel domeniu să se ocupe de acea componentă.

Datorită ierarhizării componentelor într-un schelet se pot urmări efectele fiecărui modul în ansamblul sistemului, astfel încât depanarea unor potențiale erori este mult mai uşoară, cunoscându-se aportul fiecărui modul la sistem.

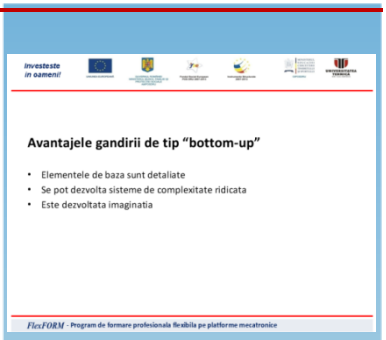
Gândirea axată pe detaliu, de tip „bottom-up”



Gândirea axată pe detaliu, de tip „bottom-up” se bazează pe integrarea unor elemente de bază sau sisteme pentru crearea unui sistem mai complex. În abordarea de tip „bottom-up” elementele de bază sunt dezvoltate până la cel mai mic detaliu. Aceste elemente sunt apoi grupate, pentru a forma un sistem sau un subsistem al unui sistem, formând nivelul superior.

Coordonatorul de proiect trebuie să aibă o intuiție bună pentru a putea decide funcționalitatea elementelor de bază în modulele necesare. Această abordare este eficientă dacă se pleacă de la nişte module deja existente pentru a crea noi sisteme.

Avantajele gândirii de tip „bottom-up”

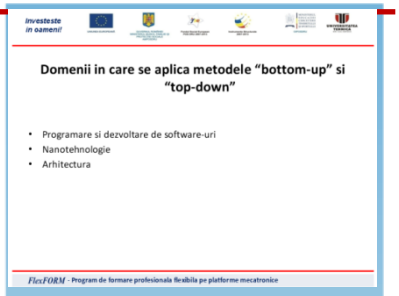


Unul din marile avantaje ale gândirii de tip „bottom-up” este faptul că modulele de bază sunt foarte detaliate, astfel încât se cunoaşte funcționalitatea lor în amănunt.

Utilizând această abordare se pot dezvolta sisteme de o complexitate ridicată, care să deservească mai multe nevoi. Cunoscând funcționalitatea modulelor de bază foarte bine, acestea se pot combina astfel încât eficiența sistemului să fie ridicată.

Gândirea de tip „bottom-up” dezvoltă imaginația, crearea scheletului sistemului poate fi facută utilizând resursele avute la dispoziție, combinate în diferite moduri, pentru a ajunge în final la sistemul dorit.

Domenii în care se aplică metodele “bottom-up” si “top-down”



Unul dintre domeniile în care se aplică uzual aceste tehnici este dezvoltarea de software-uri. Abordarea “top-down” pune accentul pe planificarea şi înțelegerea completă a sistemului. Dezvoltarea programului nu poate începe decât după ce s-a ajuns la un nivel de detaliu ridicat al componentelor sau al subsistemelor. Abordarea „bottom-up” pune accent pe partea de dezvoltare de cod şi pe teste inițiale, care pot fi făcute imediat ce a fost definit un modul de bază. Există totuşi riscul să nu se cunoască rolul acestor module de bază din program, astfel încât legăturile cu celelalte module să se facă mai greu.

Un alt domeniu în care se utilizează această abordare este nanotehnologia. Prin abordarea „top-down” se doreşte să se dezvolte sisteme la scară nanometrică, utilizând alte sisteme la scară mai mare pentru a le crea. Abordarea „bottom-up” foloseşte proprietățile chimice ale moleculelor pentru a le organiza în conformația dorită. Folosind această metodă se pot dezvolta sisteme mai ieftin decât abordarea „top-down” dar ar putea apărea probleme dacă sistemele devin foarte complexe.

