

POLITECHNIKA ŁÓDZKA
WYDZIAŁ ELEKTROTECHNIKI, ELEKTRONIKI, INFORMATYKI I AUTOMATYKI

PROJEKT KOMPETENCYJNY – SPRAWOZDANIE

Temat: *Parametryzowalny model fizyczny ręki plegicznej z wykorzystaniem układów elektronicznych.*

GRUPA PROJEKTOWA

| | Album | Nazwisko | Imię |
|---|--------|--------------|----------|
| 1 | 228399 | Jędrzejewska | Julia |
| 2 | 228422 | Kuligowski | Zbigniew |
| 3 | 228467 | Stawicka | Joanna |

PROWADZĄCY: **DR INŻ. IGOR ZUBRYCKI**

ŁÓDŹ, 2022

Spis treści

| | |
|---|----|
| 1. Cel i zakres projektu. | 3 |
| 2. Wstęp teoretyczny. | 3 |
| Historia i wybór InMoov. | 3 |
| Konstruowanie modelu ręki. | 3 |
| Wybór rodzaju sterowania. | 6 |
| Zastosowane sterowanie. | 6 |
| 3. Realizacja projektu. | 7 |
| Konstruowanie modelu ręki. Druk 3D. Opis procesu obrabiania wydruków. Montaż elementów. ... | 7 |
| Realizacja sterowania. | 11 |
| Doświadczalny wybór odpowiedniego sterowania. Elektroniczny układ sterowania. | 11 |
| Część programistyczna. | 12 |
| 4. Podsumowanie. | 15 |
| Przyszłe możliwości | 16 |
| 5. Bibliografia. | 16 |

1. Cel i zakres projektu.

Celem projektu było zbudowanie modelu ręki służącego do testowania urządzeń rehabilitacyjnych. Obiekt miał symulować główne cechy sparaliżowanej kończyny, tj. stawiać lekki opór podczas próby poruszenia stawami oraz powracać do stanu rozluźnienia, gdy nie działają żadne siły zewnętrzne (poza grawitacją).

W tym celu należało przede wszystkim zrozumieć samo schorzenie oraz sposób działania testowanych urządzeń i na tej podstawie wybrać odpowiedni model. Dodatkowo należało oszacować siłę z jaką urządzenia miały na siebie oddziaływać i dalej dobrać inne elementy układu (np. napędy), a następnie napisać program w pełni realizujący powyższe założenia.

2. Wstęp teoretyczny.

Historia i wybór InMoov.

Podczas poszukiwania modelu realizującego założenia projektowe natknęliśmy się na projekt francuskiego rzeźbiarza i projektanta Gael'a Langevin'a – InMoov. Jest to jego prywatny projekt, który powstał w 2012 roku jako pierwsza open source'owa proteza ręki. Na jej podstawie powstały takie projekty jak „Bionico”, „E-Nable” i wiele innych. Oryginalny projekt dalej się rozwijał i został pierwszym open source'owym robotem ludzkich rozmiarów, możliwym do zbudowania używając druku 3D.

Wszystkie elementy oraz instrukcje potrzebne do zmontowania ręki są ogólnodostępne na [stronie](#).

Konstruowanie modelu ręki.

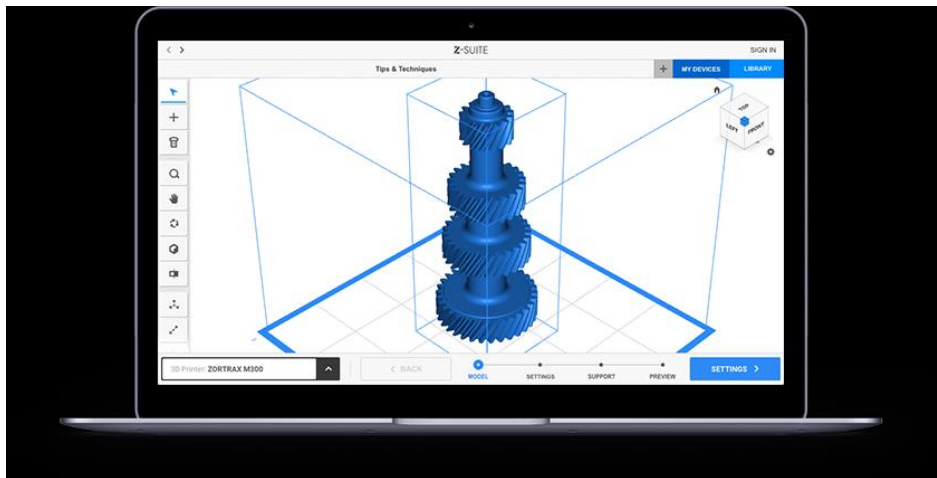
Po przeprowadzeniu research'u zdecydowaliśmy się na wydrukowanie modelu ręki za pomocą drukarki 3D w technologii FFF.

