BEZDOTYKOVÉ OVLÁDÁNÍ SERVOMOTORŮ

Ing. Filip Hrdlička[1]
[1] Západočeská univerzita v Plzni, Fakulta strojní, Katedra konstruování strojů,
ficeriov@kks.zcu.cz

CONTACTLESS CONTROL OF SERVOMOTORS

Abstract: *With a combination of LEAP sensor and Arduino board is possible to recognize different gestures of human hands and use it as output signals for control.*

Key words: *servomotor, control, LEAP, Arduino*

ÚVOD

LEAP motion je optický senzor skládající se ze 2 kamer a 3 infračervených LED senzorů schopný snímat prostor kolem sebe. Snímaný prostor je definován úhly 150° podélně a 120° příčně k zařízení. Díky těmto vlastnostem dokáže LEAP motion snímat polohy, natočení a počet rukou i prstů. To vše i v horších světelných podmínkách. [1] Možnosti tohoto senzoru využijeme v kombinaci s Arduinem pro ovládání servomotoru.

PROPOJENÍ ZAŘÍZENÍ

Způsobů, jak propojit LEAP motion s Arduinem je několik. Tento článek se bude zabývat propojením pomocí softwaru Processing, který je volně k dispozici. Processing využívá jazyku Java a knihoven LEAP motion, které do programovacího jazyka přidávají jednotlivé příkazy pro gesta rozpoznané senzorem. Z vytvořeného kódu jsou pak vybraná data exportována přímo do Arduina.

PROCESSING

Jak je vidno z Obr. 1, v prvním kroku je třeba vložit do programu knihovnu LEAP motion a knihovnu komunikace. V následujícím kroku je třeba nadefinovat komunikační port a samotné záznamové (ovládací) zařízení. Dále program pokračuje nadefinováním proměnných. V tomto případě jsou nadefinovány 4 proměnné **hand_x**, **hand_y**, **sa_1** a **sa_2**. Proměnné hand_x a hand_y budou obsahovat souřadnice polohy ruky v těchto dvou na sebe kolmých osách. Proměnné sa_1 a sa_2 pak budou obsahovat úhly servomotorů.

Program pokračuje nastavením zobrazovacího okna, v tomto případě ve velikosti 800x600 px, nastavením inicializace ovládacího zařízení a hlavně nastavením výstupního portu, přes který

bude Processing komunikovat s Arduinem. V tomto případě je tak nutné mít v kódu Processingu nastavené stejné číslo portu, jaké je přiřazeno k Arduino desce, v našem případě port COM3. [2]