În arhitectură, „Şcoala de Arte Frumoase” promovează abordarea de tip „top-down” pentru că se crede că designul arhitectural trebuie să înceapă cu o diagramă de bază, un desen al întregului proiect. Prin contrast, şcoala germană „Bauhaus” s-a axat pe gândirea de tip „bottom-up”, studiind transpunerea sistemelor organizaționale mici în arhitectură, incluzând designul mobilei în modulul de bază în crearea stilului arhitectonic.

Programarea structurii de tip pentalater

Pentru realizarea acestui program este necesară parcurgerea următorilor pași:

1. primul pas constă în realizarea unei interfețe care să permită introducerea poziției dorite pentru efectorul final.
2. următorul pas este implementarea unui algoritm care, folosind informațiile de intrare (poziția dorită pentru efectorul final în coordonate carteziene, măsurate în unități lego) să determine unghiul la care trebuie să se afle cele două motoare.
3. de asemenea mai trebuie realizat un algoritm de calibrare pentru a determina poziția absolută a motoarelor.
4. si ultimul pas, realizarea unui algoritm care să țină motoarele în poziția dorită.

Se determină modelul matematic după care va fi creat algoritmul.

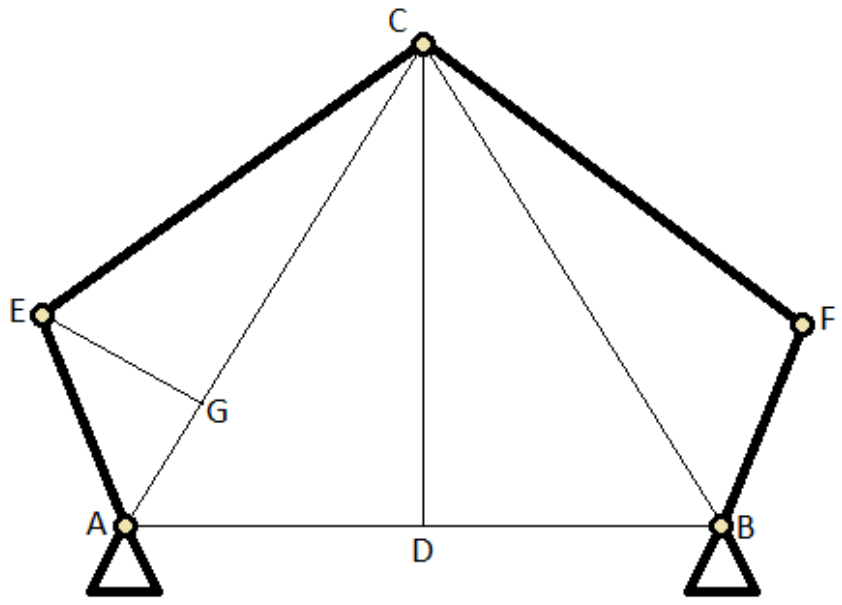


Fig.1

În fig.1 se poate observa schema mecanismului. În această schemă cunoaștem dimensiunile segmentelor: AE=BF=7, AB=CE=CF=14, AD=x și DC=y. x și y se cunosc de la pasul anterior. Scopul este de a determina unghiurile EAD și DBF. Pentru început se va calcula unghiul EAD principiul de calcul fiind același și pentru unghiul DBF.

În prima fază se va determina dimensiunea segmentului AC, care se calculează cu formula 1(Pitagora).

$$AC = \sqrt{AD^2 + CD^2}\ . \quad (1)$$

După care se determină unghiul CAD cu formula 2.

$$CAD = \tan^{-1}\frac{CD}{AD} \quad (2)$$

În următoarea fază se va determina lungimea segmentului EG cu formula 3(Heron)

$$EG = \frac{1}{2AC}\sqrt{(-AC + EC - EA)(-AC - EC + EA)(-AC + EC + EA)(AC + EC + EA)} \quad (3)$$

urmând să se calculeze segmentul CG cu formula 4 și segmentul GA cu formula 5.

$$CG = \sqrt{EC^2 - EG^2} \quad (4)$$

$$GA = AC - GC \quad (5)$$

Cunoscând GA și EG se poate determina unghiul EAG folosind formula 6.

$$EAG = \tan^{-1}\frac{EG}{AG} \quad (6)$$

Ultima etapă în constă în însumarea unghiurilor EAG și CAD pentru determinarea unghiului EAD.

