

UTILIZAREA INTERFEȚEI LEAP MOTION ÎN COMANDA ȘI CONTROLUL ROBOȚILOR EDUCAȚIONALI

HURBĂ Cosmin

Conducător științific: Prof. dr. ing. **Adrian NICOLESCU**

REZUMAT: Utilizarea interfețelor pentru recunoașterea gesturilor LEAP MOTION în comanda și controlul roboților educaționali. Utilizând limbajul de programare Java și Kit-ul de programare oferit pentru dispozitivul Leap Motion vom crea o realitate augmentată dintr-un spațiu de lucru virtual al mașinilor. Codul ajută la transformarea semnalului captat de camerele dispozitivului (un punct într-un spațiu 3d poate fi interpretat ca o matrice) în orice linie de cod care poate fi mai departe transformată într-o comandă a unui program.

CUVINTE CHEIE: Controlul Robotului cu mâna utilizatorului.

1 INTRODUCERE

Cercetări în controlul aplicațiilor mecanizate/robotizate prin sisteme de interacțiune cu omul au fost făcute de la începutul implementării sistemelor de calcul moderne.

La momentul actual datorită tehnologiilor mult mai complexe decât în trecut care permit în primul rând miniaturizarea senzorilor, camerelor, dispozitivelor de detecție a mișcării, a sunetului, a vibrațiilor, prezenței, șamd, interacțiunea se realizează aproape fără întârziere. Practic, un utilizator indiferent dacă cunoaște sau nu dispozitivele tehnologice din spatele unui sistem, îl poate utiliza pentru a mișca un robot, de exemplu, în timp real. Orice tehnician poate să mână un robot cu ajutorul unei mânuși sau a unui sistem wireless de detecție a mișcării. Dar secretul se află în spatele dispozitivelor, secretul se află în codul de bază cu ajutorul cărora sunt interpretate toate datele de intrare a senzorilor iar asta se realizează cu ajutorul codului scris în toate programele și sub-rutinele.

Mai departe voi detalia cum se poate controla un robot educațional cu ajutorul mâinii. Pentru asta am folosit un dispozitiv numit Leap Motion și robotul educațional. Aceste două părți hardware le-am „legat” prin intermediul portului USB cu ajutorul unui laptop. Inițial, pentru controlul robotului voi folosi interfața acestuia și controlerul Leap Motion și cu ajutorul programării voi face conexiunea între matricea 3d și robot.

2 STADIUL ACTUAL

În lucrare am urmărit să folosesc un dispozitiv de detecție a mișcărilor și gesturilor pentru a controla un robot de tip braț articulat. Conexiunea robotului la computer este realizată prin intermediul

unei plăci cu circuite integrate. Pentru comunicarea cu programul din Windows se folosește interfața USB. De asemenea, dispozitivul Leap Motion folosește, similar cu placa robotului, interfața USB pentru comunicarea cu calculatorul. Cel mai important aspect al acestei lucrări este programarea dispozitivului Leap Motion astfel încât fiecare mișcare și gest al mâinii să se traducă într-o mișcare a robotului.

Pentru scrierea codului am folosit librăriile Leap Motion. Acestea le-am importat în programul Eclipse pe care l-am folosit pentru scrierea codului și compilarea acestuia. Limbajul de programare folosit este Java. În subcapitole voi detalia toate procesele de conectare a dispozitivelor și un tutorial pentru începerea programării în scopul realizării programelor.

2.1 Prezentarea robotului educațional

Robotul vine în pachet dezamblat. Primul pas este asamblarea lui.

Asamblarea se face foarte ușor urmărind un ghid care este oferit odată cu robotul educațional. Este foarte interesant de observat sistemul de angrenare a mișcării: între motor și axa de rotație există patru roți dințate pentru creșterea cuplului.

Motoarele folosite la angrenarea robotului sunt de tipul Direct Current (DC). Alimentarea lor se face cu ajutorul a 4 baterii tip DR20 care oferă o tensiune de alimentare combinată de 6 Volți. Pentru acționarea motoarelor a fost folosit, în prima fază curentul dat de baterii.

