

Conținut

Introducere în Robotică	2
Modalități de programare a roboților	7
Instrucțiuni metodice și practice în ajutor instructorilor în organizarea activităților de Robotică	12
Sisteme robotice	17
Obiective de învățare:	19
Începător, intermediar, avansat	19
Scurtă descriere a activității (Rezumat)	26
Centrul de resurse – ghiduri și instrucțiuni	27
Procurarea roboților LEGO MINDSTORMS EV3	27
Masa pentru misiuni	27
Concursul FIRST LEGO LEAGUE	28
Misiunile pentru concursul FIRST LEGO LEAGUE	29
Alte concursuri pentru cluburile de robotică	29
Biblioteci echipate cu Robotică	29
Contacte utile	30

Introducere în Robotică

Termenii de bază utilizați în cadrul roboticii

În acest capitol, vom analiza termenii „robot”, „robotică”, „mecatronică”, „senzor”, „actuator” și „controler”. În plus, la finalizarea acestui capitol, veți putea determina din care generație aparține un robot (prima, a doua sau a treia generație) și veți putea explica relațiile dintre oameni și roboți, conform legilor lui Asimov.

În decursul a mai multor secole, oamenii au încercat să creeze mașini care le-ar fi ușurat munca. În secolul trecut au fost create mai multe mașini ce au ușurat munca fizică a oamenilor, de ex., excavatoare pentru săpături, macarale pentru ridicarea materialelor grele, diverse mijloace de transport pentru a se deplasa rapid, cum ar fi trenuri, automobile, avioane etc. În prezent, datorită dezvoltării mașinilor inteligente, ar fi posibil de a scuti complet oamenii de munca de rutină. Acest lucru le-ar permite oamenilor să aibă mai mult timp pentru activități creative, intelectuale, cum ar fi crearea roboților. În acest manual sunt explicate principiile ce stau la baza funcționării roboților din jurul Dvs. Acești roboți nu urmează mișcările prescrise, dar sunt capabili să ia decizii limitate.

În cele ce urmează, vom descrie ce este un robot, ce înseamnă termenul de robotică și cum putem clasifica diferiți roboți.

Robot

Un robot este un dispozitiv mecanic inteligent, ce poate rezolva probleme fie el însuși, fie ajutat de oameni. În practică, roboții sunt, de obicei, niște dispozitive electromecanice, conduse de un calculator. Dintr-un alt punct de vedere, toți roboții sunt împărțiți în două clase:

✓ *Roboți inspirați din umanoizi și alte creaturi biologice.* Acești roboți seamănă cu oamenii sau alte creaturi (de ex., ASIMO produs în anul 2000 de Honda, roboți umanoizi, roboți insecte, după cum se arată în Figura 1.1, roboți fermieri etc.).

✓ *Roboți-asistenți,* utilizați pentru a îndeplini sarcini prea monotone, periculoase sau dificile pentru oameni, de ex., roboți industriali, roboți militari sau de deservire, roboți mobili etc.

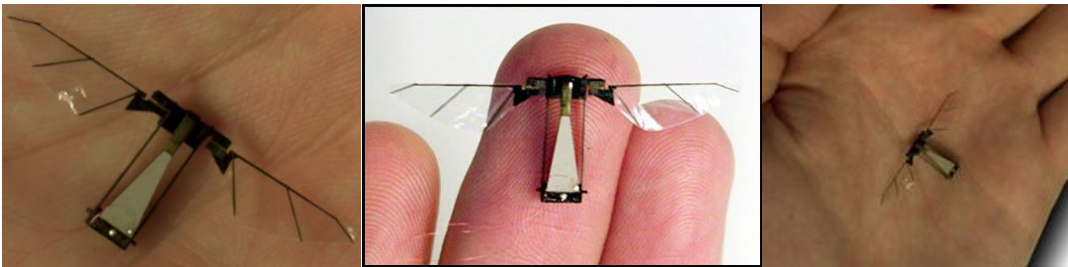


Figura 1.1. - Robot insectă <http://photo-junction.blogspot.com/2010/03/robot-insects-photos.html>

Toți roboții pot lua decizii și pot face alegeri inteligente. În unele situații mai dificile, roboții pot comunica cu oamenii și pot solicita informații suplimentare. Unii roboți sunt capabili să învețe, de ex., să acumuleze experiență și să-și schimbe comportamentul, într-o anumită limită, fără ajutor extern. Toate algoritmele, ce stau la baza deciziilor robotului, sunt programate de către creatorii robotului. Roboții se comportă conform algoritmului și toată inteligența lor depinde de creatori.

De exemplu, un automobil dispune de sistemul airbag, ce se bazează pe senzorul de accelerație. Dacă motorul automobilului funcționează, un calculator mic are funcția de a compara datele privind accelerarea într-un sistem de gestionare a airbag-ului. În cazul în care automobilul pierde din viteză mai rapid decât ar fi posibil la frânare, înseamnă că s-a produs un accident și calculatorul trimite semnalul pentru a deschide airbag-urile. Aceasta este o descriere simplificată, deoarece airbag-urile moderne au o mulțime de alți senzori. Calculatorul ia în considerare prezența unui scaun de siguranță, greutatea călătorului și alți factori importanți. Întregul sistem funcționează în mod automat și nu necesită nicio intervenție umană, astfel încât poate fi numit sistem autonom. Deoarece pentru a fi pus în mișcare este nevoie de un șofer, automobilul nu poate fi numit robot; cu toate acestea, în interiorul automobilului, pe lângă subsistemul de airbag, există o mulțime de alte subsisteme robotizate.

Robotica

Robotica este un domeniu al tehnologiilor și științelor tehnice, ce studiază cum se construiesc roboții. În acest scop, robotica dezvoltă tehnologii și științe aplicate: proiectarea, construcția, dezvoltarea algoritmilor, producerea roboților și a aplicațiilor pentru roboți. Robotica este strâns legată de electronică, științe tehnice, mecanică, mecatronică și elaborarea de software.

Nu este absolut neapărat ca roboții să aibă o formă umană. Acest clișeu este asociat cu filmele științifico-fantastice. Orice robot trebuie să aibă trei componente: senzori, ce primesc semnale din mediul înconjurător, un dispozitiv de gestionare (adică, un „creier”), ce analizează semnalele primite și ia decizii, și un actuator, ce execută comenzile „creierului”. Se știe că o mașină de spălat rufe este un robot, deoarece are senzori, un „creier” și actuatori. Astfel, într-adevăr, o putem numi robot, deoarece este un dispozitiv independent, ce nu necesită intervenția omului în timpul ciclului de spălare. Există mai mulți senzori în interiorul mașinii de spălat, dar să analizăm mai atent modul în care aceasta încălzește apa. Mașina de spălat primește un semnal de la senzorul de temperatură cu privire la temperatura apei și, dacă apa nu este suficient de caldă, „creierul” dă comandă încălzitorului să continue încălzirea.

Un al doilea robot larg răspândit este detectorul de fum. În unele țări, prezența acestui senzor este obligatorie în fiecare casă și apartament. Senzorul de fum este un dispozitiv relativ simplu. În interiorul dispozitivului există o cameră mică cu două plăci paralele. Dacă particulele de fum pătrund în această cameră, rezistența electrică dintre cele două plăci se schimbă și se declanșează un semnal ce indică prezența fumului în încăperea. Pentru a informa oamenii, se pornește alarma. Există și alte tipuri de detectoare de fum, unul din acestea este prezentat în următoarea secvență video.

<https://www.youtube.com/watch?v=pcn7MDpoZrY>

Dat fiind faptul, că roboții au fost creați pentru a ușura munca oamenilor, ar trebui să fie aplicate o serie de reguli în relația dintre oameni și roboți, ce ar împiedica roboții să execute anumite comenzi. Autorul a numeroase cărți științifico-

fantastice, Isaac Asimov, a fost primul care a formulat aceste reguli în povestirea „*Fuga în cerc*” încă în 1942. Roboții buni îndeplinesc următoarele reguli:

1. *Un robot nu poate face rău unui om prin acțiunile sau inacțiunile sale.*

2. *Un robot trebuie să urmeze comenzile omului, cu condiția ca aceste comenzi nu se află în contradicție cu prima regulă.*

3. *Un robot trebuie să-și apere existența, cu condiția că nu se încalcă prima și a doua regulă.*

Subiect de discuție: Dacă ar trebui să formulați reguli pentru roboți, ce fel de reguli ați mai adăuga?

Chiar dacă în literatura științifico-fantastică deja de mai mult timp se vorbește despre roboți și sunt abordate aspecte legate de roboți, în realitate, tehnologiile robotizate au apărut abia în ultimii 60 de ani.

În funcție de nivelul de dezvoltare, roboții pot fi împărțiți în trei generații:

✓ *Prima generație.* Roboții de prima generație au o construcție simplă și nu au senzori (adică, o înțelegere a mediului înconjurător). Aceștia își îndeplineau munca în condiții constante. De exemplu, un robot de prima generație ar fi o mână robotizată, ce plasează sticlele în cutii la o fabrică de bere. Robotul efectuează aceleași mișcări, dar, deoarece nu primește semnale din mediul înconjurător, nu are absolut nici un feedback asupra acțiunilor sale. Prin urmare, nu va mai putea funcționa corect în cazul când se vor produce schimbări (nu mai sunt sticle sau cutiile sunt de dimensiuni diferite). Acești tipuri de roboți nu mai sunt numiți roboți reali.

✓ *A doua generație.* Roboții, aparținând celei de a doua generații, datorită senzorilor, pot reacționa la schimbările din mediul în care se află. În acest caz, eficiența robotului depinde de software-ul său. Toți roboții utilizați în acest curs aparțin celei de-a doua generații, întrucât folosim senzori și îi programăm să acționeze independent.

✓ *A treia generație.* Această generație se află în curs de dezvoltare. Roboții din a treia generație pot analiza situațiile și pot învăța din propria experiență. Spre exemplu, robotul din filmul „Eu, robotul”, este un robot din a treia generație și este capabil să tragă concluzii și să învețe.

Robotul umanoid Nao, ce aparține celei de-a doua generații. Nao are o înălțime de 58 cm și cântărește 4,3 kg. Este programabil. A fost posibil să obținem Nao pentru activități științifice și ca instrument de învățare din 2008.

https://static.generationrobots.com/5474-deals_default/programmable-humanoid-nao-evolution-robotred.jpg

Manipulatoare

Un manipulator este un dispozitiv sau o mașină condusă de om. Contrar ideilor generale, manipulatoarele nu sunt roboți, deoarece sunt dirijate în mod direct de către om. Manipulatoarele nu pot lucra independent. Manipulatorul este condus de informația provenită din mintea umană, aceasta este analizată și operatorul uman ia deciziile corespunzătoare. În consecință, manipulatoarele sunt doar actuatori. Manipulatoarele pe larg cunoscute sunt, de exemplu, automobilele pentru copii conduse prin telecomandă și alte jucării ghidate prin telecomandă.

Senzori, microcontrolere și actuatori

Un *senzor* este un dispozitiv ce măsoară un parametru și îl transformă într-un semnal electric. Vorbind simplu, senzorii sunt organele de simț ale unui robot, cum ar fi ochii, nasul, urechile la om etc. În rețelele neurale umane, semnalul este transmis sub formă de impulsuri electrice. Diferitele componente ale unui robot la fel comunică prin intermediul semnalelor electrice. Robotul primește un semnal electric de la senzor și se comportă în funcție de acest semnal. Cu cât mai mulți senzori folosește un robot, cu atât mai bine poate fi modelat comportamentul acestuia și cu atât mai multe sarcini îi putem încredința. Pentru a gestiona informațiile primite de la senzori, robotul are nevoie de un controler.