Implementarea algoritmului

Pentru calculul inversei tangentei din formula 2 se va folosi blocul ”atan2”. Acest bloc necesită ca pe prima intrare să-i fie dată cateta alăturată și pe a doua cateta opusă, iar pe ieșire calculează unghiul ținând cont de cadranul în care se află ipotenuza conform semnelor. Folosind aceleași intrări se calculează mai departe şi formula 1, adică dimensiunea ipotenuzei, folosind un bloc ”e=mc^2” pentru însumarea pătratelor catetelor și ”Math” pentru extragerea radicalului. Cele trei blocuri vor fi selectate și se va folos opțiunea ”Make A New My Block” din meniul ”Edit” pentru a crea un singur bloc . În figura 2 se poate observa cum trebuie să arate blocul.

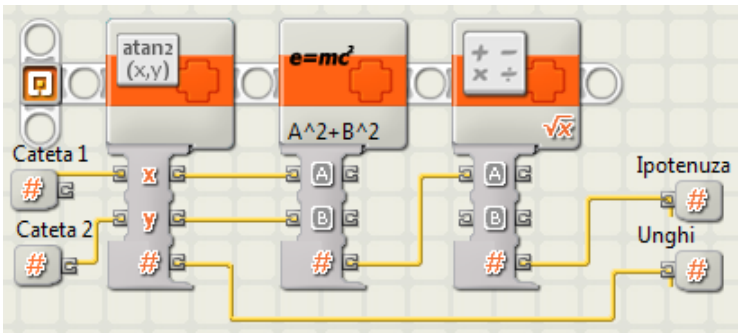


Fig. 2

În figura 3 se poate vedea cum arată ecuația 3. Pentru scrierea acesteia s-au folosit mai multe blocuri începând cu blocul ”e=mc^2”. În acesta a fost introdusă partea de sub radical a ecuației. Laturile AC, AE și EC au fost înlocuite cu caracterele A, B și respectiv C, iar laturile AE şi EC, fiind constante, au fost introduse direct in bloc. A este ipotenuza de la ecuația anterioară. Cu următoarele blocuri se extrage radicalul, iar rezultatul se împarte la AC și la 2.

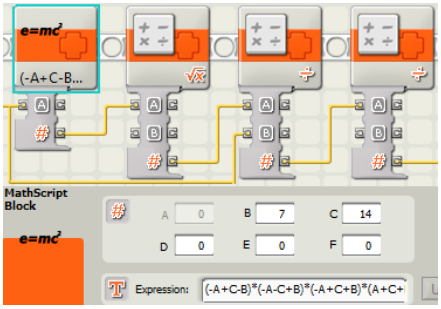


Fig. 3

Pentru calculul formulei 4 se va folosi o metodă asemănătoare cu cea din cazul formulei 1(un bloc ”e=mc^2” și unul ”Math” pentru extragerea radicalului). Formula 5 se calculează cu un bloc ”math”, iar formula 6 se calculează folosind blocul ”atan2” care a fost folosit și pentru calculul formulei 2. Urmează însumarea unghiurilor. În figura 4 se poate vedea cum trebuie să arate cele patru formule implementate în cadrul software-ului NXT Mindstorm.

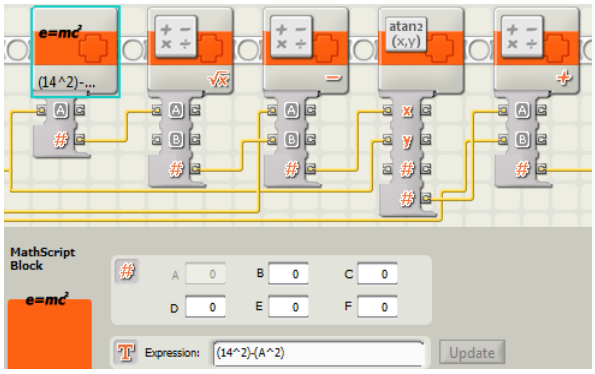


Fig. 4

La sfârșit se va selecta întregul algoritm și se va folosi opțiunea ”Make A New My Block” pentru a crea un singur bloc care va fi denumit ”poziție motor”.