Drukowanie 3D w technologii FDM/FFF stanowi jedną z najstarszych i najbardziej rozpowszechnionych technologii przyrostowych na świecie. Metoda ta polega na układaniu kolejnych warstw przetapianego materiału, pozwalając jednocześnie ostygnąć sąsiednim warstwom i związać się ze sobą przed nałożeniem następnej warstwy.

W celu przygotowania modeli do wydruku wykorzystany został dedykowany program do drukarki firmy Zortrax – ZSuite, podstawowe funkcjonalności, które daje oprogramowanie to:

- Wizualizacja 3D drukowanego obiektu,
- Konwersja plików 3D na modele gotowe do drukowania,
- Wykrywanie zbyt cienkich ścianek modelu, co pozwala wykrycie problematycznych obszarów przed wydrukiem,
- Możliwość dobrania równych wariantów wypełnienia, co pozwala w zależności od potrzeb zredukować zużycie materiału lub zwiększyć wytrzymałość modelu,

- Samodzielne edytowanie podpór minimalizuje szanse na niepowodzenie wydruku.



Rysunek 1: Przykładowy model w programie Z-Suite.

Zdecydowaliśmy się na wykorzystanie drukarki Zortrax M200, ze względu na jej dostępność w siedzibie koła naukowego SKaNeR.

Specjalna konstrukcja drukarki Zortrax M200 znakomicie odprowadza ciepło i przeciwdziała odkształceniom drukowanych modeli. To zasługa przeszklonych paneli bocznych, które pomagają zachować optymalną temperaturę wokół powierzchni roboczej. Drukarka może pracować nieprzerwanie przez wiele godzin, dzięki czemu możliwe było wydrukowanie większych elementów ręki np. część nadgarstka i przedramienia.

Według zaleceń producenta drukarka powinna być bezbłędna nawet przy kilkunastogodzinnej pracy, jednak podczas naszych doświadczeń z wydrukami, błędy się zdarzały. Jednak może to być spowodowane dość dużą eksploatacją samej drukarki, a nie jej błędami konstrukcyjnymi.



Rysunek 2: Drukarka Zortrax M200 wykorzystana do wydruków.

Drukarka posiada możliwość współpracy z wieloma rodzajami filamentów. Nasz wybór padł na półprzezroczysty filament Z-GLASS, który umożliwia drukowanie przepuszczających światło części ostatecznych i funkcjonalnych prototypów. Materiał wykonany z wytrzymałego tworzywa termoplastycznego może imitować szkło. Dodatkowo Z-GLASS jest odporny na zarysowania, działanie soli, kwasów, zasad i rozpuszczalników.

Ponadto Z-GLASS posiada możliwość klejenia, co było konieczne w naszym przypadku, ponieważ palce były drukowane w kilku częściach. Efekt przezroczystości jest widoczny, jednak nie robi takiego wrażenia, jakiego oczekiwaliśmy.

Najczęstsze zastosowania filamentu Z-GLASS to:

- Przepuszczające światło modele koncepcyjne,
- Przemysł samochodowy,
- Modele imitujące szkło,
- Akcesoria dekoracyjne,
- Części do stosowania na zewnątrz,
- Prototypy opakowań,
- Części odporne na sole, kwasy, zasady i rozpuszczalniki.



Rysunek 3: Wykorzystany filament Z-GLASS

Wybór rodzaju sterowania.

| Typ napędu | Typ sterowania | Czujniki | Poziom skomplikowania | Czas wykonania | Ruch wszystkich palców i nadgarstka oddzielnie | Ruch w obie strony | Koszty |
|-----------------------------|---------------------------------|--|---|---|--|--------------------|---|
| Serwo z linkami | Arduino/ inny kontroler | Wszelkie parametry dotyczące momentu, prędkości i położenia odczytywane bezpośrednio z serwa | Niski, ruch każdego stawu realizowany analogicznie, wszystkie parametry odczytywane z serwa | Średni (zależy m.in. od czasu dostawy elementów) | Możliwy | TAK | Wysokie (koszt serw) |
| Pneumatyczny – "strzykawka" | Arduino/ inny kontroler | Czujniki ciśnienia i/lub nacisku | Wysoki, biorąc pod uwagę brak doświadczenia z pneumatyką | Zależy od czasu dostawy elementów, raczej dłuższy ze względu na brak jakiegokolwiek wiedzy w tym zakresie | Możliwy | TAK | Same zawory - tańsze niż Dynamixele, ale należy brać pod uwagę dodatkowe czujniki itp |
| Silnik z hamulcem | Arduino ze sterownikiem silnika | Czujnik siły/ enkoder z silnika | Bardzo wysoki | Wysoki | Możliwy, ale wymaga dużo silników | TAK | Raczej droższe niż serwo |

Tabela 1: Wybór sterowania ręką

Na podstawie powyższej tabeli zdecydowaliśmy, iż najefektywniejszym rozwiązaniem będzie wykorzystanie serwonapędów (i linek) do zadawania ruchu ręki.

Zastosowane sterowanie.

