



**WYDZIAŁ ELEKTRYCZNY**  
**Katedra Automatyki i Robotyki**

Praca dyplomowa inżynierska

Marcin Przekop

Projekt i wykonanie hybrydowego wielokończynowego robota kroczącego  
z kołami Mecanum

Opiekun pracy  
dr inż. Roman Trochimczuk

Białystok 2023



Tytuł pracy

**Projekt i wykonanie hybrydowego wielokończynowego robota krocącego z kołami Mecanum**

## **Streszczenie**

W niniejszej pracy zostanie wykonany projekt wielokończynowego robota krocącego z kołami Mecanum oraz jego realizacja umożliwiającą dwa tryby poruszania się – kroczenie oraz jazdę na kołach. Przedstawiona zostanie metodyka pracy, opis robota, modele 3D robota oraz schematy elektryczne. Napisany zostanie również kod do sterowania robotem oraz wykonana będzie aplikacja na telefon, dzięki której możliwe będzie jego zdalne kontrolowanie.

Title of diploma thesis

**Design and implementation of a hybrid multi-limbed walking robot with Mecanum wheels**

## **Summary**

This work will involve the design of a multi-limbed walking robot with Mecanum wheels and its implementation ensuring two modes of movement – walking and driving on wheels. The work methodology, robot description, 3D models of the robot and electrical diagrams will be presented. The code to control the robot will also be written and an application for the phone will be created, thanks to which it will be possible to control it remotely.

## Spis treści

Spis ważniejszych oznaczeń .....	8
1. Wstęp .....	9
1.1. Cel pracy i zakres pracy .....	10
1.2. Metodyka realizacji .....	10
1.3. Charakterystyka pracy .....	11
2. Opis dostępnych rozwiązań robotów koczających oraz posiadających koła Mecanum .....	13
2.1. Przegląd dostępnych rozwiązań robotów koczających .....	13
2.1.1. Roboty czterokończynowe (Quadruped) .....	13
2.1.2. Roboty sześciokończynowe (Hexapod) .....	15
2.1.3. Roboty ośmiokończynowe (Octopod) .....	16
2.2. Koła Mecanum .....	18
2.2.1. Zasada działania i sterowanie .....	19
2.3. Roboty z kołami Mecanum .....	20
2.4. Wnioski .....	21
3. Założenia projektowe do realizacji hybrydowego robota koczającego z kołami Mecanum .....	23
3.1. Założenia dotyczące konstrukcji robota .....	23
3.2. Wnioski .....	24
4. Obliczenia wymagań napędów .....	25
4.1. Obliczenia dotyczące serwonapędów .....	25
4.2. Obliczenia dotyczące napędów kół Mecanum .....	26
4.3. Wnioski .....	27
5. Trójwymiarowy model geometryczny robota wykonany w SolidWorks .....	29
5.1. Platforma robota .....	29
5.2. Konstrukcja pojedynczej kończyny .....	31
5.3. Mocowanie napędów kół Mecanum .....	31
5.4. Obliczenia masowe .....	33
5.5. Dokumentacja techniczna .....	33
5.6. Wnioski .....	33
6. Dobór podzespołów .....	35
6.1. Mikrokontroler .....	35
6.2. Serwonapęd MG996R .....	36
6.3. Sterownik do serwonapędów PCA9685 .....	37