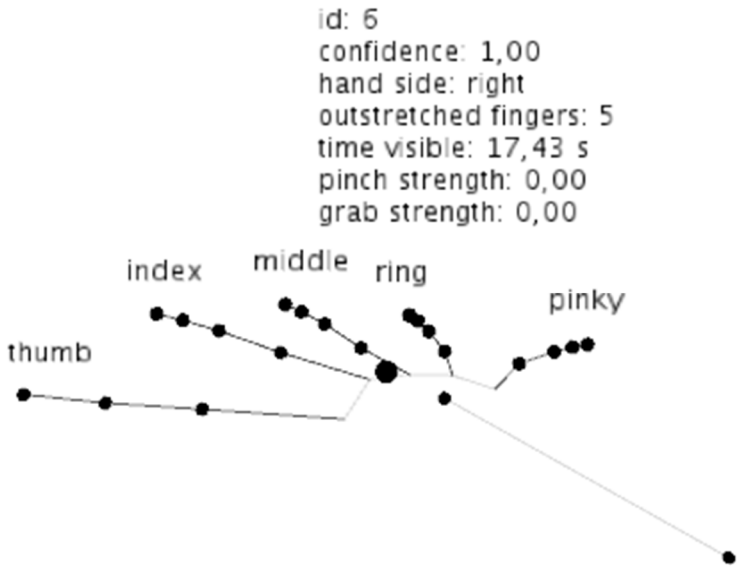
```
import processing.serial.*;                    //knihovna leap motion
import de.voidplus.leapmotion.*;               //knihovna serial communication

Serial port;                                   //definování portu
LeapMotion leap;                               //definování ovládacího zařízení (LEAP)

float hand_x;                                  //definování proměnných
float hand_y;
float sa_1;
float sa_2;


void setup() {                                 //Nastavení zobrazovacího okna
  size(800, 600, P3D);
  noStroke();
  fill(50);

  port = new Serial(this, "COM3", 9600);       //Inicializace portu (vstupní port na Arduinu)
  leap = new LeapMotion(this);                 //Inicializace ovládacího zařízení (LEAP)
  }


void draw() {                                  //Začátek kódu - vykreslení rukou
  background(255);                             //Barva pozadí vykreslování

  for (Hand hand : leap.getHands()) {
    hand.draw();
    PVector  hand_position = hand.getPosition();  //Příkaz pro získání pozice ruky

    hand_x = (hand_position.z-50)*12;          //Kalibrace os do správných směrů a do středu senzoru
    hand_y = hand_position.x-380;


    //Servo_1
    if (hand_x>-200 && hand_x<200) {           //Nastavení maximálních limitů snímání v ose x
      sa_1 = map(hand_x, -200, 200, 0, 180);   //Rozmapování max. limitů do stupňů serva
    }

    //Servo_2
    if (hand_y>-200 && hand_y<200) {           //Nastavení maximálních limitů snímání v ose y
      sa_2 = map(hand_y, -200, 200, 0, 180);   //Rozmapování max. limitů do stupňů serva
    }


    println(" x = " + nf(hand_x, 3, 1), " y = " + nf(hand_y, 3, 1),   //Vypsání souřadnic a úhlů serv do stavového řádku
            " ax = " + nf(sa_1, 3, 1), " ay = " + nf(sa_2, 3, 1));

    byte[] q = {byte(sa_1), byte(sa_2)};                              //Načtení výsledných dat do bufferu
    port.write(q);                                                    //Poslání dat do Arduina
  }
}
```

***Obr. 1** Processing kód*

Prostřední část kódu obstarává vykreslení samotné ruky (či rukou) do zobrazovacího okna. Děje se tak za pomoci příkazu **Hand hand : leap.getHands ()** a **hand.draw ()** [2]. Processing následně vykreslí ruku (viz Obr. 2) a to s údaji jako je počet neskrčených prstů, typ ruky či zda dochází k úchopu. Po této části kódu zbývá zadat, které informace chceme získat. V našem případě se jedná pouze o pozici ruky v osách **x** a **y** získané pomocí příkazu **Pvector hand_position** [2]. Takto získané souřadnice je pak potřeba ještě zkalibrovat, aby při pozici ruky přímo nad senzorem obě osy ukazovaly shodně nulu.

Předposlední část kódu se zabývá převedením souřadnic z os na natočení servomotorů. K tomuto je použita funkce **map**, která rovnoměrně rozkouskuje souřadnice z minima do maxima. V tomto případě se jedná o rozsah v osách -200 až 200 převedený na maximální potenciál serv, tj. úhel od 0° do 180°, přičemž nulová souřadnice v ose znamená natočení serva do úhlu 90°.

Poslední část kódu již zobrazuje pouze vypsání všech proměnných na stavový řádek Processingu kvůli kontrole, načtení aktuálních úhlů serv (sa_1, sa_2) a jejich export skrze komunikační port do Arduina.



Obr. 2 *Vizualizace snímané ruky*

ARDUINO

První část kódu pro Arduino je velmi jednoduchá, pojednává o nastavení servomotorů, pro které lze najít nespočet návodů v různých zdrojích. Z tohoto důvodu tuto část kódu zde nebudeme více rozebírat.