Controlerul robotului este adesea numit microcontroler, deoarece este, într-adevăr, de dimensiuni foarte mici. Controlerul este „creierul” robotului. Semnalul de la senzor ajunge la „inputul” microcontrolerului, unde este analizat, iar mai apoi, în dependență de informația analizată, robotul face următoarele mișcări. Memoria controlerului conține un algoritm, în baza căruia robotul ia deciziile. Software-ul pentru microcontroler este scris în funcție de sarcina pe care trebuie să o îndeplinească robotul. Prin urmare, dacă am conecta o cameră video la robot, ar trebui să scriem un software pentru recunoașterea obiectelor din câmpul vizual al camerei video. În baza semnalului primit de la senzor, microprocesorul pornește actuatorul. Acesta poate juca un rol specific în gestionarea oricărui dispozitiv, ce poate fi manipulat, de ex., motoare, dispozitive de încălzire etc.

Pentru un robot, *actuatoarii* sunt ca mâinile și picioarele pentru om. Scopul actuatorilor este de a influența mediul înconjurător. De exemplu, un robot ce controlează încălzirea, pornește sistemul de încălzire doar dacă temperatura scade sub pragul critic setat și o oprește la atingerea temperaturii stabilite.

Domeniul de utilizare a roboților și propagarea roboticii

Deoarece roboții sunt dispozitive mecanice, acestea pot lucra încontinuu, ceea ce le face ideale pentru activitatea de rutină. Iată de ce, în multe întreprinderi industriale, roboții sunt folosiți pentru lucrul de rutină. În plus, roboții nu sunt la fel de sensibili la condițiile de muncă ca și oamenii. Ei pot fi folosiți în locuri zgomotoase, slab iluminate sau chiar în locuri periculoase pentru om, în condiții de radiație, de exemplu, etc. Roboții pot fi de asemenea, utilizați în locuri unde nu pot fi trimiși oamenii – la fundul unui ocean, pe Marte etc. Roboții sunt, de asemenea, folosiți pentru distracție, de exemplu, jucăriile inteligente, roboții dansatori etc.

Pentru a promova robotica sunt organizate diferite concursuri și activități. Cele mai populare dintre acestea sunt Robotex, FIRST® LEGO® League și Jr. FIRST® LEGO® League. Toți cei interesați de robotică sunt bineveniți.

O prezentare generală a mecatronicii

Mecatronica este o ramură complexă a științelor tehnice, ce combină mecanica, electronica, tehnologia informațională, tehnologia software, teoria controlului și proiectarea sistemelor, cu scopul de a crea un produs final mai

bun. Termenul „**mecatronică**” este o combinație dintre cuvintele „**mecanică**” și „**electronică**”, mecanică pentru construirea dispozitivelor și electronică pentru gestionarea acestora. Cuvântul este de origine japoneză.

Mecatronica are mai multe domenii de aplicare:

- Tehnologia viziune mașină sau recunoașterea obiectelor dintr-o imagine, identificând mișcărilor și caracteristicile lor.
- Robotica – proiectarea roboților și identificarea aplicațiilor pentru aceștia.
- Dispozitive automate sau de construcție pentru a reduce necesitatea lucrătorilor umani.
- Sisteme de senzori sau de dispozitive de construcție ce colectează, prelucrează și gestionează informațiile prin intermediul senzorilor.
- Sisteme expert într-un domeniu specific, unde sistemul informatic încearcă să imite procesul de luare a deciziilor la nivel de specialist și să ajute specialiștii umani la luarea deciziilor (de ex., un sistem expert pentru diagnosticarea diabetului).
- Industria de automobile, unde roboții sunt utilizați în linia de producție, de ex., pentru sudarea sau vopsirea autoturismelor.
- Aplicații în electronică și în domenii similare, unde roboți bine programați fixează componente pe plăci cu circuite imprimate mult mai eficient ca oamenii, etc.

În contextul acestui manual, robotica este una dintre direcțiile de dezvoltare și de aplicare ale mecatronicii.

Există și un al treilea termen referitor la robotică și mecatronică – automatică. Automatică este definită drept un domeniu al tehnologiei ce utilizează teoria controlului, tehnologia informațională și a comunicației pentru a reduce necesitatea forței de muncă umane în procesul de producție. Cuvântul „automatică” provine din cultura anglo-americană.

Un robot este orice sistem ce are un senzor, un controler și un actuator. Senzorul oferă controlerului informații despre mediul înconjurător. Controlerul le procesează și verifică acțiunile dispozitivului, determinând actuatorul să-și îndeplinească sarcinile: să miște ceva, să schimbe ceva etc.

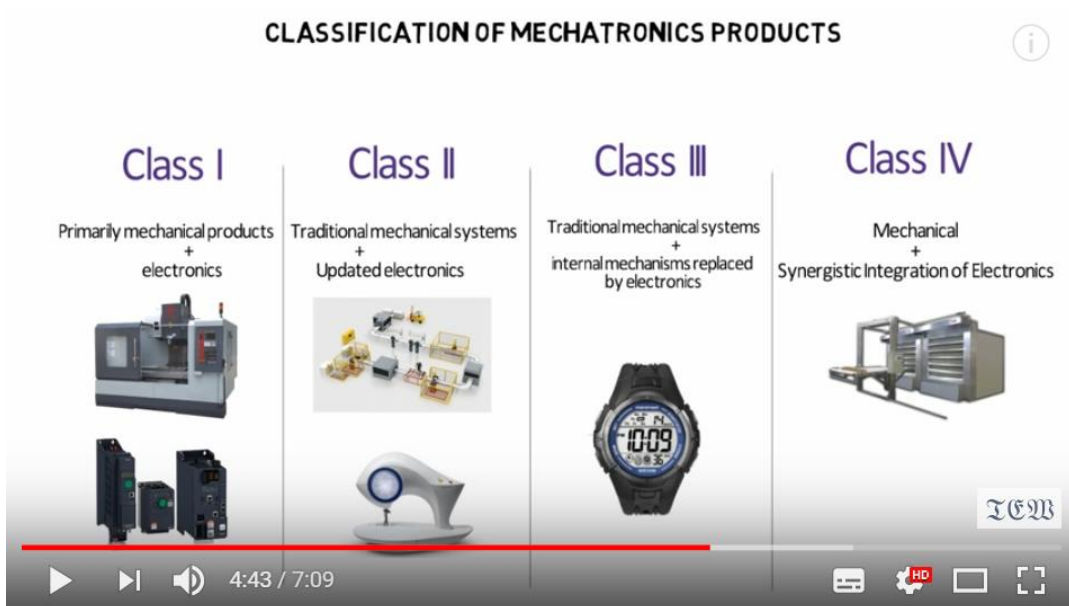


Figura 1.2 - Video pentru a demonstra mecatronica. https://www.youtube.com/watch?v=Ro_tFv1iH6g

Rezumat. Robotica, mecatronica și automatica sunt termeni ce provin din medii istorice și culturale diferite. Activitățile în toate aceste domenii sunt similare, însă obiectivele acestora diferă.

Întrebări pentru revizuire:

1. Formulați legile fundamentale ale roboticii.
2. Clasificați următoarele generații de roboți: o mașină de spălat, o mână robotică cu mișcări fixe, o lampă cu comutare automată, un robot capabil să învețe un limbaj în mod independent.
3. De ce manipulatoarele nu sunt roboți?
4. Care sunt avantajele folosirii mașinilor în comparație cu forța de muncă umană?
5. Care sunt cele trei condiții ce trebuie îndeplinite pentru ca un dispozitiv să fi numit robot?
6. În ce întreprindere industrială își va putea găsi un loc de muncă o persoană care a studiat mecatronica (în Moldova, în Europa)?
7. Ce fel de abilități puteți obține prin învățarea mecatronicii?

Modalități de programare a roboților

În acest capitol, vom analiza diferite moduri de programare, precum și diferențele dintre programarea grafică și textuală. În plus, veți înțelege ce este un algoritm și care este legătura dintre algoritmi și programare.

Într-un robot identificăm următoarele componente: un controler, senzori, actuatori, iar în memoria controlerului vom găsi un program ce controlează întregul sistem. În acest capitol, vom explica necesitatea programelor și ce instrumente pot fi folosite pentru a le crea. Ceea ce se întâmplă într-un controler poate fi comparat cu ceea ce se întâmplă în creierul uman. Creierul generează deciziile de a mișca mâinile sau picioarele. De exemplu, dacă am descoperit, în baza unui semnal primit de la ochi, că într-o încăpere este prea întuneric pentru a citi, vom hotărî să

aprimem lumina. Un robot primește de la senzori informații despre mediul înconjurător, iar un program manipulează controlerul, adică ia decizia de a porni sau opri motoarele, analizează datele primite de la senzori și generează comportamentul robotului în funcție de deciziile sale. Procesorul din controler se comportă în funcție de program și, ca rezultat, robotul se comportă după cum dictează programul. După cum s-a menționat anterior, inteligența robotului depinde de programul ce-l controlează.

În timp ce robotul funcționează, programul său se află în memoria controlerului. De unde apare acest program acolo și cum sunt create aceste programe? De regulă, programele sunt create la un calculator sau laptop obișnuit, folosind software special. De exemplu, putem folosi în acest scop software-ul LEGO NXT-G (sau Educational) sau limbajul de programare standardizat C și mediul acestuia. Programele create în acest mod sunt încărcate în memoria controlerului robotului printr-un cablu special sau prin folosirea unor dispozitive fără fir (de ex., Bluetooth). Controlerul robotului are, de obicei, un ecran mic și câteva butoane. Prin utilizarea acestora, pot fi create programe mici chiar și fără un calculator / laptop.

Există o mulțime de instrumente pentru crearea programelor. Mijloacele, pe care le folosim pentru a scrie comenzi pentru un robot, le numim limbaje de programare. Împreună cu un nou limbaj de programare, este creat și mediul în care putem scrie programe. Limbajele de programare au diferite clasificări. Aici vom aborda două categorii: *grafice și bazate pe text*.

Limbajul de programare grafică este potrivit pentru începători. Pentru a crea un program, componentele dorite se trag în câmpul programului, se conectează în mod adecvat și li se prescriu parametrii necesari. Un exemplu de limbaj de programare grafică ar fi LEGO NXT-G sau altul de o versiune mai recentă LEGO MINDSTORMS Education EV3.

Există multe limbaje de programare bazate pe text. În robotică, unul dintre cele mai folosite este limbajul C. Roboții HomeLab pot fi programați în acest limbaj de programare. Pentru a programa în orice limbaj de programare bazat pe text, este necesar ca acesta să fie învățat cu minuțiozitate, deoarece este foarte important de a cunoaște sintaxa limbajului. Dacă sintaxa va fi greșită, programul nu va funcționa.

Programarea grafică

Programarea grafică este cel mai simplu mod de a vă familiariza cu programarea. În cazul unui limbaj de programare grafică, programatorul nu are nevoie să cunoască structura și sintaxa exactă a limbajului de programare. Programarea pornește de la o percepere clară a sarcinilor pe care dorim să le efectueze robotul. Să presupunem că dorim ca robotul să se deplaseze înainte cu un metru, să întoarcă la stânga și să mai parcurgă un metru. Într-un limbaj grafic acest lucru ar fi ușor de programat, deoarece toate cele trei comenzi (deplasează-te, întoarce la stânga, deplasează-te) pot fi scrise, folosind doar trei blocuri. „Deplasarea înainte” cu un metru este descrisă printr-un bloc, „întoarce la stânga” cu un al doilea bloc, iar pentru „deplasare înainte” cu un metru putem folosi al treilea bloc. Avantajul unui limbaj grafic este că poate fi relativ ușor învățat și folosit. Programarea grafică este utilă pentru scrierea programelor, mai ales,

În cazurile când un robot trebuie să repete aceleași activități. În figura de mai jos este prezentat un exemplu de programare grafică în limbajul de programare LEGO® MINDSTORMS® Education EV3.