Fig. 5

Următorul pas este crearea algoritmului de calibrare (Fig.6) . Acesta va determina poziția absolută a motoarelor folosind un senzor de contact. Procedura constă în mișcarea fiecărui motor pe rând până la atingerea senzorului și setarea poziției pe care motorul o are la momentul contactului ca fiind poziția de zero.

Pentru a realiza acest lucru, în interiorul unui ”Loop” setat să se întrerupă când primește semnal pe senzorul tactil, se va pune un bloc de tip ”Motor”. Acesta va fi setat să miște motorul cu putere redusă spre senzor. În continuare se va pune un bloc ”Rotație Sensor” setat să reseteze senzorul la zero și unul ”Motor” setat să miște motorul 90 de grade în direcția opusă. Procedura se va repeta și pentru celălalt motor.

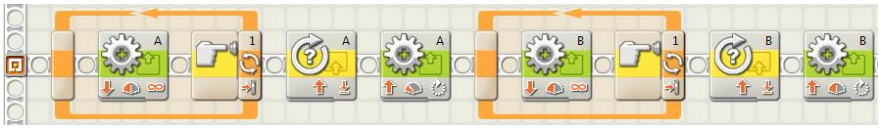


Fig. 6

Ultimul pas presupune folosirea algoritmului de la pasul doi și menținerea motoarelor la poziția specificată de acesta. Pentru a folosi același algoritm pentru ambele motoare, este nevoie ca originea sistemului de coordonate să fie modificată individual pentru fiecare motor în așa fel încât originea să fie în punctul de acționare a motorului și direcția axei x să fie îndreptată spre mijlocul mecanismului. În figura 7 se poate vedea partea de citire a pozițiilor folosind blocul ”Variable”. Modificarea lor este făcută cu blocul ”Math”. După calculul făcut cu blocul ”poziție motor” rezultatele sunt trimise mai departe în variabilele alfa și beta care reprezintă poziția motorelor.

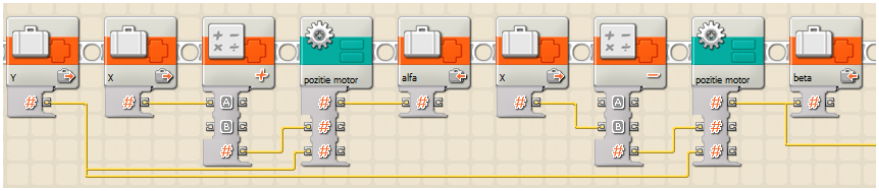


Fig.7

În figura 8 se poate vedea controlul proporțional pentru poziția unui motor.

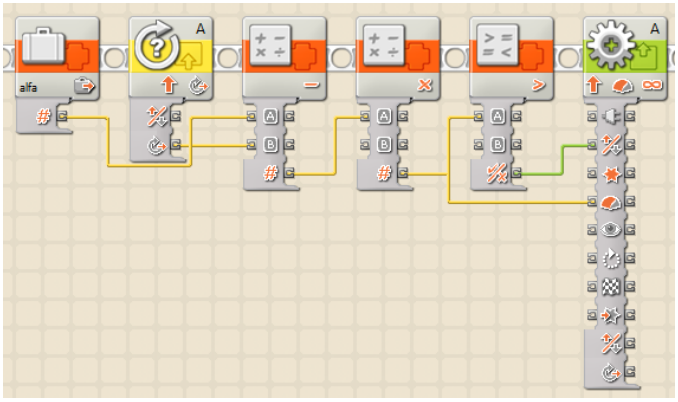


Fig.8

Ultimul pas al algoritmului de calibrare este introdus în totalitate într-un ”loop” infinit.