Důležitá část kódu začíná definováním smyčky (void loop), kde dochází k importu zaslaných dat z Processingu. Data získáváme ze sériového bufferu, kde číslo v závorce představuje pořadí poslední proměnné. Pořadí je počítáno od nuly, proto je v závorce uvedena jednička, ačkoliv proměnné máme dvě. V následujícím řádku pak již samotné proměnné načítáme. Zde je nutné uvést počet načítaných proměnných, proto je v závorce dvojka.

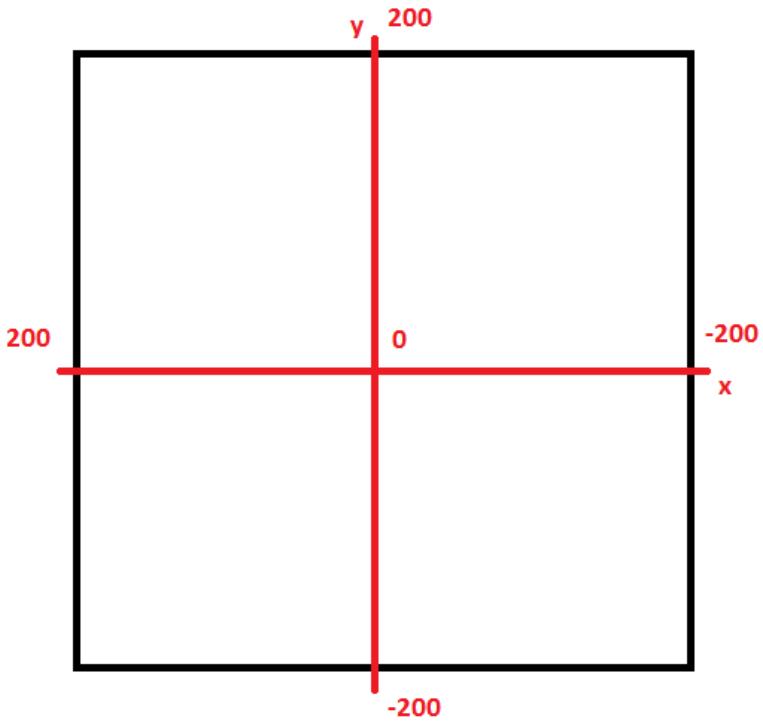
Poslední část kódu, před zapsáním výsledných úhlů získaných z Processingu do serv, je korekce úhlů (získaných dat). Serva zde mají problém s přechodem přes úhel 129°, proto je potřeba tuto chybu zkorigovat soustavou podmínek. Po takto provedené korekci se serva již budou pohybovat plynule od 0° do 180° a zpět v závislosti na pozici ruky v souřadnicích od -200 do 200.

```
#include <Servo.h>

Servo servo_1;
Servo servo_2;

int sp_1 = 5;
int sp_2 = 6;
int sa_1;
int sa_2;


void setup(){
  Serial.begin(9600);
  servo_1.attach(sp_1);
  servo_2.attach(sp_2);
  servo_1.write(90);                    //Nastavení serv do výchozí pozice 90°
  servo_2.write(90);
}

void loop(){
  if(Serial.available()){
    char buffer[1];                     //Počet proměnných v bufferu (0,1)
    Serial.readBytes(buffer,2);         //Počet proměnných bufferu

    //Servo_1
    int sa_1 = buffer[0];               //Přiřazení dat z bufferu
    if (sa_1 <= -76 && sa_1 > -129){    //Korekce úhlu serva
      sa_1 = 256-abs(sa_1);
    }
    else if (sa_1 > -76 && sa_1 < 0){
      sa_1 = 179;
    }
    else{
      sa_1 = buffer[0];
    }

    //Servo_2
    int sa_2 = buffer[1];
    if (sa_2 <= -76 && sa_2 > -129){
      sa_2 = 256-abs(sa_2);
    }
    else if (sa_2 > -76 && sa_2 < 0){
      sa_2 = 179;
    }
    else{
      sa_2 = buffer[1];
    }

    servo_1.write(sa_1);
    servo_2.write(sa_2);
  }

  Serial.flush();
}
```