¹ Specializarea Roboți Industriali, Facultatea IMST;

E-mail: cosmin.hurba@gmail.com;

Mai jos aveți o poză cu robotul complet asamblat și o detaliere a sistemului de angrenare cu roți dințate.



Fig. 1. Robotul asamblat

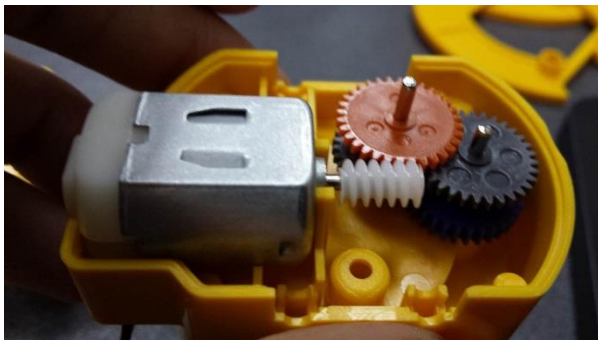


Fig. 2. Sistemul de angrenare

Pe placa conectează motoarele robotului cu computerului regăsim driverul ST1152B pentru acționarea motoarelor. Acest driver oferă la o tensiune de 7 Volți - 1.5 Amperi maxim, suficient pentru greutatea robotului.

De asemenea microcontroler-ul folosit la interfațarea robotului cu computerul prin intermediul interfeței USB este EM78M612.

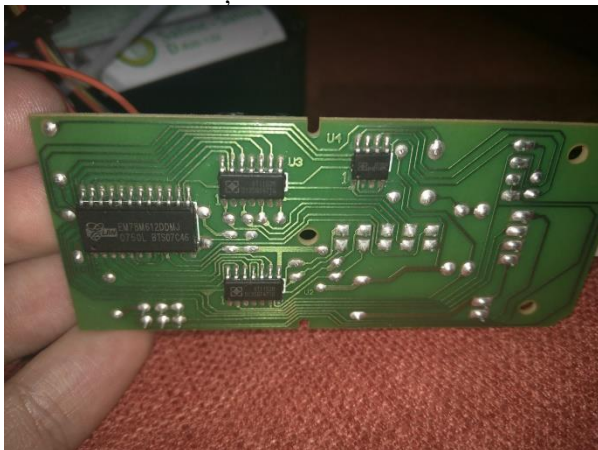


Fig. 3. Circuitele integrate de pe placa USB

2.2 Prezentarea dispozitivului LEAP Motion

Dispozitivul Leap are la bază două camere și trei LED-uri. Acestea urmăresc lumina din spectrul infraroșu, cu o lungime de undă de 850 nanometri, care se află în afara spectrului luminii

vizibile. Sensorii și camera sunt orientați în sus (atunci când dispozitivul se află în orientarea sa normală).

O caracteristică foarte interesantă pe care am observat-o în timp ce mă jucam cu dispozitivul este reglarea dinamică a puterii LED-urilor. Pe măsură ce mișcați mâna mai aproape de senzor, aparatul va estompa automat LED-urile pentru a preveni saturarea termoviziunii și pentru a păstra calitatea înaltă a datelor. LED-urile sunt ușor observabile cu ochiul liber ceea ce înseamnă că funcționează la o lungime de undă în infraroșu în apropierea sau sub 800 nm (exact cât ne este indicat din datele tehnice oferite de constructor).

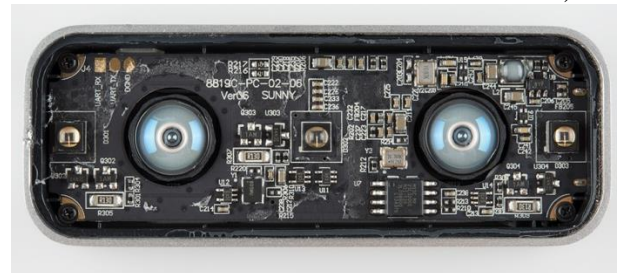


Fig.4 Vedere de sus fără sticla și protecția de cauciuc

Circuitul integrat aflat în dreapta jos, lângă cameră este o memorie flash NOR 32 Mbit într-un pachet de 8 pini SOIC. Memoria flash NOR este folosită mai frecvent în aplicații ca în acest caz în care nu sunt de așteptat foarte multe cicluri de ștergere de program; nivelul de scriere și implicit evitarea de bad-uri care însoțesc de obicei NAND flash-urile pur și simpli ar duce la creșterea costului de producție.