Figura 1.2 - Exemplu de program într-un limbaj grafic, care face robotul să se miște într-un pătrat

Dacă dorim ca robotul să se miște într-un pătrat, vom lăsa robotul să meargă înainte pe o distanță înainte (care constituie latura pătratului), apoi să se rotească la 90 de grade, să meargă din nou înainte și să repete acest lucru de 4 ori.

Pentru a obține abilitatea de a programa într-un limbaj grafic, este nevoie de câteva ore, în unele cazuri, chiar doar de câteva minute. Mediul grafic este convenabil pentru programe mai simple. În cazul programelor complexe, s-ar putea să pierdem imaginea de ansamblu a programului, deoarece blocurile sunt relativ mari. În plus, un program prea mare nu se va încadra în memoria calculatorului în această formă, deoarece programarea în blocuri ocupă un volum mare de memorie. Cu toate acestea, în formă grafică, chiar și începătorii pot citi și înțelege programele, acestea fiind cuprinzătoare. Acest tip de programare va fi analizat mai îndeaproape în sarcinile practice ce vor urma mai departe.

Programarea bazată pe text

Pentru a programa controlerul unui robot sunt folosite aceleași limbaje de programare bazate pe text ca și în cazul programării unui calculator obișnuit. În plus, pentru a programa roboți, folosim și un asamblator, chiar dacă în cazul calculatoarelor obișnuite, un asamblator este utilizat doar în cazuri speciale. Limbajele de programare utilizate pe scară largă pentru programarea controlerelor sunt C și C++ împreună cu biblioteci speciale pentru hardware-ul robotului. În plus, mai pot fi folosite limbajele Python, Basic și altele, deși acestea necesită existența unui software compilator pentru a traduce programul scris în codul intern al controlerului. De obicei, aceste compilatoare nu sunt disponibile în mod liber. Un compilator disponibil gratuit pentru familia de controlere de tip AVR este avr-gcc, ce traduce din limbajul standard C. Acest lucru permite scrierea programelor de gestionare a controlerului în limbajul C și, prin utilizarea unui compilator gratuit, traducerea acestora într-un cod automat pe înțelesul controlerului. Pe lângă limbajele de programare standard, se folosesc și așa-numitele pseudo limbaje de programare. Aceste limbaje sunt compuse pentru o anumită platformă, iar scopul general al acestora este simplificarea programării. Un bun exemplu al acestui tip de limbaj, folosit de către

amatorii de robotică, este un limbaj asemănător cu limbajul C pentru platforma Arduino, ce utilizează biblioteci și construcții dependente de platformă. Avantajele utilizării acestui tip de limbaj constau în simplitatea relativă a acestuia și compatibilitatea cu platforma. Dar există și neajunsuri. În zilele noastre, se folosesc instrumente și limbaje de programare standardizate pentru a permite transferul soluțiilor create pe alte platforme.

Mai jos avem un exemplu de program scris în C. Textul după backslash-uri nu este decât un comentariu pentru om, compilatorul îl va omite și nu-l va traduce.

```
// Macro-limbaj
#include <stdio.h>
// Declarații
int i;
// Funcții
int main(void)
{
// Propoziții
printf("Hello word!\n");
//expresii
i = 32 + 5 * 4 / (23 % 5);
}
```

În limbajele de programare, construcțiile sunt toate la fel și nu depind de limbajul ales. Cele mai importante părți ale construcțiilor sunt expresiile, clauzele condiționale și ciclurile. În construcții pot fi folosite diferite cuvinte și semne-cheie în diferite limbaje, în timp ce logica acțiunii este întotdeauna aceeași indiferent de limbaj. În consecință, nu contează în ce limbaj este scris programul și, atunci când cunoști bine un limbaj de programare, va fi ușor de învățat un alt limbaj de programare.

Bibliotecile unui limbaj de programare bazat pe text

Principala diferență dintre programarea unui microcontroler și a unui calculator este conexiunea directă cu diferite tipuri de hardware. Există diverse componente hardware – senzori, dispozitive de transmitere a datelor, dispozitive de gestionare a motoarelor, interfețe pentru schimbul de date. Software-ul pentru gestionarea acestor componente hardware poate fi destul de complicat. De exemplu, afișarea unui simbol pe ecranul LCD necesită mai multe operații, începând de la inițierea controlerelor și a interfeței de date a ecranului LCD, setarea registrelor ecranului LCD etc. Pentru a simplifica aceste procese, producătorii de hardware și o serie de entuziaști au creat diferite biblioteci software. Aceste biblioteci reprezintă o colecție de funcții software ce simplifică utilizarea și gestionarea hardware-ului. Ele permit folosirea funcțiilor gata făcute și a unei linii de cod pentru a face operații ce ar necesita o pagină întreagă de coduri.

Aceste funcții sunt, de obicei, colectate în diferite biblioteci fie de un dispozitiv sau un grup de dispozitive. Pentru a utiliza aceste funcții în programul C, este necesar de adăugat biblioteca corespunzătoare (de exemplu, `andurid.h`) în textul C prin antetul programului. Un alt obiectiv al bibliotecii este de a facilita citirea textului programului. O practică obișnuită în cazul controlerelor este de a gestiona pinii de ieșire ai controlerului, folosind registre a căror valoare poate fi binară sau hexazecimală, folosind operații logice. De obicei, aceasta este partea cea mai dificilă pentru un programator începător în domeniul controlerelor. Folosind bibliotecile, este posibil de a depăși această operație dificilă în interiorul bibliotecii și de a utiliza funcții cu denumiri prietenoase utilizatorului. În exemplul următor, săgețile sunt folosite pentru a demonstra modul în care manipularea registrelor (partea inferioară) este înlocuită cu o funcție prietenoasă utilizatorului.

```
#include <homelab/pin.h>
//
//Assigning chip output pin to LED
//
pin_debug_led = PIN(B, 7);

//
// Main program
//
int main(void)
{
    //set LED pin to output
    pin_setup_output(debug_led);
    //Turning LED on
    pin_clear(debug_led);
}

#include <avr/io.h>
//
// Main program
//
int main(void)
{
    //set LED pin to output
    DDRB|=(1<<7);
    //Turning LED on
    PORTB&=~(1<<7);
}
```

Figura 1.3 - Exemplu de program bazat pe text cu utilizarea bibliotecii.

Algoritmi ce utilizează programarea bazată pe text

Un algoritm este o instrucțiune de funcționare pas cu pas sau un model de program. Judecata robotului depinde direct de programul său; robotul nu este capabil să gândească de sine stătător. Prin urmare, este necesar să acordăm o atenție sporită algoritmului pentru a face robotul să se comporte cât mai înțelept posibil. Atunci când este vorba de aplicații mai complexe, nu este cazul să ne grăbim să scriem imediat codul programului, ci să concepem și să modelăm în prealabil întregul software. Cea mai ușoară modalitate este de a folosi un algoritm grafic ce descrie comportamentul robotului și răspunde la diverse semnale de intrare. Ulterior, va fi simplu de a deriva codul software-ului, bazându-ne pe modelul grafic. Timpul folosit pentru a modela comportamentul robotului și de a-l conforma codului este recuperat mai târziu prin faptul că alți oameni vor putea înțelege mai bine ce anume a avut în vedere programatorul inițial. La fel, este mai ușor de depistat greșelile în logica comportamentului prin analiza algoritmului grafic în comparație cu depistarea acestora în codul programului.

Modele de programe simple pot fi desenate folosindu-se scheme, unde un dreptunghi descrie o operație sau o expresie, un romb descrie condițiile, iar săgețile indică desfășurarea procesului. Programele simple nu au, de obicei, sfârșit, întrucât întregul proces este în ciclu fără sfârșit, ceea ce este rar admisibil pentru programele informatice obișnuite. De obicei, robotul este programat pentru o singură misiune și funcționarea acestuia este oprită chiar de însuși robotul. O a doua situație tipică ar fi utilizarea pauzelor, când programul principal reprezintă un ciclu nesfârșit, întrerupt de diferite evenimente, ceea ce face ca robotul să funcționeze în conformitate cu un eveniment extern sau un interval de timp. Acest lucru este ilustrat de algoritmul grafic din figura de mai jos.

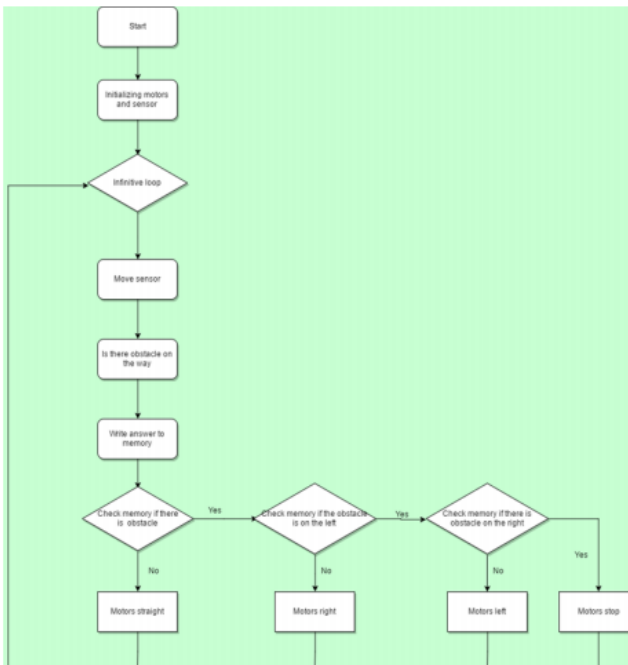


Figura 1.4 - Exemplu de algoritm ce gestionează comportamentul robotului

Studierea programării într-un limbaj bazat pe text, împreună cu bibliotecile acestuia, durează mult timp. Cu toate acestea, programele create în acest sistem utilizează în mod econom memoria și sunt mai rapide decât programele scrise într-un limbaj grafic.

Întrebări pentru revizuire:

1. Descrieți rolul unui program în funcționarea robotului.
2. Care sunt diferențele dintre programarea grafică și programarea bazată pe text?
3. De ce nu este necesar să se înceapă lucrul într-un mediu de programare grafică prin crearea unui algoritm grafic?
4. Descrieți importanța bibliotecilor în programarea bazată pe text.
5. De ce este necesar să se pregătească un algoritm grafic corect înainte de programarea bazată pe text?

Instrucțiuni metodice și practice în ajutor instructorilor în organizarea activităților de Robotică

Nimic nu funcționează din prima de ce?

Dacă totul ar fi funcționat din prima, nu am învăța nimic. A rezolva creativ problemele înseamnă a vedea potențialul în ceea ce ... încă nu funcționează. În acest caz, punem în practică ceea ce funcționează, intervenind cu modificări în ceea ce nu funcționează.

Conținutul din următoarele capitole cuprind exemple de programe, precum și instrucțiuni privind construcția roboților. Acestea constituie un bun punct de plecare. Odată actualizate, devine evident faptul că mai există suficiente posibilități de îmbunătățire a programelor pentru a le face mai eficiente în rezolvarea problemelor din lumea reală și pentru îndeplinirea unor provocări sau sarcini practice. Pentru a o face, trebuie să ne punem întrebări, să căutăm soluții alternative ca să ne atingem scopurile, să elaborăm planuri, să creăm și să testăm soluții mai bune.