Obr. 3 *Arduino kód*

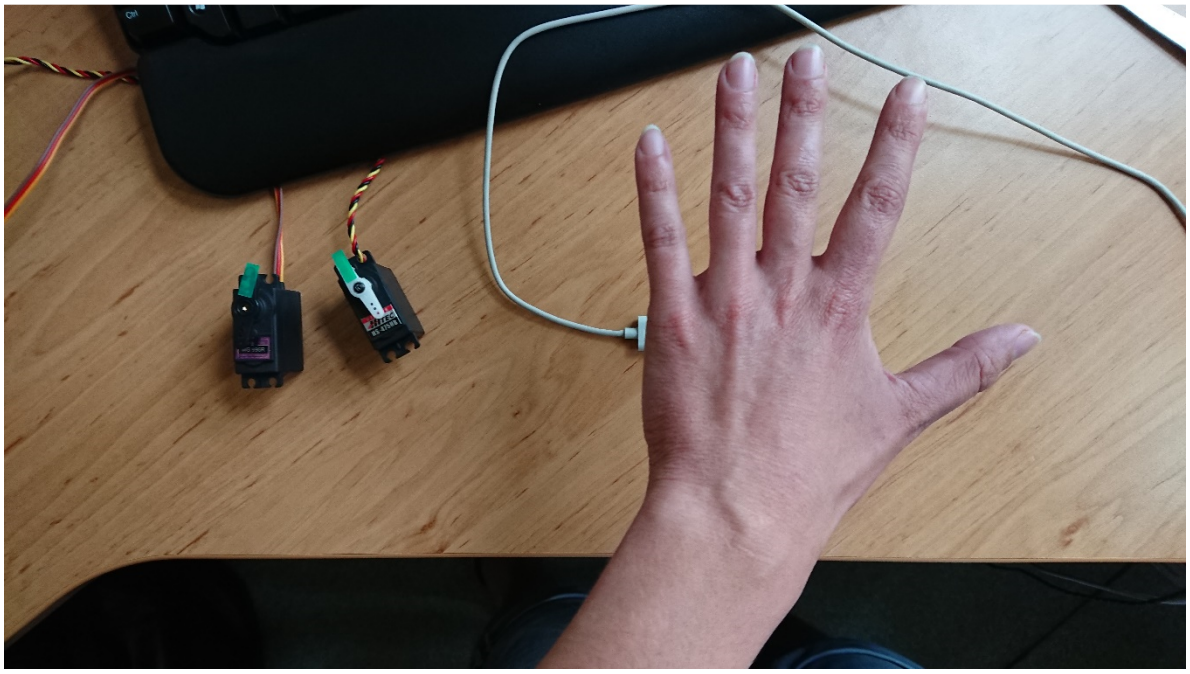
VÝSLEDKY

V processingu jsme vytvořili virtuální snímací prostor ve tvaru čtverce o velikosti 400x400 (viz Obr. 4), kde jeho střed je umístěn přímo nad senzorem LEAP. K maximům a minimům tohoto čtverce byly přiřazeny maxima a minima serv. Jedno servo pro osu x, druhé pro osu y. Maximální záporné souřadnice odpovídají 0° na servech, maximální kladné pak 180°.