Po sprawdzeniu wielu różnych dostępnych serwonapędów zdecydowaliśmy się wykorzystać te o największej ilości trybów pracy oraz najwyższym współczynniku mocy do rozmiaru. Oczywistym wyborem stały się wówczas serwa Dynamixel firmy ROBOTIS. Mają one w pełni zintegrowany napęd DC, przekładnię redukcyjną, kontroler wewnętrzny, obwód sterownika wewnętrznego i połączenie sieciowe z możliwością łączenia szeregowego. Oczywiście można zarówno odczytywać, jak i zadawać wartości parametrów takich jak: pozycja czy prędkość.



Rysunek 4: Dynamixel.

Do serw postanowiliśmy przymocować linkę służącą jako ścięgna. Wybraliśmy linkę Dyneema, ponieważ charakteryzuje się ona bardzo małą (właściwie zerową) rozciągliwością.



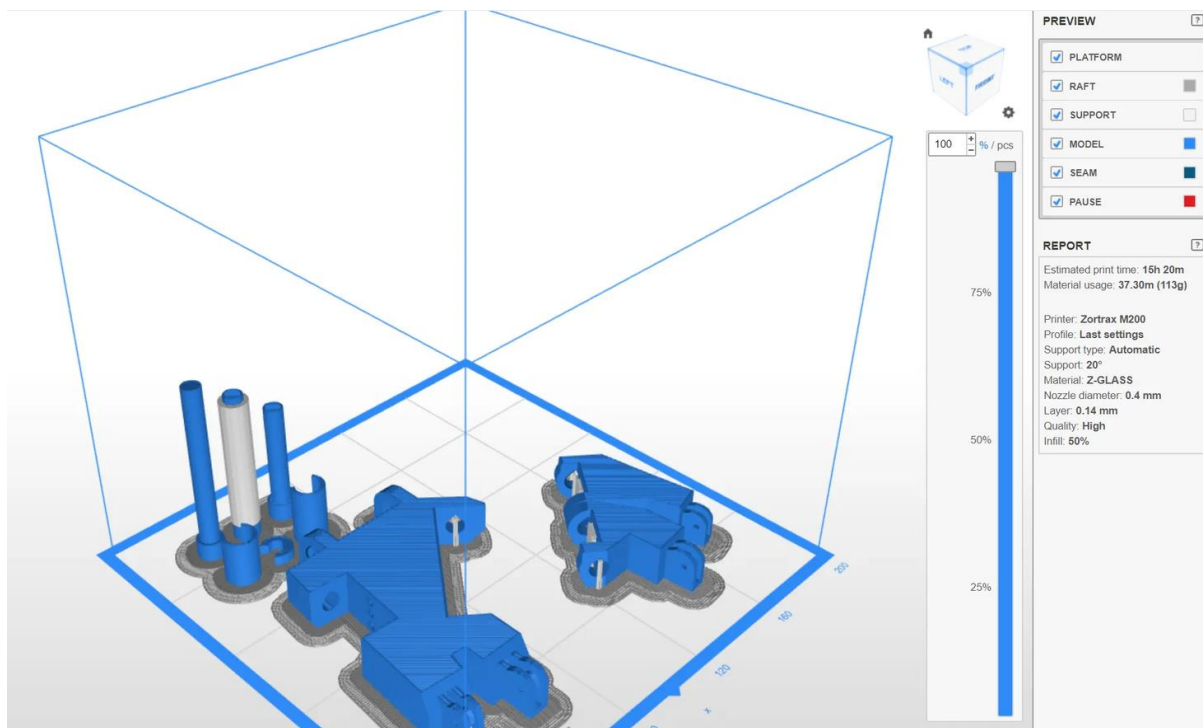
Rysunek 5: Dyneema rope.

3. Realizacja projektu.

Konstruowanie modelu ręki.

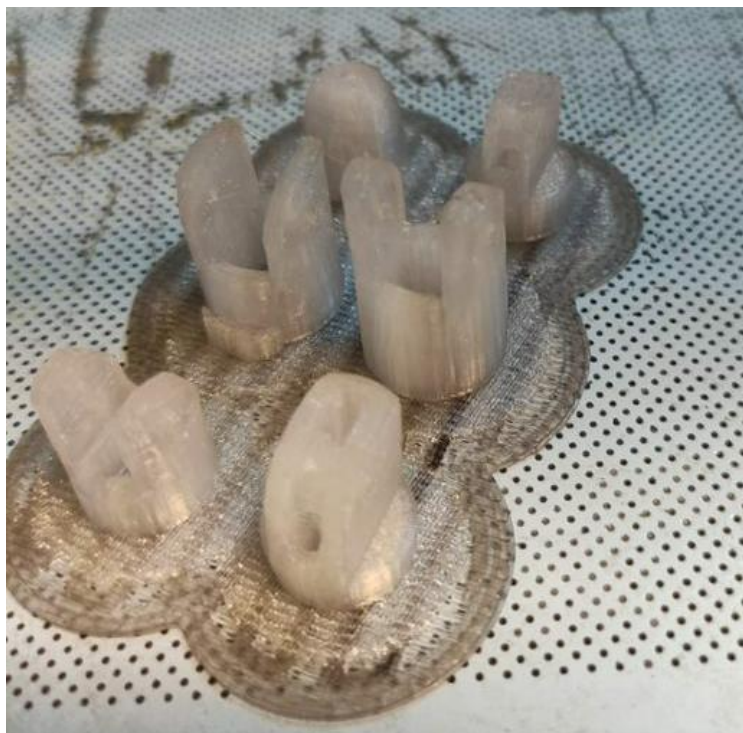
Druk 3D. Opis procesu obrabiania wydruków. Montaż elementów.