Experții

Veți putea descoperi în curând că sala este plină de experți! Adresați-vă lor în cazul în care aveți nevoie de ajutor. Nu pretindeți a fi expert. Sarcinile au fost proiectate astfel încât să îi ajutați pe copii să lucreze singuri în grupuri. Considerați că este vorba de un experiment important în cadrul căruia toată lumea învață și creează cunoștințe noi. În cadrul acestui studiu, nici unul dintre noi nu știe ce se va întâmpla în continuare. Este un pic înfricoșător, dar este adevărat!

Recomandări pentru o mai bună eficiență a activităților:

- ✓ REGULA CELOR CINCI - Dacă un utilizator ajunge la o problemă pe care a încercat să o rezolve și rămâne blocat, se va recurge la regula următoare: „solicitați ajutorul altor patru utilizatori, înainte de a cere ajutorul trainerului“.
- ✓ TRAINERUL ÎȘI REZERVĂ DREPTUL SĂ NU CUNOASCĂ RĂSPUNSUL – Dvs., trainerul, aveți dreptul absolut să nu cunoașteți soluția unei probleme.
- ✓ RESURSE PE NET - Acestea se pot dovedi a fi într-o engleză tehnică, însă pot fi utile!
- ✓ ÎNTREBAȚI EXPERȚII ÎN EV3 – Puneți întrebarea tuturor celor prezenți în clasă. Puteți descoperi că sala este plină de experți! Solicitați ajutorul lor.
- ✓ Căutați pagina de start privind robotica
- ✓ NU GĂSIȚI NICI O SOLUȚIE? Abordați problema dintr-un alt unghi!
- ✓ INSTALAȚI O TABLĂ pe perete unde veți afla problemele dificile. Atașați aici descrierea problemei. Apoi treceți la următoarea problemă. Eventual ... cineva va rezolva problema anterioară. Pentru rezolvarea acestor probleme ar putea fi distribuite puncte sub formă de bonus.

Pregătirea sălii

✓ Pentru a vă deplasa - Trebuie să dispuneți de suficient spațiu pentru a vă deplasa între bănci. Dacă este posibil, plasați băncile una lângă alta pentru a forma grupuri a câte patru utilizatori. Roboții se pot deplasa în orice direcție, prin urmare,

unul dintre utilizatori trebuie să-i prindă înainte ca aceștia să cadă! Băncile cu borduri înalte ar constitui un avantaj pentru ca obiectele să nu cadă de pe ele.

✓ *Siguranță* - Fixați mai bine cablurile trase pe podea pentru a evita riscul de împiedicare accidentală și cădere. Cablurile ar putea fi amplasate de-a lungul tavanului, iar prizele ar putea fi suspendate deasupra băncilor, dacă este posibil în condițiile specifice sălii.

✓ *Lângă fiecare calculator* - trebuie să fie disponibil un spațiu de 60 x 40 cm, astfel încât robotul să dispună de spațiu pentru a se deplasa.

✓ *Spațiu de testare* - pentru testarea finală sunt necesare două mese, bănci sau un planșeu fără obstacole, unde utilizatorii își vor putea demonstra soluțiile și vor putea organiza competiții. În acest sens, un planșeu cu o suprafață de 2m x 4m ar fi suficient. În mod ideal, ar fi bine ca acesta să se afle în mijlocul sălii, toți utilizatorii având la el acces egal.

✓ Toate bateriile ar trebui să fie încărcate complet înainte de a se începe lecția. Bateriile pot fi lăsate să se încarce peste noapte, dat fiind că nu există riscul de supraîncărcare.

✓ Pentru îndeplinirea sarcinilor practice obișnuite, grupurile de lucru ar trebui să constea din maximum 2 utilizatori. Pentru sarcini complexe, ar trebui să colaboreze (coopereze) câte două grupuri!

Gestionarea pieselor LEGO

Pentru realizarea lecției de robotică, este important să se planifice modul în care piesele LEGO vor fi distribuite utilizatorilor. Există mai multe modalități de a o face, iar metoda corectă depinde de numărul de utilizatori, precum și de numărul de kituri distribuite. În continuare sunt prezentate 3 dintre metodele cele mai răspândite.

Kituri - Fiecare echipă primește unul sau mai multe kituri Mindstorms. Când se lucrează cu mai multe casete, rețineți că nu trebuie amestecate piesele din kituri diferite. Mai mult ca atât, nu ar trebui ca mai multe echipe să împartă piese între ele, dacă kiturile trebuie să rămână asortate și complete. Această metodă comportă mai multe dezavantaje. În cadrul unor proiecte de durată, când la construcția roboților se lucrează ore în șir, kiturile sunt folosite și nimeni altcineva nu are acces la acestea. Și utilizarea unor kituri suplimentare se poate dovedi problematică. Deoarece unele piese din kitul principal și din cele suplimentare se suprapun, unele dintre ele urmând, în mod obligatoriu, să fie folosite cu celelalte. Distribuirea unor kituri suplimentare se dovedește a fi complicată, de vreme ce numărul acestora este mai mic față de cel al kiturilor principale și poate să nu fie suficiente pentru fiecare echipă. Această metodă presupune că roboții și kiturile sunt marcate pentru a împiedica crearea robotului greșit cu ajutorul kitului greșit. Sistemul bazat pe kituri este convenabil, atunci când aceeași echipă folosește un singur kit pe parcursul întregului curs. În acest caz, repartizarea kiturilor poate fi efectuată în ultima oră a cursului, iar gestionarea kitului în timpul cursului revine nemijlocit echipei.

✓ *Avantaje*: convenabilă pentru trainer, kiturile sunt ușor de transportat într-o altă sală, construcția este simplă.

✓ *Dezavantaje*: activitate de sortare solicitantă pentru utilizatori, piesele din kitul principal și din cele suplimentare se pot amesteca cu ușurință, cantitate limitată de piese pentru fiecare echipă.

Piese comune - Toate kiturile sunt adăugate într-o colecție de piese ce poate fi folosită de către toți utilizatorii. Piesele mari - motoare, senzori, roți - pot fi separate și plasate în unul dintre containerele goale. Deoarece toate piesele vor fi amestecate, construcția roboților va dura mai mult, dat fiind că va fi nevoie de timp pentru a găsi piesele potrivite. În timpul demontării, utilizatorii trebuie din nou să separe toate piesele. Pe de altă parte, procesul de demontare are loc mai repede și poate economisi din timp la sfârșitul lecției. Ar putea fi amenajată o bancă separată specială pentru piese. Poate fi o masă obișnuită, echipată cu o margine (bordură) de 5-10 cm înălțime pentru ca piesele să rămână pe suprafața acesteia și să nu cadă. Un avantaj imens al acestei metode este că aceasta poate fi folosită pentru ateliere de lucru de către toți participanții. Este vorba de muncă suplimentară pentru trainer, însă dacă roboții sunt concepuți astfel încât unitățile de control EV3 să fi ușor detașabile, trainerul le poate detașa la sfârșitul lecției și le poate reatașa înainte de începutul lecției următoare. Între timp, alți utilizatori pot folosi unitățile de control EV3.

✓ *Avantaje:* Nu este nevoie să fie sortate piesele, dat fiind că demontarea acestora are loc repede, convenabilă pentru trainer, un număr mai mare de utilizatori pot lucra cu kiturile.

✓ *Dezavantaje:* Nu există o perspectivă a pieselor, construcția necesită mai mult timp, piesele sunt greu de transportat.

Piese comune asortate - Similară cu metoda precedentă, dar pe lângă faptul că sunt împărțite, piesele sunt, de asemenea, asortate. Plăcile de plastic din partea inferioară a containerelor sau containerele goale pot fi utilizate pentru sortarea pieselor. Comparativ cu metoda anterioară, construcția robotului are loc mai repede, în timp ce demontarea durează puțin mai mult, deoarece piesele trebuie, de asemenea, sortate. Trebuie de remarcat faptul că utilizatorii tind să se grăbească și să amestece piesele similare. O soluție ar fi de a sorta toate piesele în urma demontării unui robot pe plăci de plastic separate, apoi fie trainerul, fie un utilizator, le va distribui mai târziu într-un container comun. Sortarea pieselor nu va dura mult, atâta timp cât utilizatorii vor ține cont de faptul că piesele sunt organizate pentru a le facilita construcția robotului și le vor păstra în ordine.

✓ *Avantaje:* Construcția nu necesită efort, mai mulți utilizatori pot folosi kiturile simultan, o perspectivă a pieselor utilizate.

✓ *Dezavantaje:* Sortarea consumă timp, transportarea pieselor este complicată.

Toate aceste metode prezintă avantaje și dezavantaje. Alegerea metodei și caracterul practic al acesteia depind în mare măsură de situație. Cu siguranță, este posibil de a folosi soluții creative, complet diferite.

În cazul LEGO EV3, este util de a construi roboții astfel încât unitățile de comandă EV3 să poată fi detașate cu ușurință. Această soluție este utilă, de asemenea, pentru proiecte de durată, atunci când utilizatorii programează și lucrează la același robot timp de câteva săptămâni. Când toate kiturile Mindstorms disponibile sunt combinate, roboții EV3 pot fi utilizați simultan în cadrul mai multor lecții sau ateliere.

Aspecte legate de programare

Puteți alege un mediu de programare potrivit pentru utilizatori. Pentru a începe, cel mai bun mediu de programare este EV3-Graphical, deoarece se evită aspectele legate de sintaxă. Cu toate acestea, majoritatea utilizatorilor se plictisesc în cele din urmă, deoarece simt că au progresat și au atins limita capacităților oferite de EV3-G. Acest lucru nu este chiar adevărat. Programele EV3-G mai complexe necesită mai mult efort pentru simplificarea programelor sau pentru crearea unor algoritmi eficienți. Aceasta înseamnă a face mai multă muncă în cât mai puține blocuri de cod posibil. Python/Java permit cu ușurință acest lucru, însă majoritatea erorilor ce apar țin de sintaxă. Secțiunile practice legate de cod sunt scrise în EV3-G.

La remedierea erorilor EV3-G, primul lucru ce urmează a fi făcut este de a verifica dacă cablurile sunt bine conectate. S-ar putea întâmpla ca un cablu să fie conectat, dar să nu fie introdus până la capăt în port. A doua problemă cel mai des întâlnită se referă la introducerea cablului în portul greșit. Asigurați-vă întotdeauna că fiecare senzor este conectat la portul potrivit. Doar după ce verificați aceste două lucruri, ar trebui să începeți remedierea erorilor din cadrul programului.

Stimulare și notare

La lecțiile de robotică punctajul ar trebui să fi repartizat în proces, în baza sarcinilor practice. Luați în considerare natura subiectului și, anume, faptul că la evaluarea activității în clasă a utilizatorilor nu este întotdeauna posibil de a elabora un rezultat final funcțional. Prin urmare, repartizarea punctajului ar trebui să se bazeze pe procesul de lucru și pe analiza rezultatului final. Ar trebui luate în considerare succesele și eșecurile, lucrurile ce ar fi putut fi realizate mai bine pentru a atinge un rezultat mai bun. La fel, joacă un rol important și feedbackul oral din partea utilizatorului, atunci când acesta intervine cu explicații referitor la ceea ce s-a realizat. Se va acorda atenție atât programelor finalizate, cât și gradului în care utilizatorul poate explica în detaliu activitatea programelor folosite.