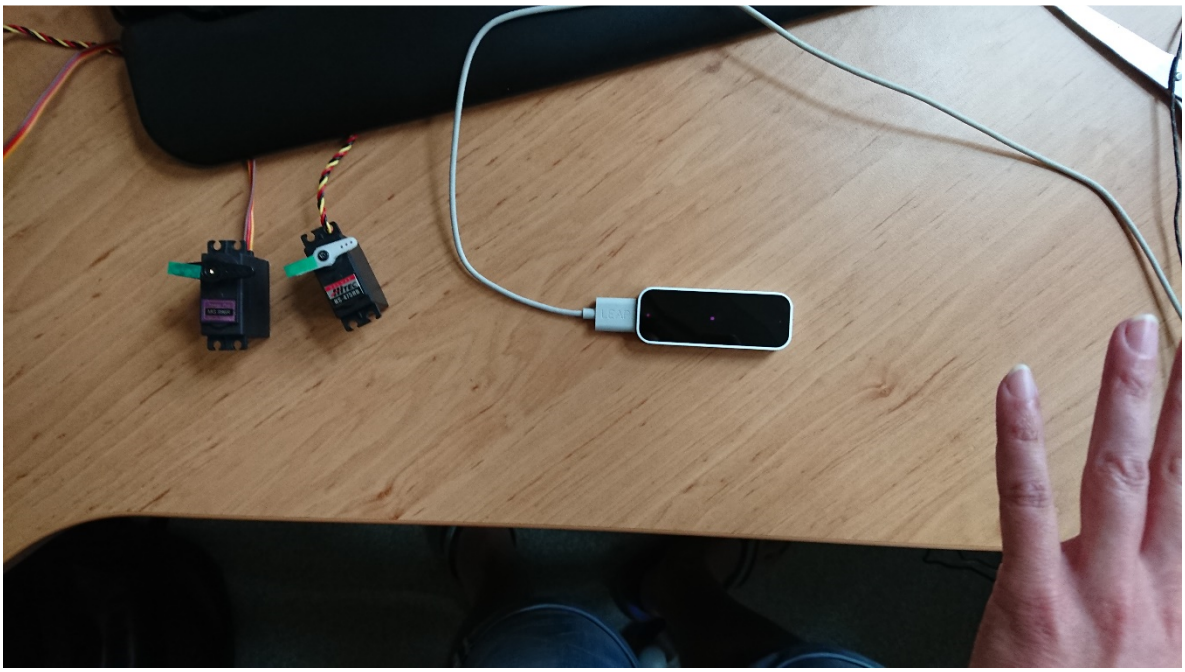


Obr. 4 *Snímací prostor*

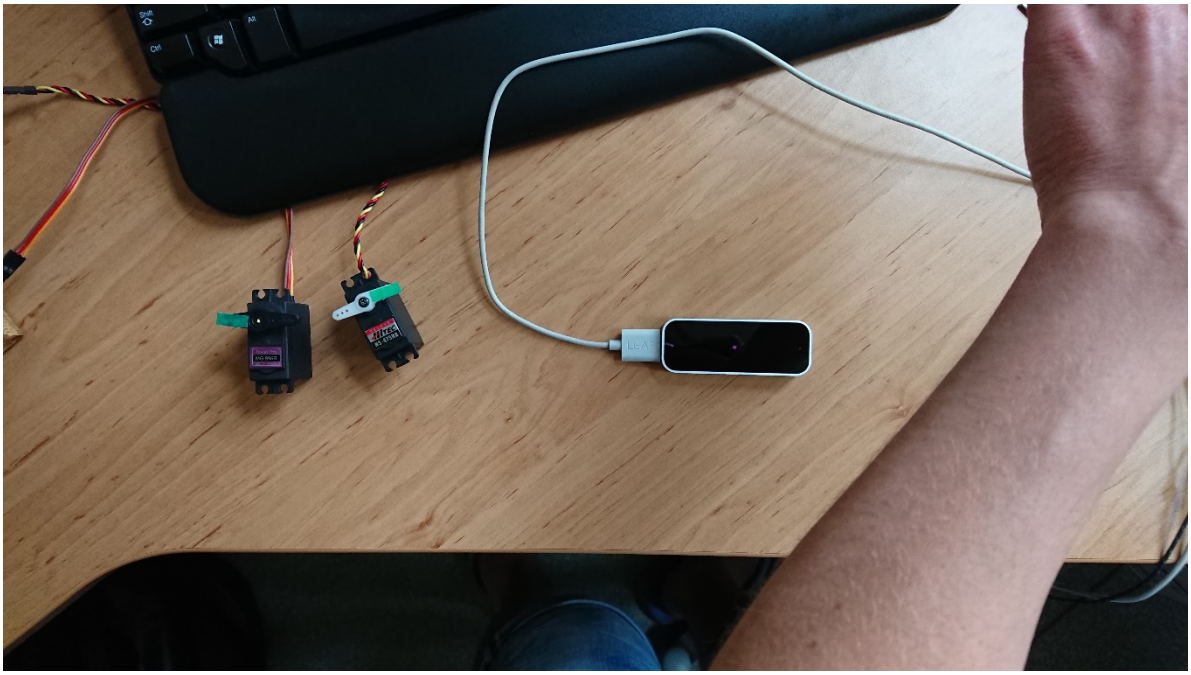
Na následujících fotografiích lze vidět pohyb serv na základě pozice ruky. Pokud je zelená část natočená vlevo kolmo k servu, znamená to úhel serva roven 0°, obdobně pak pravá strana představuje 180°.