În colțul din stânga sus se pot observa trei puncte RX port-seria, TX și GND. Conectându-mă la ele am reușit să extrag doar un semnal de 3,3Volți. Nu am detectat nici un fel de date în primele secunde de la pornire așa că nu pot spune ce rată de transfer a datelor se folosește.

În colțul din dreapta sus, există un circuit de alimentare. Acest lucru mi-a sărit în ochi, deoarece inductorul din circuit este încastrat în epoxi, ceea ce sugerează că, la un moment dat în ciclul de dezvoltare, inginerii au descoperit că acesta tinde să se dezlipească și s-a luat un pas în plus pentru a remedia acest lucru. O atenție la detaliu care face diferența dintre un produs ieftin și unul de calitate.

2.3 Schema de ansamblu a sistemului

Astfel, sistemul constă din robotul educațional OWI 535, placa de comandă USB prin intermediul computerului și dispozitivul Leap Motion.

Programul realizat în Java traduce mișcarea mâinilor în semnale ce declanșează mișcarea motoarelor robotului.



Fig. 5 Sistemul funcționabil

Mai jos am realizat o schemă de conectare a dispozitivelor atât la nivel fizic cât și detalierea succintă a interconectării sistemelor software.

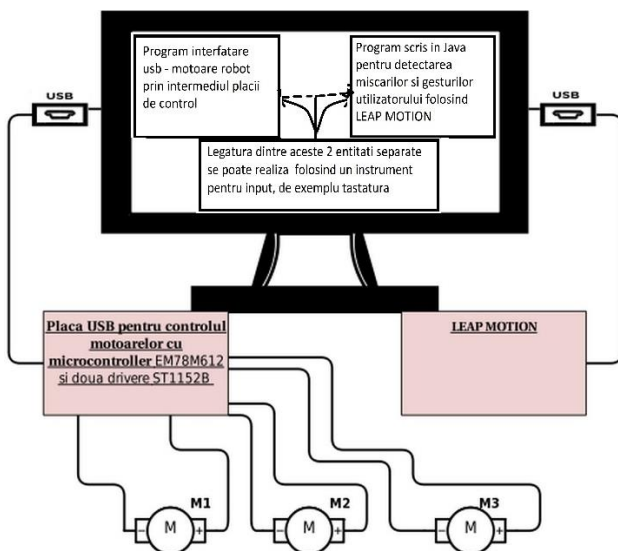


Fig. 6 Schema logica a sistemului

În schema M1, M2, M3 – reprezentarea motoarelor robotului folosite la acționarea gradelor de libertate specifice.

3 INSTALAREA, CONFIGURAREA ȘI PROGRAMAREA DISPOZITIVULUI LEAP MOTION

3.1 Softul Leap Motion

Dispozitivul Leap Motion vine la pachet cu o suită de aplicații gata realizate. Aceste aplicații pot fi downloadate odată ce driverul și softul inițial au fost instalate în sistem.

Dispozitivul este văzut de sistem ca un periferic și poate funcționa în background. Odată instalată, aplicația de detecție este permanent

deschisă și se găsește în toolbar-ul de jos de lângă ceas.

Aplicația care permite începerea interacțiunii dintre dispozitiv și sistem este prezentată sub formă de magazin virtual. În acest magazin se pot găsi programe gata scrise de dezvoltatorii sistemului Leap Motion cât și de utilizatori mai puțin experimentați.