Cea mai importantă sarcină este cu siguranță proiectul ce urmează a fi finalizat până la sfârșitul cursului, ce ar trebui să rezume cunoștințele dobândite pe parcurs. Se va evalua rezultatul final al proiectului, în special, în baza integrității soluției. De asemenea, se vor lua în considerare diferitele părți ale sistemului în ansamblu: munca mecanică, programul, lucrul în echipă, documentația și prezentarea. Vă recomandăm să folosiți o diagramă pentru a nota punctajul. În diagrama cu scoruri pentru proiectul Vehicul inteligent și Peștera, punctajul acordat pentru valoarea totală a activității este după cum urmează:

- * Costul robotului
- * Creier – 5
- * Senzor – 3
- * Fiecare 50 de grame a tuturor celorlalte piese – 1
- * Cheltuielile de lucru ale echipei
- * Numărul membrilor din echipă x orele de lucru x 5

* Fiabilitatea și performanța robotului

Sisteme robotice

În acest capitol, vom explora sistemele robotice și mecatronice, componentele și structurile acestora. În plus, vom analiza funcționalitatea controlerului și semnalul digital simplu.

Sisteme mecatronice

În general, un sistem mecatronic constă din senzori, actuatori și software pentru a-i controla. Aceste tipuri de sisteme mai sunt numite dispozitive inteligente, deoarece pot îndeplini operații complexe, se pot adapta mediului și pot interacționa cu mediul ambiant, în funcție de complexitatea dispozitivului de control. Robotica este unul dintre domeniile în care se aplică cel mai des mecatronica. Când vorbim despre robotică, majoritatea oamenilor își imaginează un fel de mașină inteligentă sau un umanoid (un robot asemănător cu omul). Pentru persoanele care activează în domeniul industriei, robotica se asociază, de obicei, cu manipulatorii, roboți care execută operații de sudură și vopsire etc. Prin urmare, oamenii au o înțelegere absolut diferită despre robotică, o serie variată de dispozitive putând fi numite roboți, atât timp cât îndeplinesc interacțiunea (senzor-controler-actuator) descrise în capitolul precedent.



Figura 2.1 ilustrează un exemplu de sistem mecatronic. Săgețile indică interacțiunea și direcția acesteia între diferite părți ale sistemului și mediul extern.

Într-un sistem mecatronic senzorii primesc informații din mediul extern, controlerul folosește aceste informații pentru a lua decizii și pentru a reacționa prin intermediul actuatorilor (de exemplu, prin mișcare).

Senzorii colectează diferite tipuri de informații din mediul înconjurător (temperatură, distanța față de alte obiecte, semnale radio) sau de la dispozitiv (viteză, accelerație, turațiile motorului) și le transmit modulului de comandă. Un modul de comandă este, de obicei, un controler în care se realizează algoritmul, altfel spus programul. În conformitate cu programul și informațiile primite de la senzori, controlerul dirijează mecanismele actuatorilor, care adesea sunt

motoare electrice. Solenoizii, mușchii artificiali, cilindrii pneumatici etc. pot fi și ei actuatori. De obicei, actuatorii produc un fel de mișcare, ce influențează mediul extern. Un exemplu ar fi acțiunea unui sistem de securitate auto simplificat în situații de urgență (Figura 2.2).

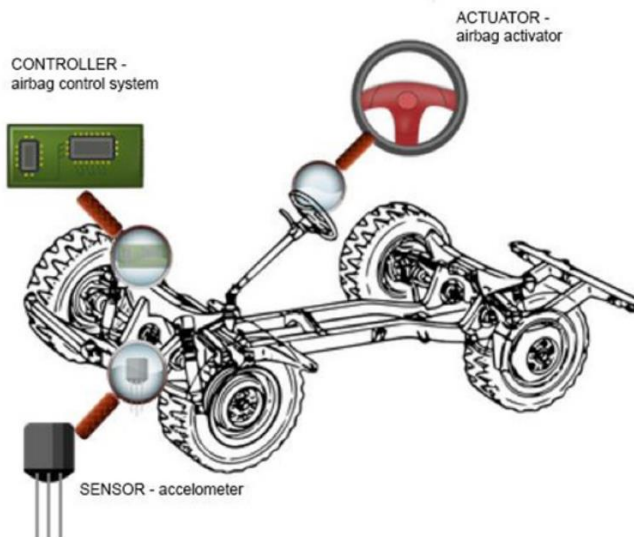


Figura 2.2. Airbag-ul automobilului și model schematic al sistemului.

Sistemul airbag este un sistem mecatronic tipic. Constă din trei părți importante: un senzor, un controler și un actuator. Senzorul de accelerație înregistrează schimbarea bruscă a accelerației și transmite un semnal corespunzător către controler. Algoritmul controlerului oprește toate activitățile automobilului și trimite un semnal de declanșare către dispozitivul de umflare, ce este conectat la airbag. Dispozitivul de umflare deschide airbag-ul, umplându-l rapid cu gaz, și protejează șoferul de impactul cu volanul. Acum, când am determinat lanțul senzor - controler - actuator, putem spune că acesta este un sistem mecatronic.

Microcontrolere

Una din părțile cele mai importante ale unui sistem mecatronic este microcontrolerul sau, pe scurt, controlerul, ce procesează informațiile și le trimite către actuatori. Un controler poate fi comparat cu un calculator mic, ale cărui componente (procesor, memorie, convertor intrare-ieșire etc.) sunt plasate pe un singur chip. De aceea, este adesea numit microcontroler. În acest caz, nu putem compara funcțiile pe care le îndeplinește acesta cu cele ale unui calculator personal, dar acestea sunt suficiente pentru a controla un sistem robotic. Microcontrolerul este similar cu un circuit logic obișnuit. O diferență importantă dintre acestea este faptul că pinii de ieșire ai microcontrolerului pot fi programați pentru a acționa în funcție de cerințele și opțiunile controlerului. Aceasta înseamnă, că în funcție de program, unul și același pin fizic poate fi de intrare (informația sau un semnal se deplasează spre controler) sau de ieșire (controlerul, la rândul său, transmite un semnal sau informație). Diferiți senzori pot fi conectați la intrare și diferiți actuatori la ieșire. În cazuri simple,

semnalele sunt semnale digitale logice (0 sau 1), unde 0 arată că nu există tensiune electrică și 1 indică prezența tensiunii (de ex., 5V). În figura 2.3, puteți vedea forma diferitelor microcontrolere și o reprezentare grafică a unui semnal digital.

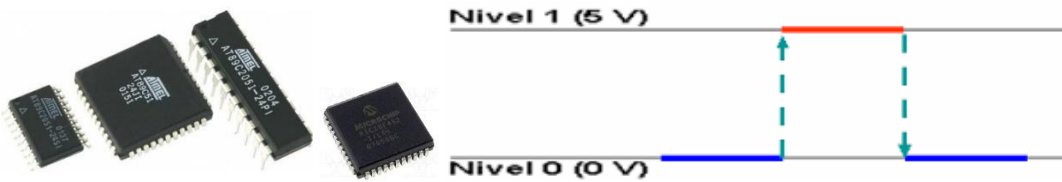


Figura 2.3. Microcontrolere de diferite tipuri (stânga) și un semnal digital simplu: logic 0 și logic 1 (dreapta).

O particularitate a controlerelor este gama variată a acestora. Unele sunt simple și ieftine (potrivite pentru gestionarea proceselor mai simple); oarecum, altele sunt mai complexe și mai puternice, cu o mulțime de pini de intrare-ieșire. Controlerelor sunt, de asemenea, clasificate după funcționalitate și domeniu de aplicare. De exemplu, cerințele față de controlerul utilizat în industria producătoare de automobile sunt diferite față de cele pentru controlerul utilizat în mașinile de uz casnic.

Pentru a controla sistemele mecatronice și roboții este important de a cunoaște posibilitățile microcontrolerelor și de a putea compila un program pentru un anumit controler. Prin urmare, a doua parte a capitolului prezintă principalele posibilități și funcții ale controlerelor.

Întrebări pentru revizuire:

1. Care sunt principalele părți ale unui sistem mecatronic?
2. Care este funcția unui controler într-un sistem mecatronic?
3. Ce tipuri de dispozitive oferă informații roboților?
4. Furnizați două exemple (de dispozitive ce pot fi conectate cu un controler) când pinul fizic al microcontrolerului ar trebui să fie setat ca: a) ieșire b) intrare.

Obiective de învățare:

- Utilizatorul înțelege că un sistem robotic constă din trei părți: senzor, controler și actuator;
- Utilizatorul este capabil să descrie programe fie prin descriere, fie prin algoritmi, în funcție de vârstă;
- Utilizatorul poate începe programe legate de roboți.

Începător, intermediar, avansat

Scopul este ca utilizatorii să înțeleagă și să ghicească logica programelor. Profesorul trebuie să descarce în memoria roboților înainte de lecție programele pentru toți roboții implicați la instruire. În cadrul acestei lecții nu se face programare.

✓ Descărcați programul „Sisteme Robotice.ev3” pentru toți roboții.

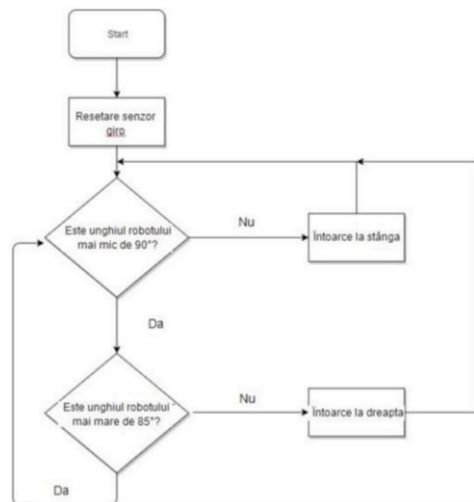
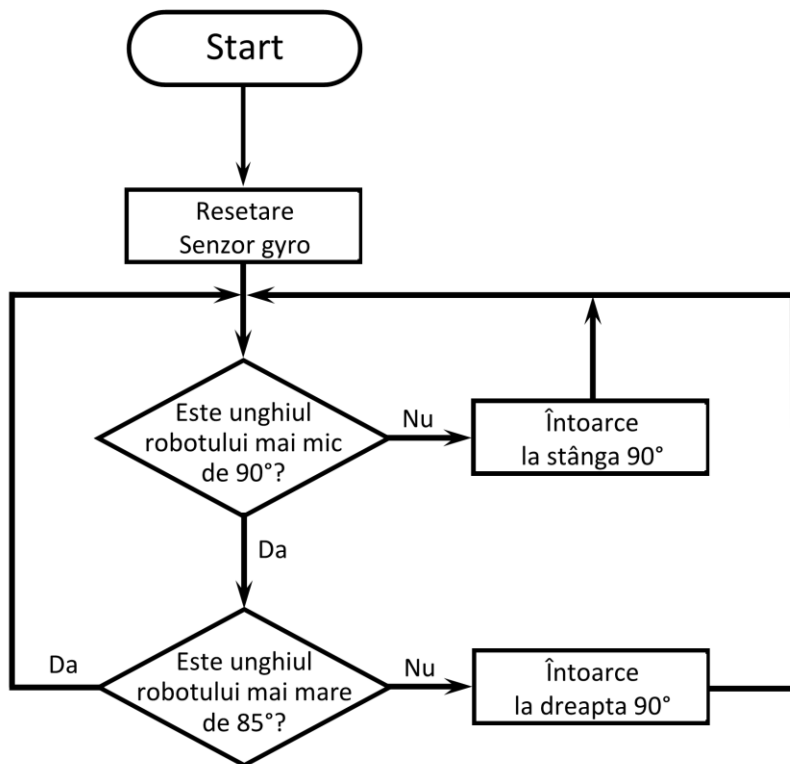
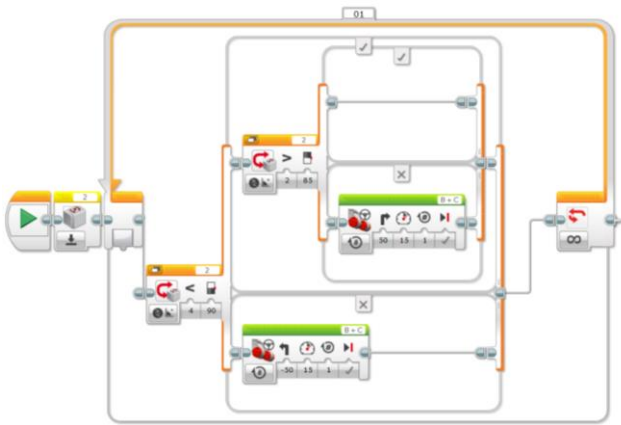
✓ În robot, în mapa „Sisteme Robotice”, utilizatorii vor găsi programele „1”, „2”, „3”, „4” și „5”. Se va da câte un robot la doi utilizatori. Lăsați-i să lanseze programul în mod repetat pentru a înțelege ce declanșează mișcarea robotului. Lăsați-i să noteze pe hârtie o descriere sau un algoritm.

