SŠPU
STŘEDNÍ ŠKOLA PRŮMYSLOVÁ
A UMĚLECKÁ, OPAVA

ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Robotické rameno pohybující se pomocí gest

Autor: Sofja Klopcová

Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování

Třída: IT4

Školní rok: 2023/24

Poděkování

Chci poděkovat Pánu učiteli Godovskému za poskytnutí veškerého hardweru a také konzultací během celé práci.

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracovala samostatně a uvedla veškeré použité informační zdroje.
Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2024

.....
Podpis autora

Abstrakt

Výsledkem projektu je funkčí pohyblivé rameno, které zvláda chytnout kuličku z konce stavebnice a posune ji na začátek a také za dobře odvedenou práci se uklonit. Ovládáne je gesty které jsou snímány kamerou notebooku. Hlavním aspektem projektu je kontrola jednotlivých serv ramene pomocí gest. Kód který umožní zobrazovat gesta je psaný v Pythonu, a holá pohyblivost ramene je obstarána Arduinem které využívá C++.

Klíčová slova

poghyblivé rameno, gesta, Python, C++ ...

Abstract

Write your abstract here! Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Keywords

Template, LaTeX, High school proffessional activity, ...

Obsah

Úvod

Tahle dokumentace vysvětluje celkovou funkčnost mého projektu. Kdyý jsme byli seznámení s tím, že čtvrtý ročník musíme zakončit ročnikovým projektem, byla jsem už rozhodnutá, že se můj projekt bude zabývat spíše Hardwarem. Po dlouhém a náročném pátrání a s pomocí pana Godovského jsem si pro svůj projekt nakonec vybrala robotické rameno. Bohužel samotné ovládání mě zavedlo primárně do Softwaru.

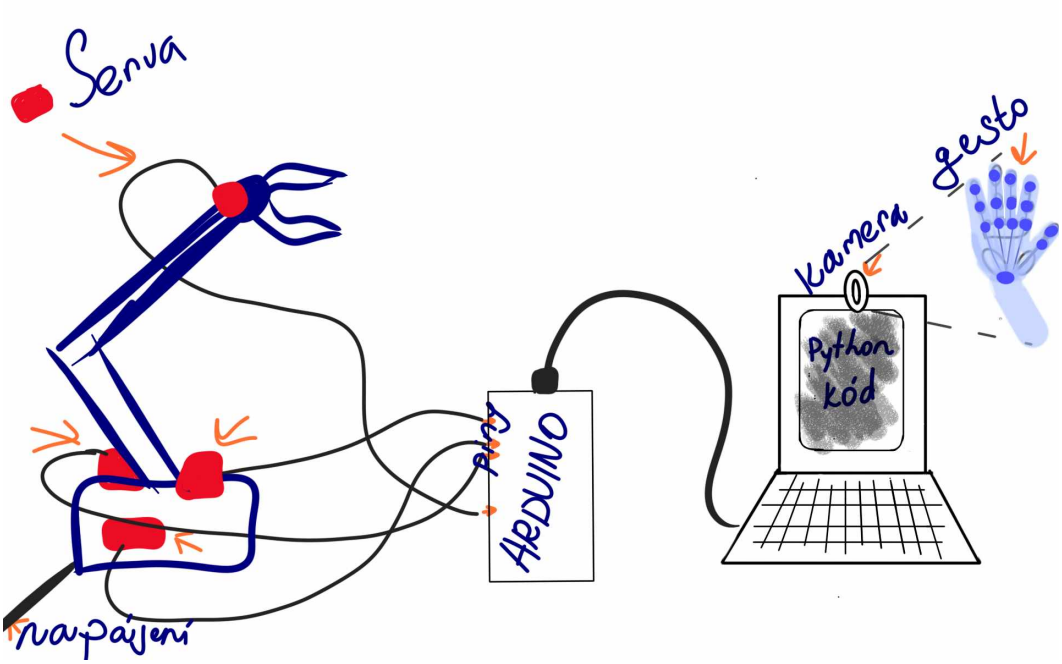
Rameno je ovládáné předvolenými gesty snímanými z kamery notebooku. Byl také nápad kompletně ovládat rameno způsobem, že rameno kopíruje pohyb ruky. Ale k tomu bylo potřeba dokoupit Pohyblivý senzor Kinect pro Xbox. To je ale celkem finančně náročné, tudíž od toho jsem odstoupila.