Toate aplicațiile legate realizate pentru a fi folosite în modul „plug & play” pot fi rulate de orice utilizator. Modul în care acestea primesc date de la dispozitiv: aplicațiile celulează în „prim-plan” și sunt scrise cu codul specific dispozitivului primesc automat date de intrare de la serviciul Leap Motion care rulează în fundal în Windows. O aplicație compatibilă cu Leap se poate conecta la serviciul Leap Motion folosind biblioteca nativă Leap Motion.

Pentru a scrie aplicația în C++ sau Java se folosește o bibliotecă care trebuie accesată din una dintre librăriile limbajului disponibil. Aceste librării se găsesc pe site-ul producătorului. În capitolul următor voi detalia tot procesul de începere a programării cât și de scriere a unui program simplu care afișează pe ecran câte mâini au fost detectate, degete, gesturi, etc.

Aplicațiile pot rula fie în „prim-plan”, când este vorba de o aplicație care afișează mâna pe ecran sau îți permite să modifici diferite obiecte 3D cu ajutorul mâinii și a gesturilor, fie pot rula în fundal atunci când datele de intrare se folosesc la controlarea altor programe sau chiar a dispozitivelor de intrare, de exemplu mouse-ul sau tastatura.

În ambele cazuri serviciul Leap rulează în continuu în fundal și furnizează date în timp real indiferent de aplicațiile rulate.

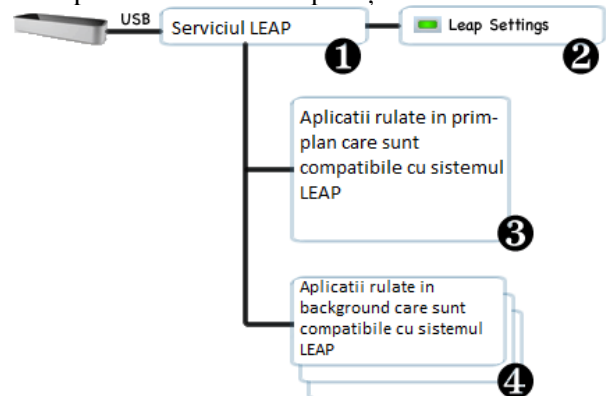


Fig. 7 Schema de funcționare a aplicațiilor native compatibile LEAP

Magazinul virtual oferă o soluție rapidă de a folosi dispozitivul fără a avea cunoștințe de programare, indiferent de aria de utilizare.

După un anumit timp de interacțiune continuă cu dispozitivul am observat că reprezentarea mâinii pe ecran a început să tremure și să se miște haotic. Pentru a rezolva această problemă cei ce au dezvoltat sistemul au implementat o opțiune de calibrare a dispozitivului.

Înainte de lansarea operațiunii de calibrare se poate verifica foarte ușor dacă legăturile dispozitivului cu calculatorul nu sunt optime. De asemenea orice urmă de zgârietură, amprente, ulei sau alte urmede pe sticla dispozitivului poate agrava detecția corectă a poziției mâinii, unghiului sau a degetelor.

Pentru asta, softul ne anunță în timp real prin notificări în windows dacă ceva este în neregulă, de exemplu dacă sticla este murdară – asta poate duce la detecții eronate. Despre starea conexiunii putem verifica și manual intrând în softul dispozitivului.

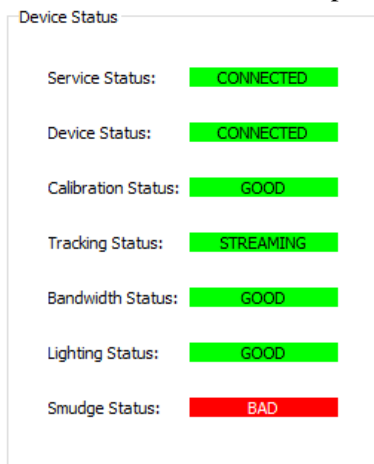


Fig. 8 Stare conexiunii dispozitivului

Am observat că totul funcționa la capacitate optimă și nu erau probleme cu conexiunea între sisteme, totuși, detecția era realizată cu erori. Mișcările nu erau detectate corect și mâna tremura pe ecran.