✓ Comparați descrierile sau algoritmii diferitor grupuri.

În continuare, veți găsi algoritmii și programele pentru toate cele cinci variante de sarcini. Utilizatorii ar putea să construiască acești algoritmi un pic diferit.

Începător:

Algoritmi 1

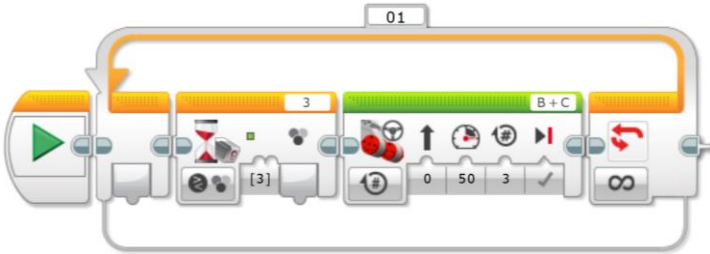


Din mapa robotului, numită Sisteme Robotice, porniți programul unu. Solicitați ajutorul trainerului, dacă este necesar. Încercați să interacționați cu robotul atunci când rulează programul și să ghiciți ce îi comandă programul robotului să facă.

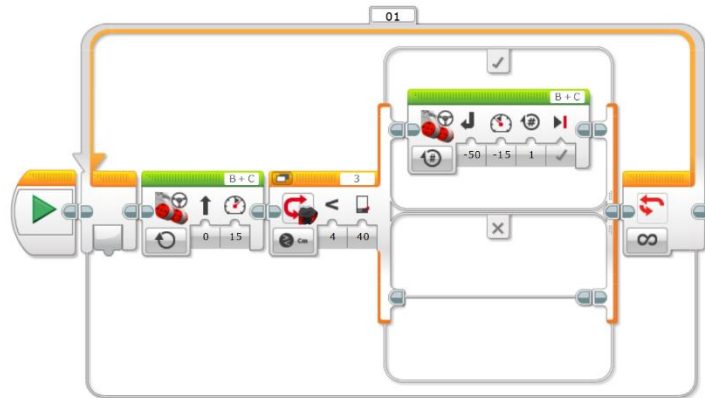
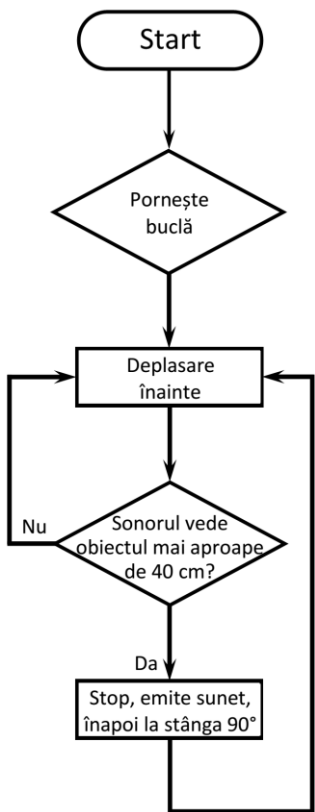
Descrieți programul.

Ar putea fi ceva de genul:

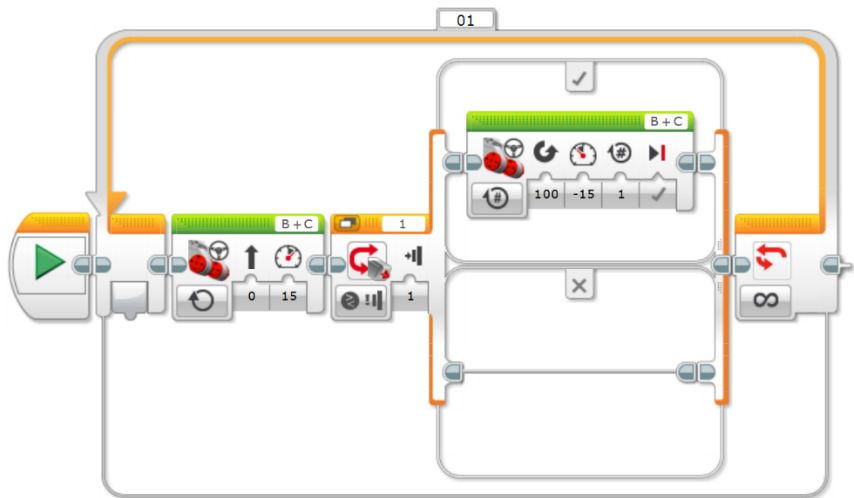
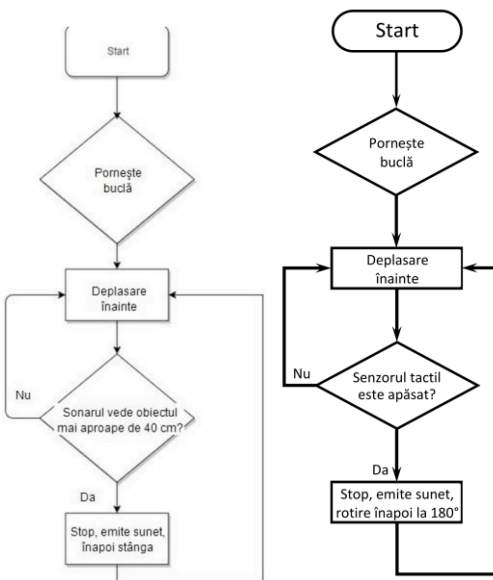
Acest program face ca robotul să se miște când este activată culoarea verde a senzorului. Robotul nu acționează dacă vede altă culoare.



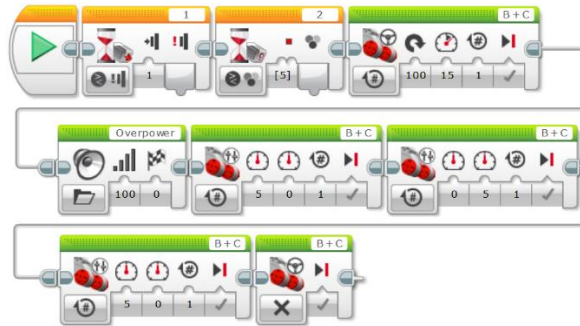
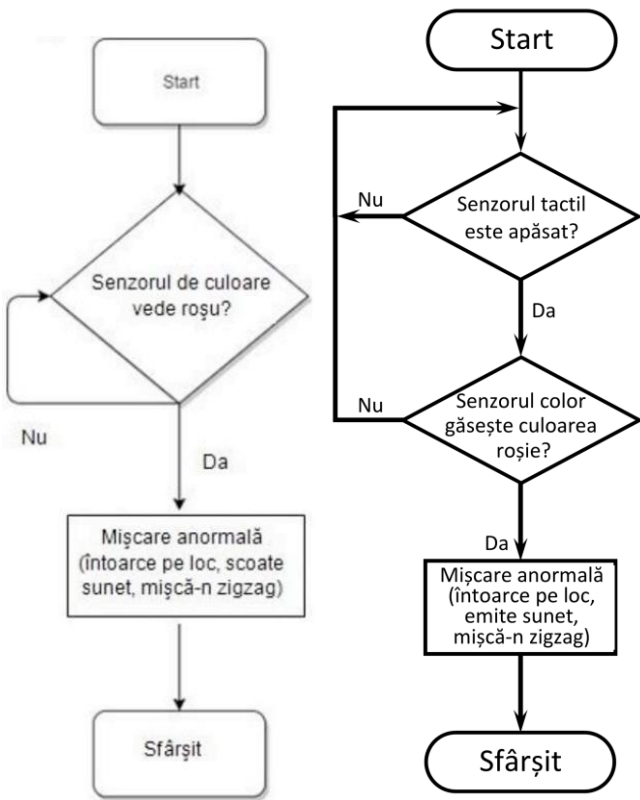
Program nr. 2.



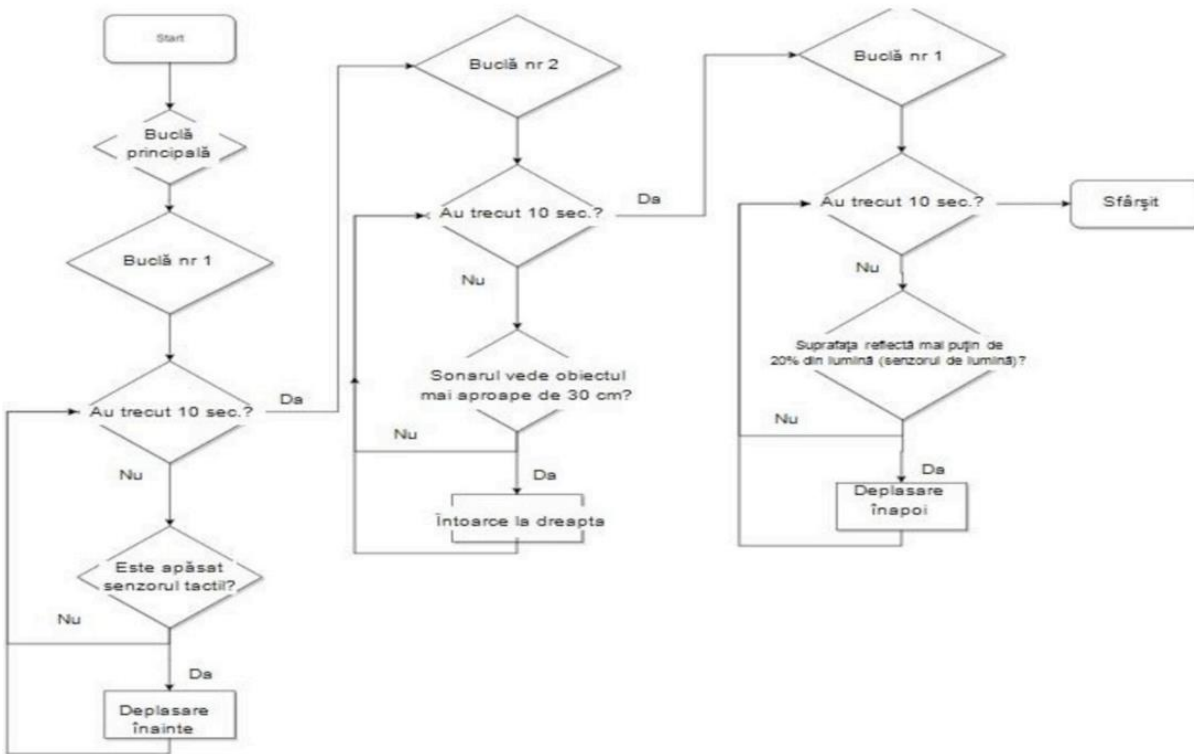
Program nr. 3.