Początkowym etapem realizacji projektu był druk 3D poszczególnych komponentów ręki. Pierwszym krokiem w drukowaniu 3D jest wykonanie modelu pożądanego elementu. My skorzystaliśmy z dostępnych na stronie InMoov. Następnie plik należy obrobić przy pomocy programu Zortrax, gdzie można dokonać wyboru między różnymi wypełnieniami, filamentami i wynikowo powstaje plik możliwy do importu do samej drukarki.



Rysunek 6: Widok programu Z-Suite.

W urządzeniu, po załadowaniu pożądanego filamentu, następują automatyczne procesy przygotowania do druku. Czas wydruku pojedynczej części oscylował między 30 min. a 15 godz. w zależności od wielkości komponentu.



Rysunek 7: Wydruk przed obróbką.

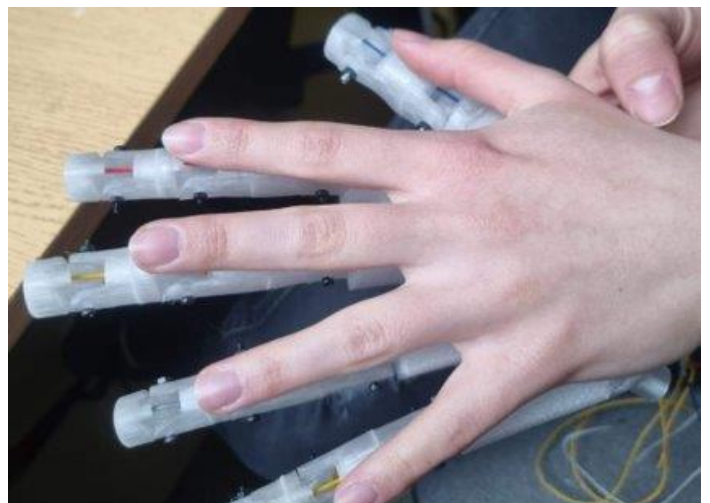
Dalej należało odpowiednio obrobić każdy element, tj. usunąć zbędne podpory, oszlifować ostre krawędzie. Gdy każda część była już gotowa, można było przystąpić do montażu. By przetestować

cały proces, wybraliśmy jedną z mniej skomplikowanych części – palec. Efekt widoczny jest na zdjęciu poniżej.



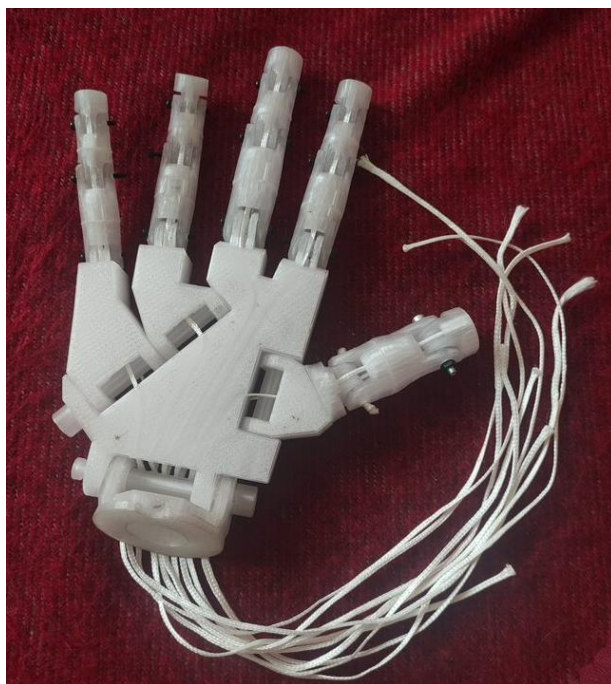
Rysunek 8: Gotowy wydruk 3D palca.

W miejscu stawów zastosowaliśmy śruby 2 mm, które utrzymują paliczki w odpowiedniej pozycji. W części miejsc konieczne było klejenie. Zastosowaliśmy klej producenta Pattex 2-cyanoakrylan etylu. Jak było wspomniane przy wyborze filamentu braliśmy pod uwagę możliwość klejenia. Gdy poznaliśmy już, jak działa proces, mogliśmy przystąpić do drukowania kolejnych części. Jak się końcowo okazało, nie wystąpiły większe problemy przy tym etapie realizacji projektu. Wdrukowana i złożona dłoń jest widoczna na zdjęciach poniżej.