Procesul de calibrare este diferit de orice altul prin care am mai trecut de-a lungul experienței mele cu dispozitive utilizate la interacțiunea om-mășină.

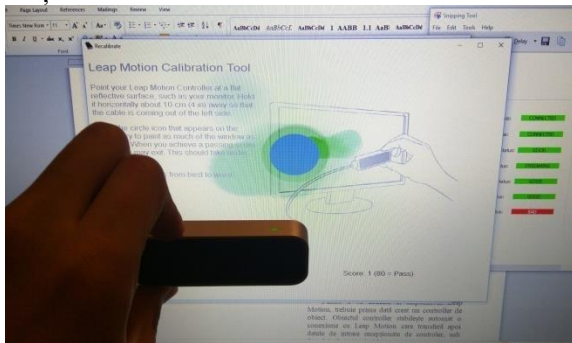


Fig. 9 Procesul de calibrare

Practic, acesta trebuie orientat spre o suprafață lucioasă și folosit ca o pensulă. Astfel, giroscopul și camerele care funcționează cu ajutorul luminii infraroșu sunt calibrate.

3.2 Pregătirea programului de scriere a codului în limbajul de programare JAVA pentru dispozitivul LEAP Motion

Înainte de a începe scrierea oricărui program vă voi trece prin fiecare etapă care, ulterior, cu ajutorul codului mi-a permis să controlez datele de intrare preluate de la dispozitivul Leap Motion.

Pentru scrierea codului cu ajutorul limbajului de programare JAVA am folosit programul Eclipse. Pentru ca acesta să recunoască datele de intrare a dispozitivului trebuie descărcate de pe site-ul producătorului LEAP Motion anumite librării specifice fiecărui limbaj de programare. În cazul de față, pentru JAVA am folosit librăria specifică acestui limbaj.

Librăriile folosite pentru controlul dispozitivului se găsesc pe pagina producătorului <https://developer.leapmotion.com/get-started>.

Este necesară verificarea versiunii Java instalate. Deocamdată dispozitivul funcționează doar cu versiunea java 1.7 sau 1.6. Pentru a verifica versiunea de Java instalată trebuie doar deschis un cmd și tastată comanda: „java – version”.

După instalarea programului Eclipse, pentru a folosi comenzile Leap am urmărit pașii:

- File - New - Java Project
- Înainte de a da next trebuie modificată opțiunea „Use an execution environment JRE” la versiunea JavaSE-1.7