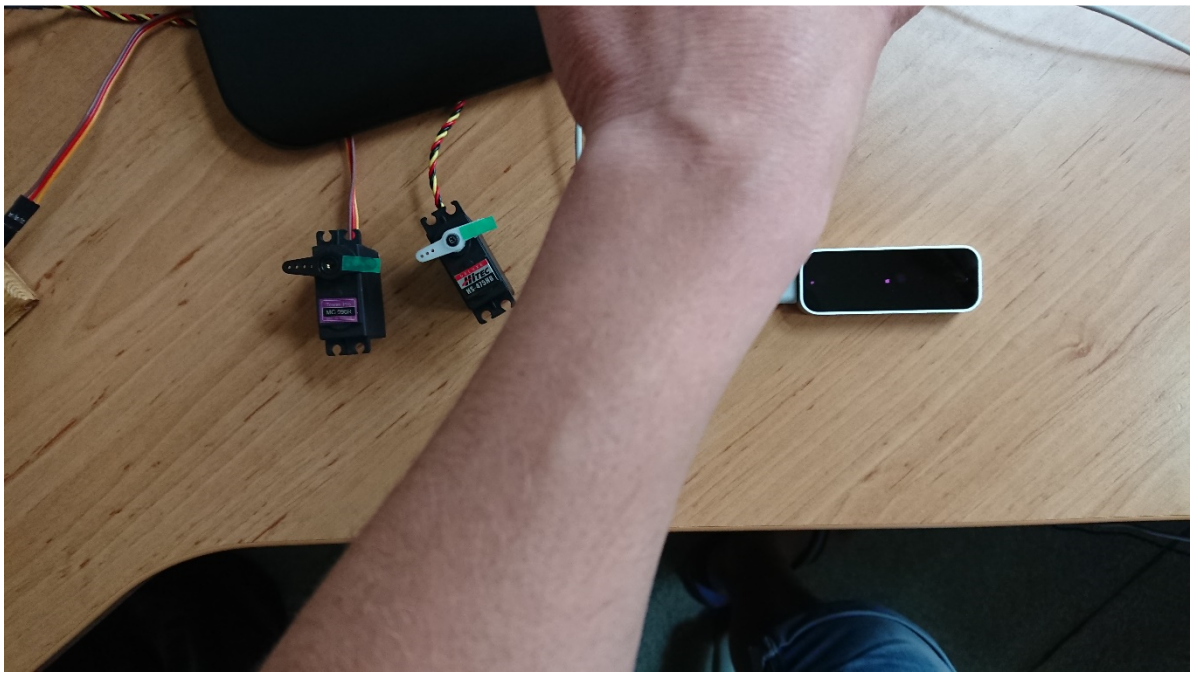
Obr. 5 Pozice ruky přímo nad senzorem, úhel obou serv je roven 90°