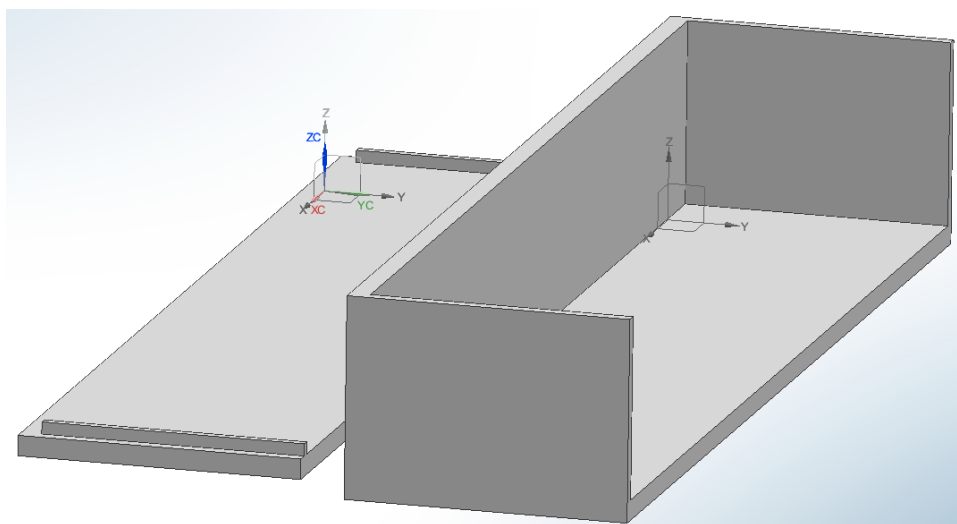
Rysunek 9: Wydrukowana dłoń wraz z referencją do dłoni rzeczywistej.

Po złożeniu dłoni można było zastosować linki Dyneema w funkcji ściągien. Model jest tak zaprojektowany, że ich użycie nie stanowiło żadnego kłopotu. Pociągnięcie za jedną z nich powoduje zgięcie danego palca.



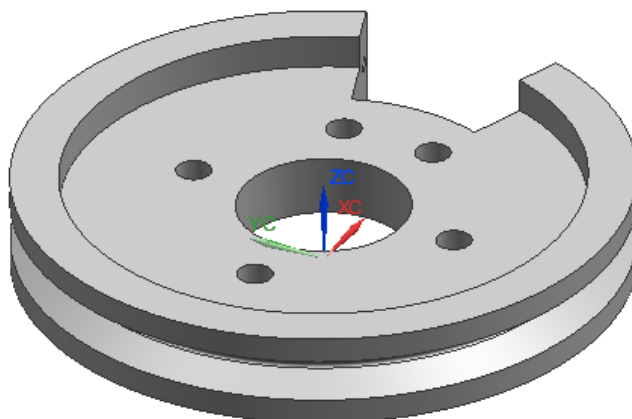
Rysunek 10: Widok złożonej dłoni z zastosowanymi linkami Dyneema.

W celu odpowiedniego umiejscowienia serw względem ręki, zaprojektowaliśmy pudełko z otwartą ścianką, gdzie można je umieścić. Model 3D widoczny jest na zrzucie poniżej.



Rysunek 11: Model pudełka na serwa.

Dodatkowo, aby zwiększyć moment siły między wałem serwa a linkami, przeprojektowaliśmy nakładki na Dynamixel'e oferowane przez InMoov. Nakładka wygląda analogicznie do oryginalnej, została jedynie zwiększona średnica z zachowaniem innych, kluczowych wymiarów. Jej model, przystosowany do druku, jest widoczny poniżej.

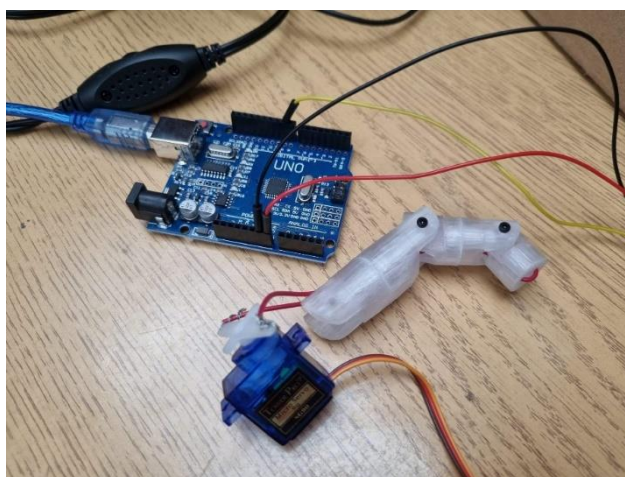


Rysunek 12: Model nakładki na wał serwa.

Realizacja sterowania.