Cílem projektu bylo zprovoznit pohyb ramene celkem jednoduchým ale také efektivním způsobem. Diky tomu jsem si skvěle procvičila Python a C++. Motivoval mě ten pocit že to rameno rozpohýbuji.

Hned v první kapitole naleznete všechny potřebné součástky k sestavení samotného ramene. To jsem ale nesestavovala já, tudíž se budu věnovat spíš softwaru. Je to doprovázené krátkou teorii o využitých technologiích. Nakonec také samozřejmě nebudou chybět odkazy ne všechny využité zdroje.

1 Princip fungování projektu

Tady se nachází strukturovaný náčrt, jak funguje celý projekt. A to tak že kamera snímá určité gesto. Kód s využitím knihoven obraz převádí na potřebné hodnoty které rozpozná rameno. Potom se pošlou do Arduina a ono ve své řadě je dá na určité piny aby serva rozpoznala co mají dělat.



Obrázek 1.1: Princip fungování projektu.

2 HARDWARE

2.1 NETIŠTĚNÉ SOUČÁSTKY

Jelikož jsem to rameno nesestavovala tak nemam možnost popisovat postup, ale tady je výčet potřebných součástek

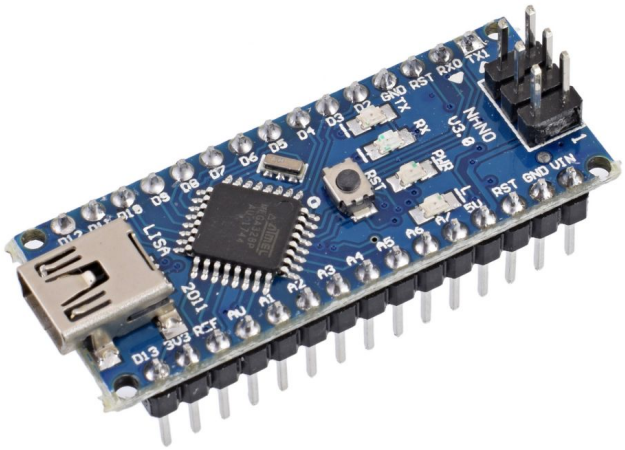
- 3 - 955 nebo 946 servo
- 1 - SG90 SERVO
- 1 - Samojistná matice M6
- 1 - Šroub M6x25
- 2 - Samojistné matice M3
- 2 - Šrouby M3 x 20
- 1 - Šroub M3 x 10 se šestihrannou hlavou
- 9 - Samojistné matice M4
- 1 - Šroub M4 x 40
- 1 - Šroub M4 x 30
- 5 - Šroub M4 x 20
- 1 - Závitová tyč M4 x 60mm
- 1 - Závitová tyč M4 x 32 mm
- 25 - Kulové koule o průměru 6 mm
- Ložisko 1 - 606zz
- Některé podložky M4

Návod na samotné sestavení ramene naleznete na tomhle odkazu

2.1.1 Arduino nano 328

popis

Arduino napomáha rozhýbat samotný hardwere. Napsaný kód, který popíšu v další kapitole nahrajeme pomocí USB kabelu do něj a diky správnému zapojení ramene vše skvěle funguje.