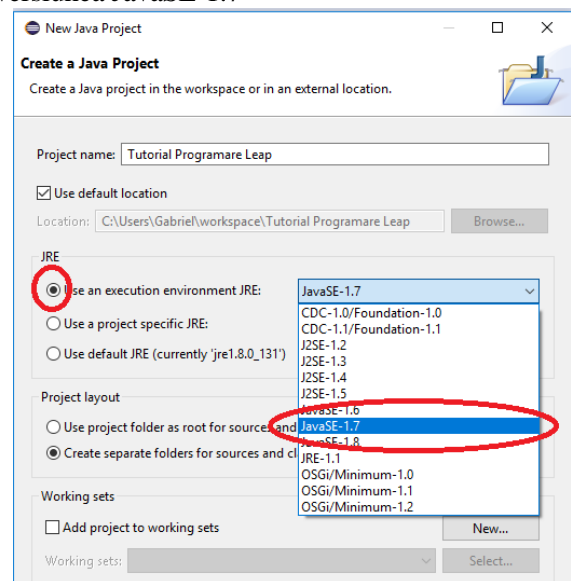


Fig. 10 Schimbarea Execution environment la JavaSE-1.7

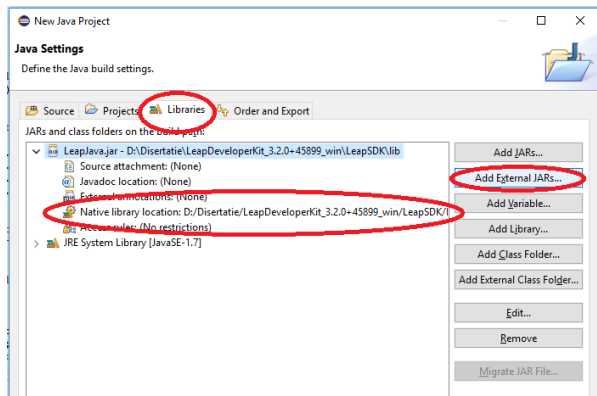


Fig. 11 Adăugarea librăriei externe Leap

- Adăugarea librăriei externe este un pas foarte important
- În tabul Libraries se apasă pe Add External JARs și se alege folderul unde a fost salvat fișierul downloadat de pe site-ul Leap
- Apoi se dă dublu click pe Native library location și se alege locația librăriei

Astea sunt implicațiile instalării programului în așa fel încât să poată detecta dispozitivul Leap și să se poată scrie cod care permite manipularea datelor de intrare.

Mai departe vă voi da un exemplu de cod scris care prelucrează datele de intrare a dispozitivului și realizează diferite acțiuni.

3.3 Detalierea liniilor de cod specifice fiecărei date de intrare de la dispozitivul Leap Motion

Pentru a vă conecta la dispozitivul Leap Motion, trebuie prima dată creat un controller de obiect. Obiectul *controller* stabilește automat o conexiune cu Leap Motion care transferă apoi datele de intrare recepționate de la dispozitiv, sub formă de frame (cadre) la cererea utilizatorului.

Linia de cod care realizează asta este:
`Controller controller = new controller ();`

Funcția `Controller` se utilizează pentru a obține informații despre starea conexiunii și hardware-ul conectat și pentru a seta opțiunile de conectare pentru aplicația proprie.

Următorul pas în programare este Obținerea de frame (cadre) de la dispozitiv.

Prin intermediul funcției `Controller.Frame()` programul primește obiecte de tip „Frame” care conțin date de intrare. Această funcție se poate apela de fiecare dată când aplicația este pregătită pentru a obține cele mai recent set de date generate de dispozitivul Leap Motion.

De asemenea, se poate implementa, un „Listener object” care definește o funcție de apel

invers, iar *controller* va invoca această funcție atunci când un nou cadru de date este gata.

Mai jos aveți un exemplu de program care detectează controllerul și afișează diferite informații:

- `import java.io.IOException;`
- `import com.leapmotion.leap.*;`
- `import com.leapmotion.leap.Gesture.State;`

Aceste linii sunt folosite pentru introducerea în program a librăriilor Leap Motion.

- `class LeapListener extends Listener`
- {

- `public void onInit(Controller controller) {`
- `System.out.println("Program Initializat");}`
- `public void onConnect(Controller controller) {`
- `System.out.println("Program conectat la Leap Motion");`

În aceasta serie de cod am inițializat programul care detectează dispozitivul atunci când acesta este conectat la computer.

- `controller.enableGesture(Gesture.Type.TYPE_SWIPE);`
- `controller.enableGesture(Gesture.Type.TYPE_CIRCLE);`
- `controller.enableGesture(Gesture.Type.TYPE_SCREEN_TAP);`
- `controller.enableGesture(Gesture.Type.TYPE_KEY_TAP);`
- }

- `public void onDisconnect(Controller controller) {`
- `System.out.println("LEAP Motion a fost deconectat");`
- }

- `public void onExit(Controller controller) {`
- `System.out.println("Exited");`
- }

- `public class LeapController {`
- `public static void main(String[] args) {`
- `//TODO Auto-generated method stub`

- `LeapListener listener = new LeapListener();`
- `Controller controller = new Controller();`

- `controller.addListener(listener);`
- `System.out.println("apasă Enter pentru a iesi");`

```

- try {
-     System.in.read();
-     } catch (IOException e) {
-         e.printStackTrace();
-     }
-     controller.removeListener(listener);
-     }
- }

```

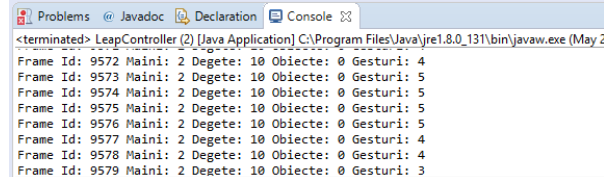
Un exemplu de program simplu care afișează pe ecran câte mâini sunt detectate, degete, obiecte și gesturi.