6.4. Silnik DC MT80 70.....	38
6.5. Sterownik do silników DC L298N.....	39
6.6. Bateria LiPo REDOX 11,1V 3S 900mAh 20C .....	40
6.7. Przetwornica step-down SZBK07 .....	41
6.8. Przetwornica step-down LM2596S .....	42
6.9. Moduł BT HC-05 .....	43
6.10. Wnioski .....	43
7. Drukowanie części robota z wykorzystaniem druku 3D .....	45
7.1. Drukarka MSLA.....	45
7.2. Drukarka FDM .....	46
7.3. Slicer-y .....	48
7.4. Obróbka druku z żywicy .....	49
7.5. Złożenie robota.....	49
7.6. Wnioski .....	50
8. Schemat elektryczny .....	51
8.1. Schemat połączeń opracowany w programie KiCAD.....	51
8.2. Wnioski .....	53
9. Aplikacja do sterowania robotem .....	55
9.1. Oprawa graficzna aplikacji.....	55
9.2. Część logiczna aplikacji .....	56
9.3. Wnioski .....	58
10. Program sterowania robotem .....	59
10.1. Sterowanie częścią kroczącą robota .....	59
10.2. Sterowanie napędami kół Mecanum .....	63
10.3. Zmiana trybu poruszania się .....	63
10.4. Komunikacja z wykorzystaniem Bluetooth .....	65
10.5. Wnioski .....	65
11. Testy hybrydowego robota krocącego z kołami Mecanum .....	67
11.1. Testy trybu krocącego .....	67
11.2. Testy trybu jazdy .....	69
11.3. Wnioski .....	70
12. Koszt konstrukcji robota .....	71
12.1. Zestawienie kosztów poszczególnych części .....	71
12.2. Wnioski .....	72

13. Podsumowanie.....	73
Bibliografia.....	75
Załączniki pracy .....	78

## Spis ważniejszych oznaczeń

ABS	Akrylonitryl-Butadien-Styren jest to typ kopolimeru charakteryzujący się dużą twardością i uduernością
AGV	Automated Guided Vehicle jest to autonomiczny pojazd bezzalogowy sterowany bez udziału operatora
BT	Technologia Bluetooth
FDM	Fused Deposition Modeling – technologia wytwarzania addytywnego polegająca na wytłaczaniu materiału przez dyszę
Hexapod	Robot sześcionożny
MSLA	Masked Stereolithography Apparatus – technologia druku 3D wykorzystująca ekran maskujący
Octopod	Robot ośmionożny
Quadruped	Robot czworonożny
Silnik DC	Silnik prądu stałego



# 1.Wstęp

Postęp technologiczny przyczynił się do powstania nowatorskich i bardziej zaawansowanych metod realizacji ruchu robotów, co znacząco poszerzyło ich możliwości i potencjalne zastosowania w różnych środowiskach. Szczególną uwagę warto poświęcić robotom kroczącym oraz posiadającym koła Mecanum, które dzięki swojej wszechstronności i zwinności są coraz częściej wykorzystywane w różnych dziedzinach takich jak przemysł czy też wojskowość.

Niniejsza praca zostanie poświęcona wykonaniu i realizacji autorskiego projektu hybrydowego robota kroczącego łączącego ze sobą zalety obydwu wyżej wymienionych metod realizacji ruchu. Zakłada się wykonanie modelu trójwymiarowego robota wraz z doбором odpowiednich komponentów, takich jak napędy, sterowniki oraz zasilanie. Stworzony zostanie program umożliwiający ruch robota oraz wykonana zostanie aplikacja na telefon umożliwiająca jego sterowanie. Dodatkowo, praca obejmuje sporządzenie rysunków technicznych elementów robota oraz schematów elektrycznych. Praca zakłada również wykonanie odpowiednich testów robota, dzięki czemu możliwe będzie określenie jego możliwości.

Na końcu pracy przedstawione zostanie podsumowanie, osiągnięte cele oraz wskazany zostanie potencjalny kierunek dalszego rozwoju projektu.

## 1.1. Cel pracy i zakres pracy

Celem niniejszej pracy dyplomowej jest wykonanie robota krocącego z kołami Mecanum oraz sporządzenie odpowiedniej dokumentacji technicznej.

Zakres pracy obejmuje wykonanie następujących czynności:

- przegląd literatury dotyczącej rozwiązań technicznych stosowanych w robotach krocących oraz mobilnych z kołami Mecanum,
- analiza konstrukcji robotów krocących wielokończynowych oraz z kołami Mecanum,
- wykonanie założeń inżynierskich do własnej konstrukcji robota krocąco-jeżdżącego,
- opracowanie dokumentacji autorskiego projektu technicznego w środowisku programu CAD w postaci rysunków wykonawczych i złożeniowych,
- praktyczne wykonanie projektu systemu mechanicznego, elektrycznego i informacyjnego robota hybrydowego oraz ich integracja,
- testowanie wykonanej konstrukcji wykonanego robota mobilnego hybrydowego krocącego z kołami Mecanum.