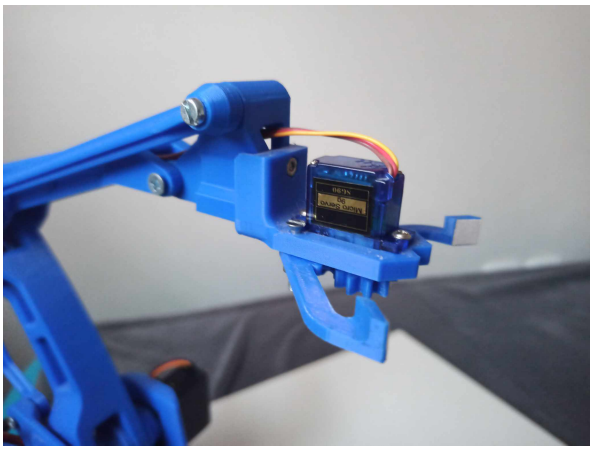


Obrázek 2.1: Arduino nano 328

2.1.2 Servo

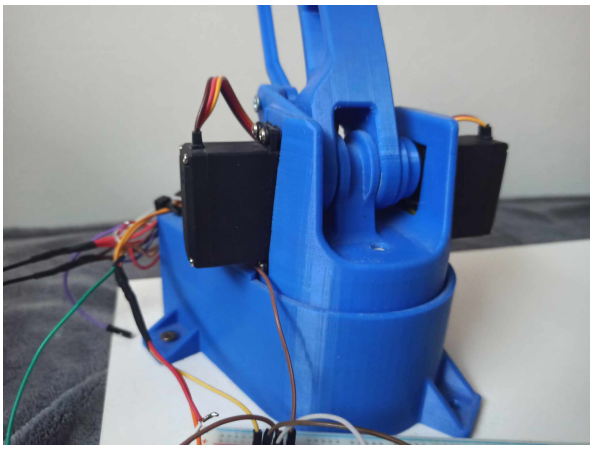
popis

Tohle rameno obsahuje tři velké Serva a jedno malé.Každý z nich má také tři výstupy, a to: kabel určený na nějaký port, mínus a plus.
To nejmenší se nachází v kleštičkách



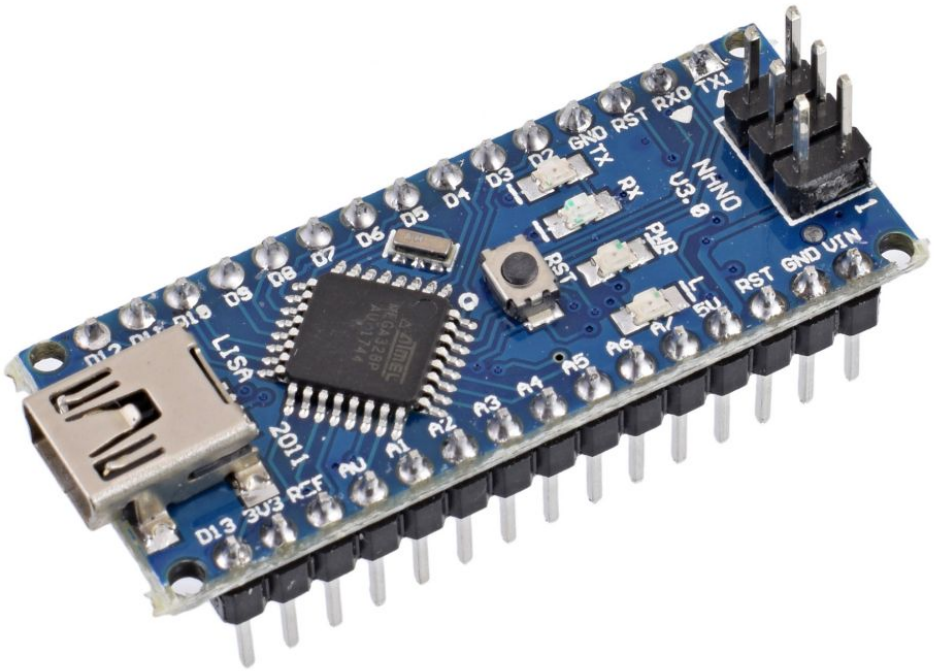
Obrázek 2.2: Servo nacházejíci se v kleštich ramene[**?**].