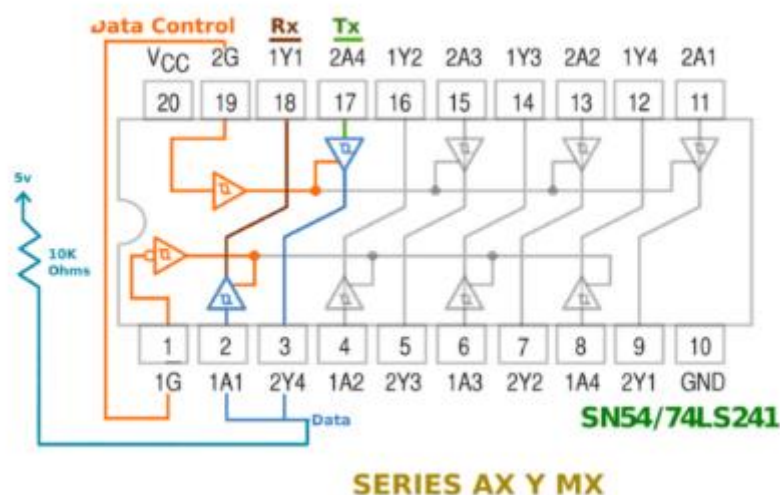
Doświadczalny wybór odpowiedniego sterowania. Elektroniczny układ sterowania.

Na podstawie rozważań teoretycznych zdecydowaliśmy się na zastosowanie do sterowania serw. Początkowo rozważaliśmy użycie mikro-serw, często dołączanych do zestawów Arduino. Takie rozwiązanie nie wymagało stworzenia dodatkowych układów elektronicznych, gdyż wystarczyło podłączyć mechanizm bezpośrednio do płytki. Przeprowadziliśmy prosty test czy moc takiego urządzenia będzie w stanie poruszyć samym palcem. Unieruchomiliśmy palec i serwo poprzez przymocowanie ich do stołu. Linki zostały prowizorycznie przymocowane do śmigieł. Dalej wgraliśmy prosty program wykonujący ruch wałem o 180 stopni w jedną i drugą stronę. Niestety nie spełniło to naszych oczekiwań, gdyż takie urządzenia posiadają zbyt małą moc, by móc zgiąć palec. Stąd konieczne było wykorzystanie bardziej zaawansowanych układów.



Rysunek 13: Układ testujący działanie mikro-serw.

Po przeprowadzeniu wcześniej wspomnianych badań postanowiliśmy, że najprostszym i zarazem najefektywniejszym sposobem będzie użycie serw Dynamixel z płytką Arduino. W tym celu koniecznym było zastosowanie dodatkowego układu bufora trójstanowego. Zakupiliśmy bufor SN74LS241N i zrealizowaliśmy podłączenia na podstawie schematu poniżej.



Rysunek 14: Schemat podłączeń bufora trójstanowego.

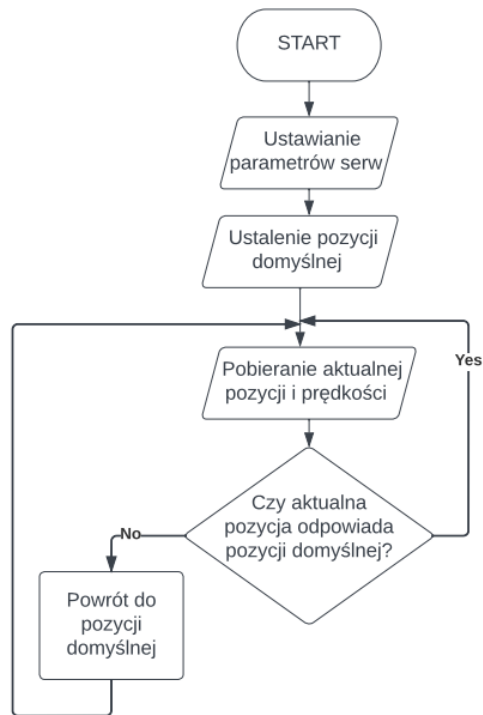
Dzięki temu byliśmy w stanie podłączyć płytkę do serw Dynamixel i dalej działać pod względem programistycznym. Aby cały układ mógł prawidłowo funkcjonować, niezbędne jest podłączenie dodatkowego zasilania. W tym celu użyliśmy zasilacz 12V.

Część programistyczna.

W trakcie pisania programu zauważyliśmy, że zakupione przez nas serwonapędy blokują możliwość poruszania tarczą (przy podłączonym zewnętrznym zasilaniu) oraz nie mają funkcji sterowania momentem, co nie było jasno przedstawione w dokumentacji. Powodowało to, że mimo poprawnie reagującego urządzenia na zmianę prędkości lub pozycji nie byliśmy w stanie manualnie zmieniać tych parametrów. Przez to byliśmy zmuszeni do zmiany założeń - realizacja plegii nie była już możliwa. Ustaliliśmy więc, że w zamian model będzie symulował rękę z przykurczem. Jednakże, aby móc zaprezentować chociaż część założonej wcześniej funkcjonalności do projektu dołożyliśmy jedno serwo Dynamixel XC430-W150-T, które ma możliwość sterowania momentem.