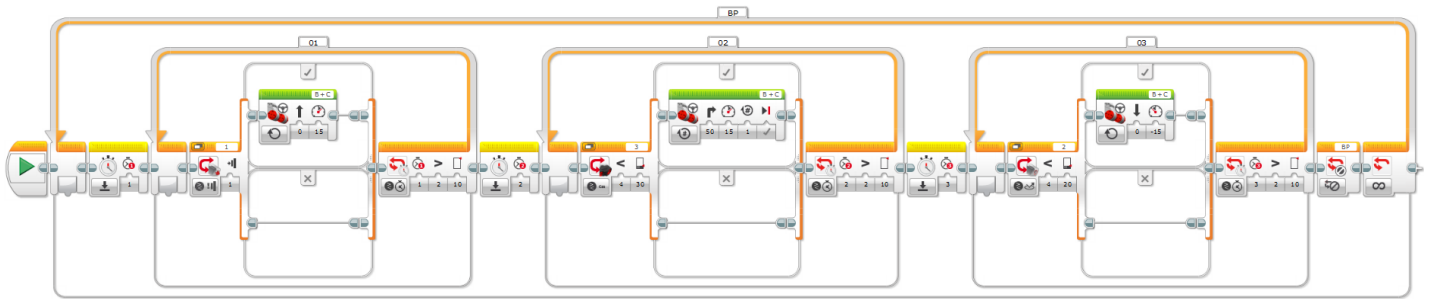
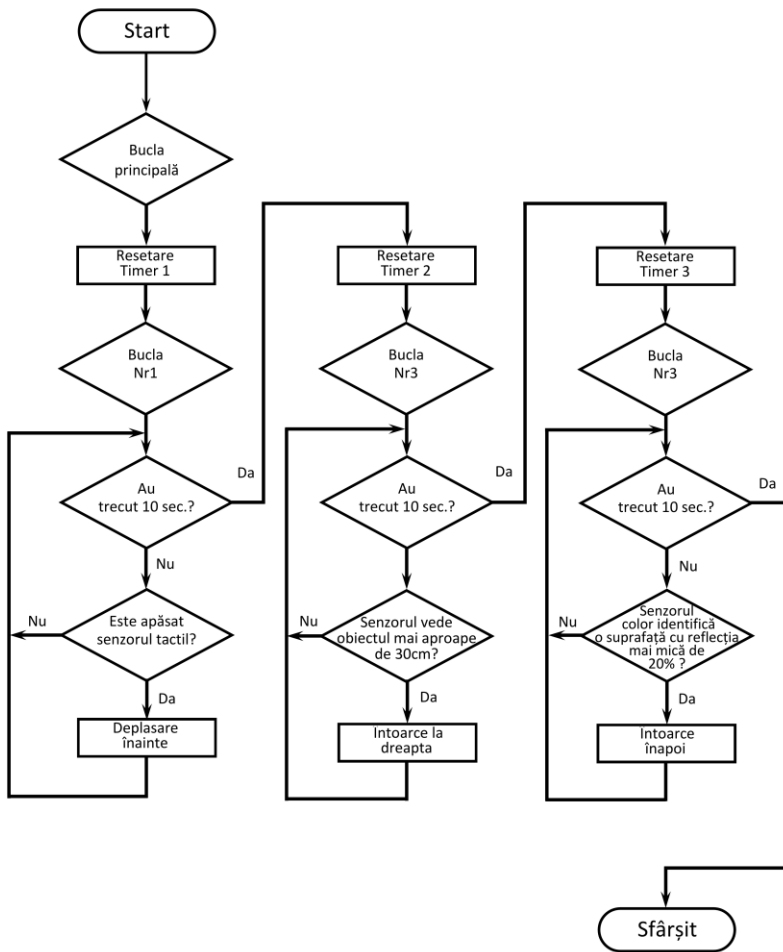


Program nr. 4.



Program nr. 5.





Coaching-ul poate fi una dintre cele mai pline de experiență din viața ta.

Scopul conținutului acestui ghid este pentru a vă ajuta în organizarea activității clubului de robotică și operarea eficientă cu membrii echipei prin distracția de a cunoaște știința, tehnologia, ingineria și matematica. Dacă echipa dvs. primește sau nu un premiu la un concurs, membrii echipei câștiga doar prin participare.

Dacă este anul dvs. începător, bucurați-vă de ceea ce este: o experiență de învățare. Scopul tău ar trebui să simțiți experiența FIRST® LEGO® League pentru prima dată. Cu o distracție experiență și atingeți obiective realiste sub centură, voi și copiii veți fi plin de idei despre ce să facem anul viitor.

Formarea unei echipe de robotică începe: de la 4 până la 16 copii participanți + 2 traineri = care împreună formează 1 echipă de robotică. O echipă care aspiră să participe la concursul național FIRST LEGO League este formată de la 2 până la 10 membri participanți și 2 traineri adulți. Este important ca membrii echipei să aibă vârsta cuprinsă între 9 și 16 ani. Nici un membru al echipei nu poate să fie peste vârsta maximă înainte de 1 ianuarie a anului când începe noul sezon de concurs.

Trebuie de ținut cont de aspectul multilateral în procesul de selectare a participanților întrucât rolul de bază al competiției este lucrul în echipă și problemele sau provocările cu care se confrunta echipa, să fie soluționate la fel în echipă și să demonstreze capacitatea de a rezolva sarcini majore! Vă încurajăm să încadrați membrii cu diferite vârste întrucât viziunile asupra tratării sarcinilor sânt diferite și este probabilitatea să aveți o soluție mult mai reușită cu ajutorul acestora, dar nimeni nu interzice să păstrați forma de abordare cu cât este mai simplu cu atât e mai bine. Trainerul nu se ocupă de programarea sau de cercetarea propriu-zisă, ci ajută echipa să găsească propriile soluții.

Membrii echipei

Dacă aveți un membru al echipei care are nevoie de o abordare mai diferită, trebuie să mai aveți puțină cunoștință din psihologie pentru a-l ajutat să se încadreze mai ușor în echipă. Deseori copiii care sunt mai retrași și preferă să muncească de unul singur pot genera idei foarte geniale, doar că trebuie de găsit instrumentele (cheia) care i-ar permite să se regăsească și să se afirme ca membru al echipei, și în același timp să conștientizeze că conlucrarea din cadrul echipei va aduce succese întregii echipe. Este mult mai prețuit câștigul întregii echipe decât performanța unui singur membru.

În ceea ce privește alegerea componenței echipei formate din cei mai buni copii, nu în toate cazurile aceasta va aduce la o echipă de performanță ba în unele cazuri la una nelucrative (foarte sterilă). Asigurați-vă că alegeți o echipă cu o gamă variată de aptitudini, având în vedere faptul că acei copii, care sunt după natura lor cu capacități native de auto-regizare a activităților, se vor încadra mai ușor în acest joc al competiției.

Trainerul

În calitate de trainer, sunteți un facilitator pentru a vă ajuta echipa să își finalizeze activitatea și să-și îmbunătățească modul de conlucrare împreună. Echipa are nevoie de dvs. ca să le oferiți îndrumare și să le oferiți ghidarea, încurajarea și, mai presus de toate, o experiență distractivă.

Sunteți responsabil pentru îndrumarea echipei în dezvoltarea obiectivelor sale și realizarea sarcinilor conform calendarului planificat și programului întâlnirilor echipei. Trainerul face legătura armonioasă dintre membrii echipei, mentori, părinți și voluntari. Străduiți-vă să-i cunoașteți pe toți din echipa dvs. Încercați să aflați de la ei care sunt așteptările lor de la frecventarea clubului de robotică. Trainerul controlează procesul, nu conținutul. O echipă de începători ar putea avea nevoie de ajutor în procesul de învățare cum să utilizați software-ul de programare, înțelegerea conceptelor de inginerie sau învățarea cum să faceți cercetări în viața cotidiană. Dvs. și alți adulți vă puteți ajuta echipa să învețe aceste abilități sau concepte atâta timp cât nu le spuneți copiilor cum să rezolve, doar îi ghidați spre idei.

Provocare.

Membrii echipei trebuie învățați să ia decizii asupra construcției diverselor forme de roboți pentru joc sau să dezvolte diverse forme de proiect doar în cadrul consultării de echipă. Aici trebuie să fie inclusă decizia privind strategia, construirea, programarea, cercetarea, alegerea unei probleme, alegerea unei soluții inovatoare și prezentarea către un public pentru apreciere.

Nu este recomandat să le fie dată soluția, dar și nici nu trebuie lăsate lucrurile să ajungă în situații de impas, se poate de ajutorat echipa prin adresarea de întrebări precum:

- ▲ "Ce ați putea schimba pentru a obține rezultatul dorit?"
- ▲ "Cum va afecta asta ...?"
- ▲ "De ce informații aveți nevoie pentru a răspunde la această întrebare?"

Copiii sunt cei care rezolvă problema, ei înșiși găsind soluții! Noi înțelegem ca adulții pot deveni la fel de pasionați de concurs ca și copiii, dar ei trebuie să-și amintească întotdeauna că prioritate au copiii.

Scurtă descriere a activității [Rezumat]

Acest atelier de lucru inspiră utilizatorii să gândească și să își imagineze utilizarea roboticii în viața reală. Utilizatorii discută și învață ce este un robot și modul în care roboții pot fi utilizați în viața de zi cu zi prin demonstrații și exerciții de creativitate. Activitatea este special concepută pentru a dezvolta abilități de genul:

Creativitate: în cadrul atelierului utilizatorii învață și exersează proiectarea și generarea ideilor inovatoare pentru a aborda problemele din viața reală, folosind tehnicile lui Tony Buzan de Mind Mapping (Hartă Conceptuală).

Fluență digitală: Utilizatorii discută, învață și utilizează elementele cheie ale roboticii, generând idei de utilizare creativă a roboticii, precum și construirea, programarea unui robot real (concret).

În special, utilizatorii învață și discută principalele elemente ale roboticii (tehnologie); construirea unui robot (tehnologie și inginerie), elaborarea unui program vizual pentru a controla robotul și a executa sarcini (tehnologie); dezvoltarea abilităților de gândire creativă necesare pentru a găsi diferite aplicații pentru robotică în alte domenii.

Comunicare: în timp ce lucrează în echipă, utilizatorii construiesc un robot urmând instrucțiunile generice. Ei sunt încurajați să descrie în mod clar și să comunice planurile și ideile lor privind modul de construire a corpului robotului; conectarea corectă a componentelor de bază și programarea robotului. Instructorii explică utilizatorilor că aceștia pot face greșeli și că ei trebuie să identifice și să refacă porțiunile corespunzătoare pentru a construi și programa robotul.

Colaborare: întrucât utilizatorii din fiecare echipă își schimbă rolurile în cadrul fiecărei sarcini, ei învață cum să conlucreze eficient și respectuos cu ceilalți membri pentru a construi un sistem robotic corespunzător de complex.

Robotica este interesantă și distractivă, motiv pentru care aceasta este un instrument minunat de a implica copiii în știință, tehnologie, inginerie și matematică. Este bine-cunoscut faptul că învățarea prin practică îi motivează pe copii să devină mai creativi, iar robotica îi ajută să mențină curiozitatea lor naturală despre lumea înconjurătoare și tehnologie.