Zbylé tři se zabývají rotací a pohyby nahoru-dolů, dopředu-dozadu



Obrázek 2.3: Serva v těle ramene.

2.1.3 Obvod



Obrázek 2.4: Zapojení ramene.

3 SOFTWARE

3.1 ZÍSKÁNÍ GEST

Začneme importem knihoven, a to **OpenCV math** a **mediapipe 0.10.8**

Knihovna OpenCV

OpenCV je multiplatformní svobodná knihovna pro práci s obrazem.

Samotná knihovna je z velké části napsána v C++, ale poskytuje Python wrapper, díky kterému ji můžeme používat i v Pythonu. OpenCV se používá pro zpracování obrazu z kamer, rozpoznání psaného textu nebo obličejů.. Tuto knihovnu jsem využila na celkové zpracování obrazu z kamery mého notebooku

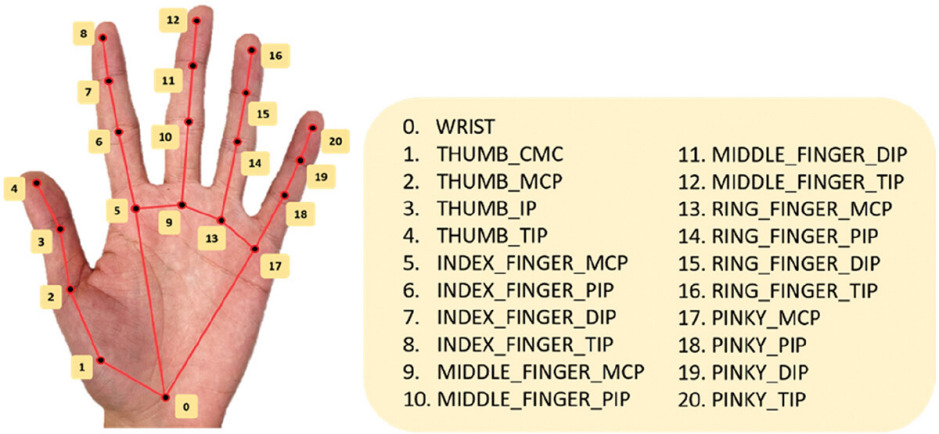
```
1 cap = cv2.VideoCapture(0)
2 ...
3 image = cap.read()
4 imageRGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
5 results = hands.process(imageRGB)
```

Kód 3.1: Obraz, a jeho převedení do správného formátu

Převede snímáný obraz do formátu RGB, který potřebujem pro vykreslení pixelů ve správném pořadí.

Knihovna MediaPipe

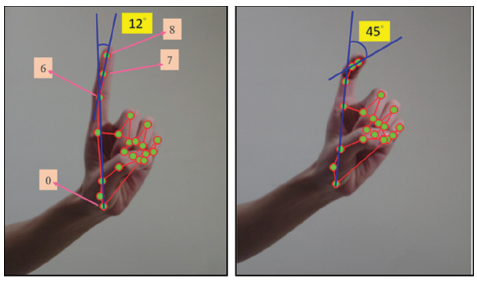
Tahle knihovna se zaměřuje na detekci lidských části těla. Já využila pro svůj projekt pouze detekci dlaní. Pro detekci potřebuje mít orientační body ruky. Model ručního orientačního bodu dokáže předpovědět 21 přesných souřadnic umístění každého orientačního bodu na ruce.



Obrázek 3.1: Orientacni body dlaně.

Knihovna math

Ve spolupráci se zvyše zmíněnou knihovnou mi poskytla souřadnice jednotlivých kloubu, pro přesné zachycení ruky. Další výpočty provedla také při zachycení vzdálenosti prstů potřebné k rozhýbání kleštiček



Obrázek 3.2: Použití úhlu v gestech.

Obrázek 3.3: Použití měření vzdálenosti v gestech.