Mai departe acestea pot fi folosite ca date de intrare în programul ce controlează robotul.

```

29 public void onFrame(Controller controller) {
30     Frame frame = controller.frame();
31     System.out.println("Frame Id: " + frame.id()
32         /* "Time: " + frame.timestamp()
33         + " Maini: " + frame.hands().count()
34         + " Degete: " + frame.fingers().count()
35         + " Obiecte: " + frame.tools().count()
36         + " Gesturi: " + frame.gestures().count() );
37     }

```



```

<terminated> LeapController (2) [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (May 2
Frame Id: 9572 Maini: 2 Degete: 10 Obiecte: 0 Gesturi: 4
Frame Id: 9573 Maini: 2 Degete: 10 Obiecte: 0 Gesturi: 5
Frame Id: 9574 Maini: 2 Degete: 10 Obiecte: 0 Gesturi: 5
Frame Id: 9575 Maini: 2 Degete: 10 Obiecte: 0 Gesturi: 5
Frame Id: 9576 Maini: 2 Degete: 10 Obiecte: 0 Gesturi: 5
Frame Id: 9577 Maini: 2 Degete: 10 Obiecte: 0 Gesturi: 4
Frame Id: 9578 Maini: 2 Degete: 10 Obiecte: 0 Gesturi: 4
Frame Id: 9579 Maini: 2 Degete: 10 Obiecte: 0 Gesturi: 3

```

Fig. 12 Adăugarea bibliotecii externe Leap

Pentru programul de control al robotului, codul folosit a fost mai complex. Baza este ceea ce am scris mai sus. Important este să se ajunga într-un punct în care pot fi preluate datele de intrare de la dispozitiv. Apoi prelucrarea datelor pentru a le transforma în funcții ține de programarea de bază.

Aici am folosit un gest pentru mișcarea robotului pe axa Z

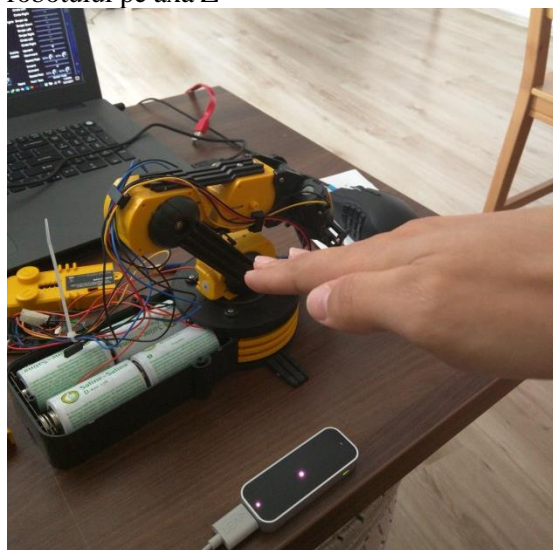


Fig. 13 Mișcarea pe axa Z

Un fragment de cod folosit la preluarea mișcării și trimiterea lor către motoare:

```

for (Hand hand : frame.hands()) {String handType
= hand.isLeft() ? "left hand" : "right hand"
System.out.println(handType + " " + " ", id: " +
hand.id()+ " ", Palm Position: " +
hand.palmPosition()) Vector normal =
hand.palmNormal();
Vector direction = hand.direction();
System.out.println("Pitch: " +
Math.toDegrees(direction.pitch())
+ "Roll: " +
Math.toDegrees(normal.roll())
+ "Yaw: " +
Math.toDegrees(direction.yaw()) );

```

Acesta preia datele legate de unghiul palmei.