Centrul de resurse – ghiduri și instrucțiuni

Centrul de resurse este dezvoltat și actualizat de profesorii de robotică. Link-ul ce conține toată informația necesară - <https://goo.gl/LjpE00>

- Ghidul general pentru utilizarea roboților LEGO Mindstorms EV3 - <https://drive.google.com/open?id=0B0hWyJLQRKc7VnBFMEtmc19JRm8>
- Instrucțiuni pentru asamblarea roboților LEGO MindStroms EV3 - <https://drive.google.com/open?id=0B0hWyJLQRKc7QW1uVjdJeEZZeEk>
- Scenarii și activități - <https://drive.google.com/open?id=0B0rIkKOUi341Wnh2QXJMa05sWU0>
- Introducere în programarea roboților LEGO MindStorms EV3 - <https://drive.google.com/open?id=0B0hWyJLQRKc7YXJYZmdnbV9GbFE>
- Concursul FIRST LEGO League - <https://drive.google.com/open?id=0B0hWyJLQRKc7b3hMLWJ3dFBzU0U>

Procurarea roboților LEGO MINDSTORMS EV3

Roboții LEGO MINDSTORMS EV3 pot fi procurați de la compania "KIDSCO SRL", adresa mun. Chișinău, str. Valea Crucii 22/1, oficiu 101. Telefon +373 22283656, email - andrei.bagrin@gmail.com

Prețul variază în dependență de cantitate și alți factori. Costul mediu ar fi în jur de 400 Dolari SUA.

De asemenea seturile pot fi comandate de la compania **OU Estrotech** din Estonia. Tel. +372 51 46091, pood@robomiku.ee, www.robomiku.ee.

Link-uri către alte companii care afișează pe site-ul lor acest model de roboți -

<http://www.dostavka.md/ro/buy/childish/toys/Konstruktori/LEGO-LEGO-MINDSTORMS-EV3.1770155>

<https://smadshop.md/ro/produse-copii/constructor-pentru-copii-lego-mindstorms-ev3.html>

<http://maxmart.md/ro/produse/214269/lego-lego-mindstorms-ev3-v24>

Masa pentru misiuni

Proba Robotul în Misiune are loc pe o Masă special concepută; prin urmare, va trebui să construiți o astfel de Masă pentru antrenamente, în cazul în care nu aveți deja acces la una. Un design simplu, care ia în considerare siguranța, greutatea, înălțimea, precum și costurile, este oferit în cele ce urmează, însă atât timp cât suprafața este netedă, iar pereții dvs. de bordură sunt dimensionați și amplasați în mod corespunzător, felul în care construiți structura secundară rămâne la propria latitudine. Deși este destul de simplă, construcția necesită unele abilități de tâmplărie.

Desigur, în cadrul competițiilor, sunt alăturate două mese cap la cap, dar având în vedere că urmează să evoluați doar pe una dintre acestea, pentru antrenamente este suficient să construiți o singură Masă. Puteți considera Masa antrenamentelor ca fiind o "jumătate de masă".

Cele mai multe provocări au o Misiune "comună", a/-l cărui/-or Model/-e urmează a fi amplasate parțial pe Masa dvs. și parțial pe Masa celeilalte echipe. Astfel, pe lângă construcția propriei Mese, va trebui să construiți o mică porțiune a unei a doua Mese, așa încât ambele jumătăți ale Modelului comun să poată încăpea.

Vom numi această secțiune adăugată "Perete de Exercițiu".

Instrucțiunile pentru construcția unei "jumătăți de masă", precum și a Peretelui de Exercițiu, sunt prezentate mai jos:

<i>Material</i>	Cantitate
Kit de Montare a Ringului (Elemente de LEGO ale Modelului p/u Misiune, Covoraș, Panglică adezivă dublă velcro (Dual Lock™))	1
placaj șlefuit (sau alt tip de scândură netedă) 2438mm X 1219mm X cel puțin 10mm	1
scândură / cherestea, 2438mm [secțiune transversală efectivă = 38mm X 64mm]	6
vopsea neagră mată	1/2 L
șuruburi p/u gips carton grosiere, 64mm	1/4 kg
capre de tăiat lemne, aprox. 610mm înălțime / 914mm lățime	2

COMPONENTE

Componentă	Obținută din...	Dimensiuni	Vopsire	Cantitate
Suprafață Masă (A)	foaie de furnir / placaj	2438mm X 1219mm	nu	1
Perete bordură lung (B)	scândură / cherestea	2438mm	da	3
Perete bordură scurt (C)	scândură / cherestea	1143mm	da	2
elemente rigidizare (D)	scândură / cherestea	1219mm	nu	4
capră de tăiat lemne	de achiziționat	Î * 610mm L * 914mm	nu	2

*Dacă utilizați o suprafață de masă cu grosimea de peste 13mm, verificați-i deformarea/distorsionarea - s-ar putea să nu aveți nevoie de elemente de rigidizare.

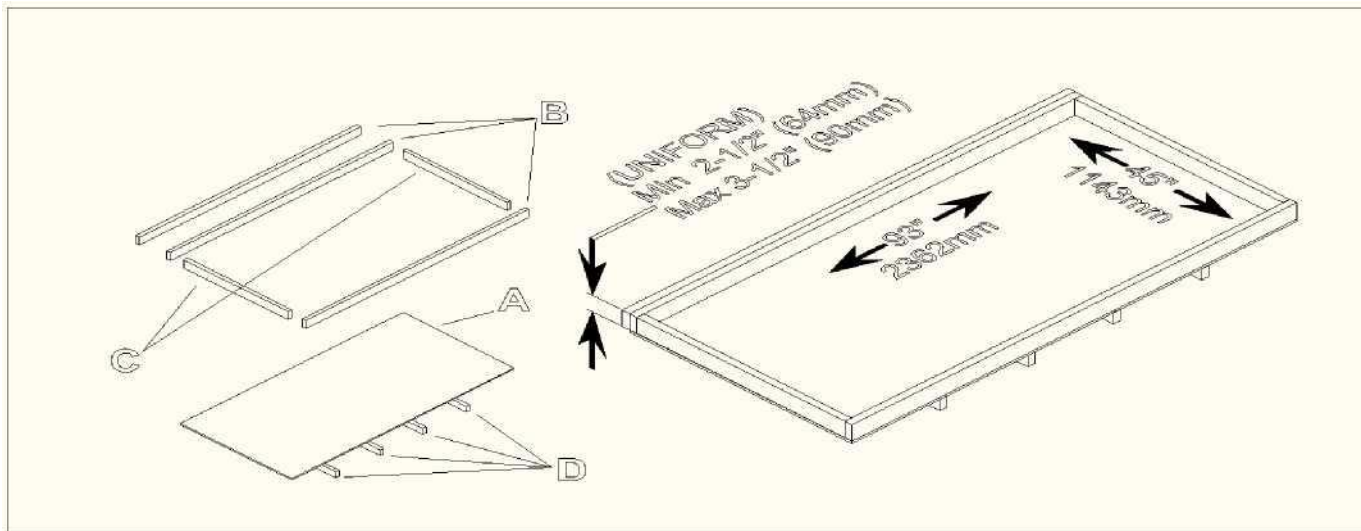
ASAMBLAREA

PASUL 1 - Observați care față a placajului (A) este mai puțin netedă și considerați-o ca fiind fața inferioară. Pe fața inferioară respectivă, prindeți, apoi înșurubați elementele de rigidizare (D) la distanță a câte 457mm. Capul șuruburilor și așchiile nu trebuie să iasă în afară - ceva spre a fi verificat.

PASUL 2 - Pe partea superioară a placajului, amplasați, prindeți și înșurubați Pereții de Bordură (B, C) în jurul perimetrului superior.

- Dimensiunile de la un perete la celălalt trebuie să se încadreze între 2362±3mm pe 1143±3mm.
- Înălțimea B și C trebuie să se încadreze între 64mm și 90mm.
- În cadrul unei competiții, toți pereții de bordură trebuie fie de aceeași înălțime la toate Mesele. Înălțimea bordurilor la competiții poate fi diferită față de înălțimea lor la Masa de antrenament.

PASUL 3 - Fixați tăblia superioară a mesei pe niște capre mici de tăiat



Concursul FIRST LEGO LEAGUE

Concursul FIRST LEGO LEAGUE Moldova se organizează o dată pe an de obicei la sfârșitul lunii februarie de către Asociația Comaniilor din Domeniul IT (ATIC) <https://www.facebook.com/ictmd/>

Pentru a participa la acest concurs o bibliotecă publică trebuie să:

1. Dispună de roboți LEGO MINDSTORMS EV3
2. Dispună de masa pentru misiuni

3. Procure misiunea FIRST LEGO LEAGUE (FLL) pentru anul curent
4. Se înregistreze la concursul FIRST LEGO LEAGUE Moldova pe situl <http://sportsforthemind.eu>
5. Achite taxa de participare la concurs în valoare de 2000 lei moldovenești
6. Să aibă o echipă pregătită de concurs în componență de cel puțin 4 oameni cu vârste între 9 și 16 ani inclusiv.

Concursul FLL Moldova constă din câteva probe, iar echipa câștigătoare se nominalizează prin sumarea punctajului obținut la toate probele. Proba tehnică nu este decisivă.

Echipa câștigătoare obține dreptul de a reprezenta R. Moldova la concursul internațional FLL ce se organizează peste hotare.

Pentru mai multe detalii despre concurs accesați - <https://www.facebook.com/FLLMoldova/>

Misiunile pentru concursul FIRST LEGO LEAGUE

Misiunile pentru concursul FIRST LEGO LEAGUE sunt distribuite doar de partenerii autorizați ai companiei LEGO. Compania **OU Estrotech** este singura companie care poate furniza aceste seturi în R. Moldova. Tel. +372 51 46091, pood@robomiku.ee, www.robomiku.ee.

Site-ul unde este descrisă misiunea pentru anul curent - <http://www.firstlegoleague.org/challenge> este disponibil și în română.

Cererile pentru rezervarea misiunilor se pot depune prin email la adresa pood@robomiku.ee în fiecare an, începând cu luna aprilie. Costul misiunii este în mediu de 3500 lei moldovenești.

Alte concursuri pentru cluburile de robotică

FLL Moldova nu este singurul concurs de robotică. Pe parcursul anului mai pot fi organizate

- SumoBot Challenge – lupta sumo pentru roboți <https://www.facebook.com/events/684587075080515/>
- Robo Liga Bibliotecilor <https://www.facebook.com/events/1994827057421978/>

Biblioteci echipate cu Robotică

1. [BNC "Ion Creangă", Chișinău](#)
2. [BPR Căușeni](#)
3. [BPR Fălești](#)
4. [BPR Telenеști](#)
5. [BPR Rezina](#)
6. [BPO Ialoveni](#)
7. [BP Drujba, Ungheni](#)
8. [BP Izbiște, Criuleni](#)
9. [BP Râșcova, Criuleni](#)
10. [BP Voinova, Strășeni](#)
11. [BP Gălești, Strășeni](#)
12. [BP Molovata Nouă, Dubăsari](#)
13. [BP Cricova, Chișinău](#)
14. [BPR Drochia](#)
15. [BPR Ungheni](#)
16. [BPR Cahul](#)
17. [BPO pentru Copii Glodeni](#)
18. [BPO pentru Copii Orhei](#)

19. BM "B.P. Hașdeu" Chișinău

Contacte utile

Numele	Funcția	Contact
Viorica Bordei	Coordonator principal Robotica (ATIC)	vbordei@ict.md
Adrian Grădinaru	Instructor Robotică cu experiență de instruire a reprezentanților bibliotecilor (Colegiul Politehnic, Chișinău)	adyngcom@mail.ru
Svetlana Lisnic	Reprezentant al BNC „Ion Creangă”, instituție care coordonează cluburile de robotică din biblioteci	svetlana.lisnik@gmail.com
Grupul Facebook al Profesorilor de robotică (cuprinde și instructorii din biblioteci)	Cele mai noi știri despre evenimentele și activitățile legate de robotică apar aici.	https://www.facebook.com/groups/1310851068978810/
Grupul Facebook Cluburi de Robotică@Biblioteci	Informații, știri despre evenimente și experiența cluburilor de robotică din biblioteci	https://www.facebook.com/groups/robotics2018/