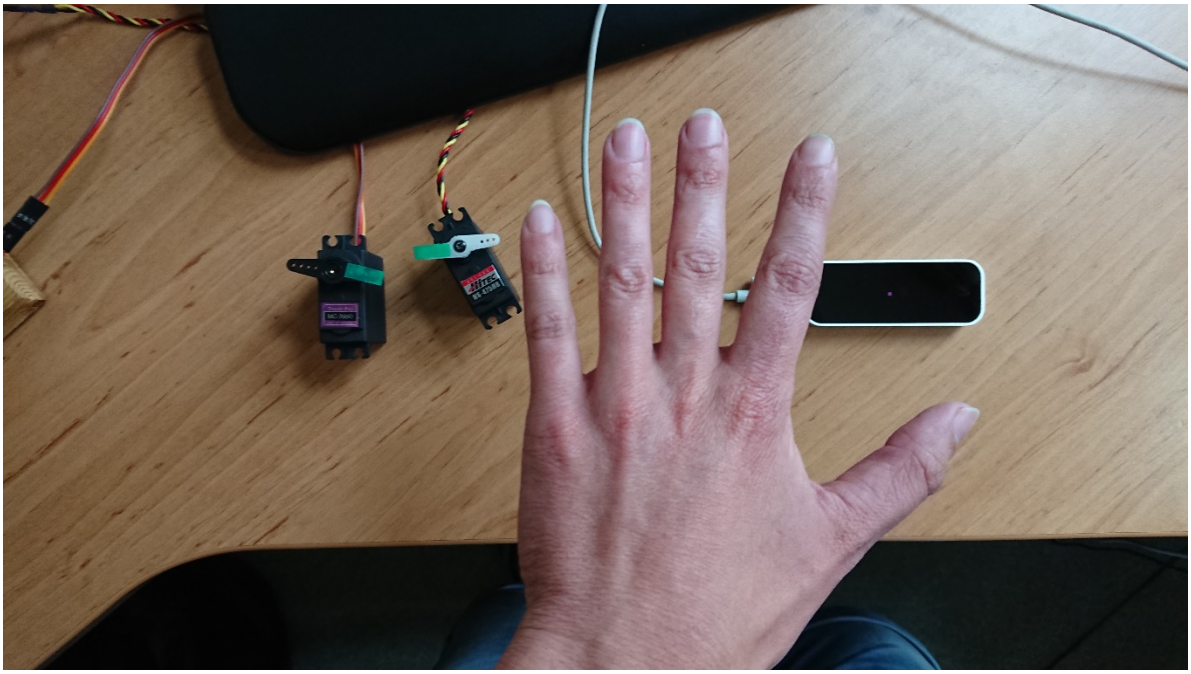
Obr. 6 *Pozice ruky -200, -200, obě serva jsou v nule*



Obr. 7 *Pozice ruky -200, 200*



Obr. 8 *Pozice ruky 200, 200, obě serva jsou ve 180°*



Obr. 9 *Pozice ruky 200, -200*

ZÁVĚR

Díky propojení Processingu s Arduinem je možné převádět jakákoliv gesta ruky na řídící signály, s kterými lze kromě serv ovládat prakticky cokoliv, od robotických ramen počínaje po drony konče. V tomto příspěvku byl ukázán pouze jeden příkaz gesta, avšak LEAP motion jich podporuje hned několik, díky čemuž je možné dostat z jedné ruky hned několik řídících signálů.

LITERATURA

[1] COLGAN, Alex. *How does the Leap Motion controller works* [online]. 2014, [cit. 9.8.2019]. Dostupné z: http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/

[2] NANDY, Abhishek. *Leap motion for developers* [online]. 2016, s. 108-109 [cit. 13.8.2019]. Dostupné z: https://books.google.cz/books?id=th61DQAAQBAJ&pg=PA109&lpg=PA109&dq=leap+motion+pvector&source=bl&ots=o2FoYhqSTN&sig=ACfU3U2jlMe_9ES7Dlxs20qIvBJHtdCMnQ&hl=cs&sa=X&ved=2ahUKEwj77LHd9IXkAhUP16QKHcIZCCIQ6AEwB3oECAkQAQ#v=onepage&q=leap%20motion%20pvector&f=false

Poděkování

Tento příspěvek vznikl za podpory projektu SGS-2019-001 "Komplexní podpora konstruování technických zařízení III"