În arduino, pentru traducerea datelor preluate din programul Leap am folosit `m1.attach(10);` - am atașat motorul 1 de mișcarea pe axa z (Yaw);

```

if(readstring.length() >0) {
StringY=readstring.substring(0,readstring.indexOf
('Y')); - pentru citirea poziției pe axa z (Yaw)
char carray[Y.length() + 1]; - definește un nou
string pentru data Y
Y.toCharArray(carray,sizeof(carray)); -
inițializează stringul Y
int yv = atoi(carray); - convertește din string în int
pentru a putea fi scris către funcția motor
m1.write(yv); - scrie în funcția int pentru a-l misca

```

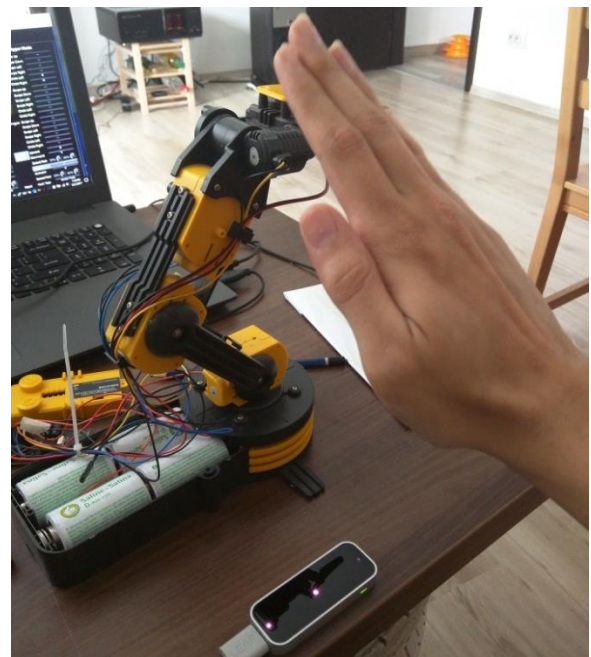


Fig. 13 Mișcarea în sus a robotului de-a lungul axei Z

4 CONCLUZII

Robotul educațional, așa cum vine el la pachet poate fi folosit foarte ușor prin interfața USB. Se poate folosi mouse-ul sau tastatura pentru mișcarea fiecărui motor. În schimb, dacă vrem să realizăm o formulă de control cu ajutorul unui alt dispozitiv implicațiile sunt mai complexe.

Aceste implicații vin sub forma codului care trebuie scris în diferite limbaje de programare. Pentru înțelegerea codului și modificarea lui este nevoie de multă practică (minim un an de scriere de cod).

Prin încercări și eșecuri am dus lucrarea într-un punct în care pot să spun că robotul este controlabil cu ajutorul device-ului Leap Motion. Dar dacă este să vorbim despre precizia mișcării sau calitatea la care aceasta se realizează, intrăm într-o arie a programării mult mai vastă și pentru asta este nevoie ca motoarele robotului să fie de tip stepper pentru a putea prelua în timp real și date despre poziția exactă a unghiului fiecărei axe a robotului.

5 MULȚUMIRI

Vreau să mulțumesc profesorului îndrumător Adrian NICOLESCU pentru suportul acordat în cadrul realizării fizice a lucrării. În același fel aș dori să mulțumesc doamnei Cezara AVRAM pentru disponibilitatea de care a dat dovadă, de fiecare dată când a fost nevoie.

6 BIBLIOGRAFIE

<https://developer.leapmotion.com/documentation/java/api/gen->

[java/classcom_1_1leapmotion_1_1leap_1_1_hand.html](https://developer.leapmotion.com/documentation/java/classcom_1_1leapmotion_1_1leap_1_1_hand.html) (data accesării 15.05.2017)

<https://infoacademy.net/>(data accesării 15.05.2017)

<https://developer.leapmotion.com/>(data accesării 15.05.2017)

<https://learn.sparkfun.com/tutorials/leap-motion-teardown>(data accesării 15.05.2017)