3.2 OVĚŘENÍ FUNKČNOSTNÍ RAMENE

Než jsem mohla začit se sprovozněním ramene dle myšlenky, jsem musela ověřit jejích funkčnost a taky zjistit jak se vlastně jednotlivá část hýbe. Proto jsem použila jednoduhý kód. Připojením pokaždé jednoho drátku do desky s arduinem jsem se postupně dozvěděla zaprve jaký drátek zodpovídá za co a také i jak se hybou. Byl tam jeden zadrhel a to to že serva stávkovaly, když jsem posílala ty krajní hodnoty(0,180) po úpravě intervalu ve funkci, se všechno srovnalo.

```
#include <Servo.h>

Servo myservo;

int pos = 0;
void setup() {
  myservo.attach(9);
}

void loop() {
  for (pos = 2; pos <= 160; pos += 1) {
    myservo.write(pos);
    delay(15);
  }
  for (pos = 160; pos >= 2; pos -= 1) {
    myservo.write(pos);
    delay(15);
  }
}
```

Kód 3.2: Ukázka kódu k ověření funkčnosti

3.3 PRÁCE S ARDUINEM

První věc, kterou musíme vzít v úvahu, je, že nezbytně potřebujeme mít na svém počítači program Python a také PySerial. Většinu kódu v Pythonu se dá sehnat na internetu a odkazy na ty stránky nalezenete níže v použité literatuře. Takže se jdeme přesunout k popisu. Pro kód v C++ je na zčátek potřeba inicializovat veškere potřebné hodnoty. např:

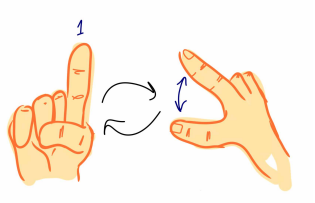
```
const int min= 2;  // minimalní hodnota pro pozici ramene
const int max = 160; // maximalní hodnota pro pozici ramene

bool end;
int data = 0;
int serialData =0;
bool changeMode;
int mode;
const int servoPose = 6; //kolik stavu má rameno
const int servoCount = 4; // počet serv

Servo servo[4]; // pole pro serva
const byte servoPins [] = {2,3,4,5}; //piny pro serva
```

Kód 3.3: Inicializace potřebných hodnot

- Jak jsem již zmiňovala výše zádrhel byl, když se do ramene posílali krajní hodnoty(0 a 180). Kvůli tomu jsem je nastavila na 2 a 160. a to je uložene v proměnných min a max.
- changeMode je pro přehazování mezi snímáním počtu prstu nebo vzdálenosti mezi prsty



Obrázek 3.4: Dvá módy pro kontrolu ramene.

- V proměnné servoPos je uložen počet všech stavu ve kterých se moje rameno může nacházet.

- Jelikož obsahuje 4 serva, tak jsem pro ně vytvořila pole. A každé potřebuje svůj pin k tomu je proměnná servoPins.

```
if(j == (servoMoves[data][0] -1)) {
  if(data == 0 || data == 1) {
    changeMode = true;
    mode = data;
  }
```

Kód 3.4: Podmínka pro přepínání módu

- Tahle jednoduchá podmínka je vytvořena pro kontrolu, kde se zrovna nachází rameno. A když se nachází nahoře nebo dole, tak se přepné mód pro měření vzdálenosti. Ať se může chytnout nebo pustit kulička.

```
   int servoValues [13] [3] = {
    {20,50,20},  //puvodni stav
    {170,80,45}, //dole sběr kuličky cast1
    {123,113,45}, //dolu sběr kuličky cast2
    ...
    {123,130,133},  //uklona cast2
  };

```

Kód 3.5: Dvourozměrné pole na potřebné hodnoty

- Dále máme dvourozměrné pole pro hodnoty pro serva. jsou vždy v jednom řádku jen tři, protože pro ovádání serva v kleštích jsou potřeba jiné hodnoty.

```
  int servoMoves [servoPose] [3] = {
    {2,1,500}, //dolu ke kulice 1
    {2,3,500}, //nahoru ke kulice 2
    {5,5,700}, //ne  3
  ...
  };