Funkcjonalność zakupionych komponentów nie była, niestety, jedynym napotkanym problemem w trakcie prób ich zaprogramowania. Wiele czasu poświęciliśmy na szukanie powodu nagłego przestawiania działania programów, które wcześniej bez problemu się kompilowały i działały zgodnie z założeniami. Zakładamy, że część problemów mogła wynikać z problemów komunikacyjnych, tj. oczekiwane przez nas działanie wymagało użycia dwóch rejestrów szeregowych, a wykorzystywana wówczas płytką Arduino UNO zapewniała tylko jeden. Została później wymieniona na Arduino MEGA, co znacznie ułatwiło odczytywanie aktualnych wartości parametrów, ale nie zniwelowało innych problemów.

Ostatecznie napisaliśmy program realizujący przykurcz ręki. Skorzystaliśmy z biblioteki „Dynamixel2Arduino.h”, która bezproblemowo współpracuje z serвом AX12. Przemysleliśmy, jak ma działać program i nasze założenia przedstawiliśmy w postaci schematu blokowego.



Rysunek 15: Schemat blokowy programu ostatecznego.

Ostateczny kod jest widoczny poniżej.

```

#include <Dynamixel2Arduino.h>

// Please modify it to suit your hardware.
#if defined(ARDUINO_AVR_UNO) || defined(ARDUINO_AVR_MEGA2560) // When using DynamixelShield
#include <SoftwareSerial.h>
SoftwareSerial soft_serial(7, 8); // DYNAMIXELShield UART RX/TX
#define DXL_SERIAL Serial
#define DEBUG_SERIAL soft_serial
const int DXL_DIR_PIN = 2; // DYNAMIXEL Shield DIR PIN
#elif defined(ARDUINO_SAM_DUE) // When using DynamixelShield
#define DXL_SERIAL Serial
#define DEBUG_SERIAL SerialUSB
const int DXL_DIR_PIN = 2; // DYNAMIXEL Shield DIR PIN
#elif defined(ARDUINO_SAM_ZERO) // When using DynamixelShield
#define DXL_SERIAL Serial1
#define DEBUG_SERIAL SerialUSB
const int DXL_DIR_PIN = 2; // DYNAMIXEL Shield DIR PIN
#elif defined(ARDUINO_OpenRB) // When using OpenRB-150
//OpenRB does not require the DIR control pin.
#define DXL_SERIAL Serial1
#define DEBUG_SERIAL Serial
const int DXL_DIR_PIN = -1;
#else // Other boards when using DynamixelShield
#define DXL_SERIAL Serial1
#define DEBUG_SERIAL Serial
const int DXL_DIR_PIN = 2; // DYNAMIXEL Shield DIR PIN
#endif

const uint8_t DXL_ID = 1;
const float DXL_PROTOCOL_VERSION = 2.0;

Dynamixel2Arduino dxl(DXL_SERIAL, DXL_DIR_PIN);

//This namespace is required to use Control table item names
using namespace ControlTableItem;

void setup() {

  // Set Port baudrate to 57600bps. This has to match with DYNAMIXEL baudrate.
  dxl.begin(57600);
  // Set Port Protocol Version. This has to match with DYNAMIXEL protocol version.
  dxl.setPortProtocolVersion(DXL_PROTOCOL_VERSION);
  // Get DYNAMIXEL information