```

Kód 3.6: Další pole pro několik hodnot

- Další pole se skládá z několika různých hodnot. A to na prvním místě počet opakování pohybů, na druhém místě se nachází pořadí hodnot serv z předešlého pole.Poslední hodnota je pro delay.

```
1  void move(int data) {
2    int cycle = 0;
3    for(int j = 0 ; j < servoMoves[data][0];j++) {
4      for(int i = 1; i < servoCount;i++) servo[i].write(servoValues[
5                      (servoMoves[data][1])+cycle][i -1]);
6      cycle++;
7      if(cycle > 1) cycle=0;
8      if(j == (servoMoves[data][0] -1)) {
9        if(data == 0 || data == 1) {
10         changeMode = true;
11         mode = data;
12       }
13       end = true;
14       Serial.begin(9600);
15     }
16     delay(servoMoves[data][2]);
17   }
18 }
19
20
```

Kód 3.7: Funkce na rozpohybocání ramene

- Jedna z nejpodstatnějších funkcí. Je pro rozpohybování ramene. Jakmile se načte jedna hodnota, funkce se posune v cyklu o jedno políčko dál v servoValues, a tak než dojde na konec. Potom se zkontroluje zda data jsou 0 nebo 1 pro pochopení že se rameno nachází v pozici u kuličky. Tehdy se změní changeMode a přepne se na měření vzdálenosti mezi prsty.

ZÁVĚR

Cílem samotné práce bylo rozpohybovat rameno a udělat si jasno jak fungují knihovny. Ve finální verzi rameno je schopno uchytit kuličku a přenést ji z konce na začátek. Všechno to je vyřešeno pomocí Pythonu a C++. Využití tohohle projektu může být různé. Například v továrnách, ve skladech, v nanochirurgii nebo jako vzdělávácí hračky. Začátečný cíl se mi podařilo splnit a rameno funguje tak, jak jsem si původně představovala při zahájení vývoje. I přes můj výkon kód přece jen není napsán nejoptimálněji a věřím, že by mohl být mnohem lépe zkonstruován, což do budoucná bych mohla opravit. Také později bych se chtěla víc zaměřit na knihovnu OpenCV a prozkoumat další její funkce.

SEZNAM POUŽITÝCH ZDROJŮ

[1] Základní informace o gestech [Online]. 2023 CEMI MBA Studies s.r.o. [cit. 2020-08-24]. Dostupné z: https://www.cemi.cz/blog/neverbalni-komunikace-co-prozradi-gesta

[2] Typy gest v běžném životě [online]. 1998 [cit. 2020-08-24]. Dostupné z: https://orangeacademy.cz/clanky/rec-tela/

[3] Řeč tela a znakový jazyk, 2023, Slůně - svět jazyků, s.r.o. Dostupné z: https://www.slune.cz/aktualita/znakovy-jazyk/

[4] Knihovna math v pythonu Copyright © 2023 itnetwork.cz. https://www.itnetwork.cz/python/zaklady/python-tutorial-knihovny-math-a-random

[5] Knihovna opencv v pythonu Copyright © 2023 itnetwork.cz. https://www.python-programator.cz/clanky/python-pro-zpracovani-obrazu-opencv

[6] Komunikace s arduinem © 2023 Arduino https://projecthub.arduino.cc/ansh2919/serial-communication-between-python-and-arduino-663756

[7] Komunikace s arduinem 2 Copyright © 2023 Circuit Digest. All rights reserved. https://circuitdigest.com/microcontroller-projects/arduino-python-tutorial

[8] Zkouška funkčnosti https://docs.arduino.cc/learn/electronics/servo-motors

[9] Zkouška funkčnosti https://problemsolvingwithpython.com/11-Python-and-External-Hardware/11.03-Controlling-an-LED/

Seznam obrázků