```

```

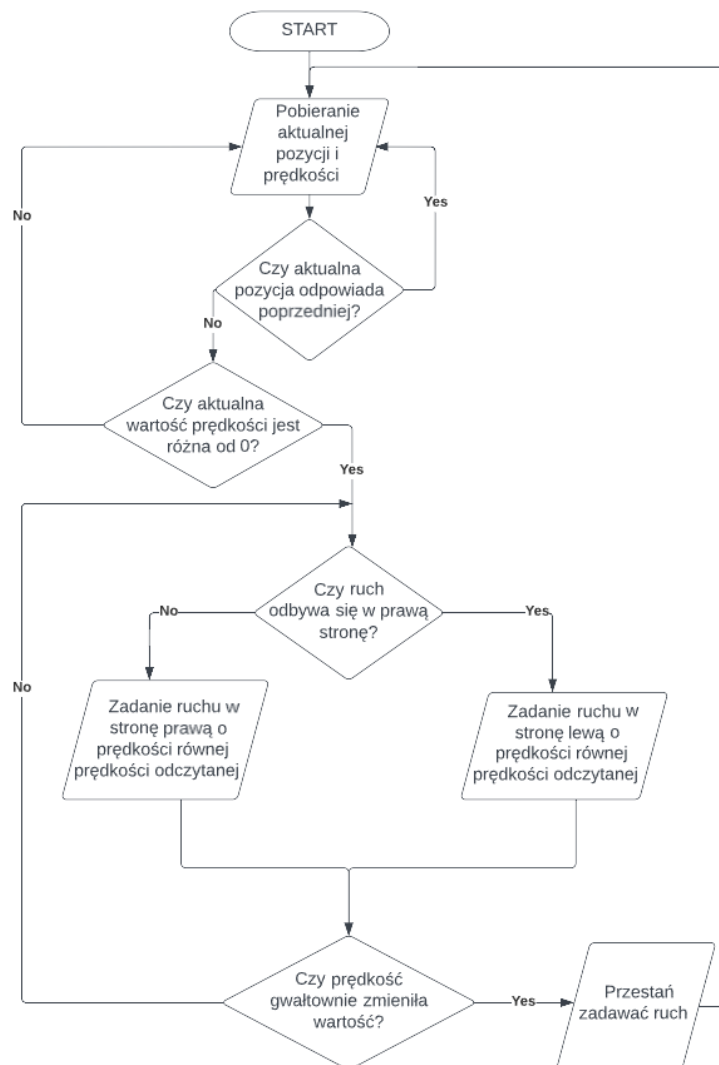
dxl.torqueOff(DXL_ID);
dxl.ledOn(DXL_ID);
int start_position = dxl.getPresentPosition(DXL_ID);
delay(25);
}

void loop() {
int position_servo = dxl.getPresentPosition(DXL_ID);
delay(25);
int velocity_servo = dxl.getPresentVelocity(DXL_ID);
delay(25);

if(position_servo >= (start_position - 10) && position_servo <= (start_position + 10))
{
dxl.setGoalVelocity(DXL_ID, velocity_servo);
dxl.setGoalPosition(DXL_ID, start_position);
}
}

```

W naszych założeniach program miał symulować zachowanie ręki plegicznej. Realizacja takiego programu miała być oparta na schemacie blokowym poniżej.



Rysunek 16: Schemat ideowy działania programu.

4. Podsumowanie.

Podsumowując realizację projektu, udało nam się wydrukować wszystkie elementy ręki na drukarce 3D oraz złożyć je za pomocą śrub i kleju. Zamontowaliśmy linki, które pozwalają na ruch poszczególnymi palcami, który jest generowany z serw. Zatem udało się skompletować cały model dłoni z elementami przedramienia.

Ostatecznie model ręki wygląda jak na zdjęciu.



Rysunek 17: Zdjęcie złożonej ręki z zamontowanymi serwami i ścięgna.

Przeprowadziliśmy wstępne testy serw Dyamixel, korzystając z płytki Arduino Uno wraz z układem bufora trójstanowego. Jednak docelowo użyliśmy płytki Arduino Mega2560, ze względu na problem z komunikacją magistrali szeregową. Następnie napisaliśmy prototypowy program w Arduino IDE.

Nie udało nam się osiągnąć zakładanego celu, ponieważ wystąpiły pewne trudności, takie jak: niska dostępność dokumentacji technicznych, długi czas oczekiwania na dostawę komponentów (około 2 miesiące), problemy ze znalezieniem aktualnych bibliotek, niszowość tematu – trudności z odnalezieniem materiałów, na których można by polegać. Ponadto na stronie producenta serw zostały zawarte niepełne informacje, co do funkcjonalności oferowanych przez Dynamixele. Między innymi zostaliśmy wprowadzeni w błąd przy opisie dotyczącym sterowania momentem. To uniemożliwiło zrealizowanie założeń wstępnych projektu, które polegały na zasymulowaniu plegii ręki. Zostaliśmy więc zmuszeni do zrealizowania samego przykurczu.

Przyszłe możliwości

Ze względu na niespełnienie wszystkich założonych celów zostawiliśmy wiele możliwości dalszej rozbudowy, czy też nawet częściowej przebudowy projektu. Po pierwsze, do samego modelu nie zostały jeszcze przymocowane opuszki palców, co pozwala na zmianę linek, służących jako ścięgna. Podobnie, śrubki łączące kolejne elementy palców nie zostały docięte, aby umożliwić chociażby demontaż. Dodatkowo serwa oraz linki nie zostały przymocowane na stałe, co z kolei daje możliwość doboru odległości (od samego modelu) na jakiej zostaną zamocowane silniki. Ze względu na niedziałanie programu zostawiliśmy też możliwość doboru płytki, jaka będzie wykorzystywana w przyszłości do sterowania napędami.



Rysunek 18: Do przyklejenia opuszki palców i schowanie śrub w modelu.

5. Bibliografia.

| | |
|-----|---|
| [1] | https://zortrax.com |
| [2] | https://3dgence.com/pl/3dnews/technologie-druku-3d-ktora-z-nich-wybrac-dla-swojego-projektu-i-dlaczego/ |
| [3] | https://inmoov.fr/hand-and-forarm/ |
| [4] | https://savageelectronics.com/arduino-y-biblioteca-dynamixel-ax-12a/ ary – Savage Electronics Blog |
| [5] | https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/ |
| [6] | https://github.com/zkuligowski/Hand-project |