## 1.2. Metodyka realizacji

Pierwszym etapem pracy będzie przegląd obecnych rozwiązań dotyczący realizacji robotów krocących oraz posiadających koła Mecanum. Następnie sporządzone zostaną założenia projektowe będące fundamentem do dalszej pracy. Kolejnym krokiem jest wykonanie obliczeń, na podstawie których dobrane zostaną komponenty robota. Dalej wykonany będzie trójwymiarowy model robota, który w dalszej części pracy posłuży jako podstawa do wydrukowania jego części, z wykorzystaniem technologii druku 3D. Następnym etapem jest stworzenie programu sterującego na mikrokontroler oraz aplikacji do kontroli ruchu robota. Ostatnim elementem pracy jest wykonanie odpowiednich testów oraz sporządzenie odpowiedniej dokumentacji technicznej.

## 1.3. Charakterystyka pracy

W rozdziale pierwszym przedstawiono wstęp do pracy, cel i zakres oraz metodykę jej realizacji.

Drugi rozdział dotyczy przeglądu dostępnych rozwiązań robotów kroczących oraz posiadających koła Mecanum.

W rozdziale trzecim przedstawiono założenia projektowe dotyczące realizacji robota.

Rozdział czwarty poświęcono obliczeniom dotyczącym wymogów poszczególnych napędów.

W piątym rozdziale przedstawiono szczegółowe modele trójwymiarowe robota stworzone w oprogramowaniu SolidWorks oraz wyznaczono ich przewidywaną masę.

Szósty rozdział zawiera informacje dotyczące dobranych podzespołów wraz z uzasadnieniami.

W siódmym rozdziale omówiono wykorzystane drukarki oraz urządzenia w celu uzyskania części składowych robota oraz przedstawiono jego całkowite złożenie.

Rozdział ósmy zawiera schemat oraz opis połączeń elektrycznych podzespołów robota.

W rozdziale dziewiątym opisano oprawę graficzną aplikacji oraz część logiczną.

Dziesiąty rozdział dotyczy programu na mikrokontroler Arduino wraz z opisem przykładowych funkcji.

W rozdziale jedenastym zawarto wyniki testów robota oraz wyliczono odpowiednie parametry.

Dwunasty rozdział zawiera zestawienie kosztów poszczególnych części.

Rozdział trzynasty zawiera podsumowanie całej pracy, opisano osiągnięte cele oraz zaproponowano dalszy kierunek rozwoju pracy.

Na końcu pracy zamieszczono bibliografię oraz listę załączników pracy.



## **2. Opis dostępnych rozwiązań robotów kroczących oraz posiadających koła Mecanum**

Analiza dostępnych rozwiązań w dziedzinie robotów kroczących oraz robotów wyposażonych w koła Mecanum jest pierwszym etapem w celu zaprojektowania własnego rozwiązania, łączącego oba sposoby poruszania się. Obie technologie oferują unikalne cechy, które sprawiają, że są one użyteczne w innych zastosowaniach. Roboty kroczące, wzorowane na biomechanice organizmów naturalnych, charakteryzują się przede wszystkim zdolnością do pokonywania nierówności terenu, natomiast roboty z kołami Mecanum umożliwiają płynne ruchy w różnych kierunkach bez konieczności obracania samym robotem, co przekłada się na ich zwinnosć i krótki czas wymagany do pokonywania krętych tras. Poprzez analizę istniejących rozwiązań w tych dwóch obszarach, możliwe jest wyciągnięcie cennych wniosków dotyczących skuteczności, zalet i ograniczeń każdego z tych typów robotów, które stanowią istotny punkt wyjścia do dalszych rozważań nad hybrydowym robotem kroczącym z kołami Mecanum.

### **2.1. Przegląd dostępnych rozwiązań robotów kroczących**

Obecnie realizację ruchu kroczącego robota można wykonać na wiele sposobów i jest ona w głównej mierze podyktowana liczbą kończyn robota. Każda konfiguracja ma swoje wady oraz zalety, w związku z czym kluczowy będzie jej odpowiedni wybór.

#### **2.1.1. Roboty czterokończynowe (Quadruped)**

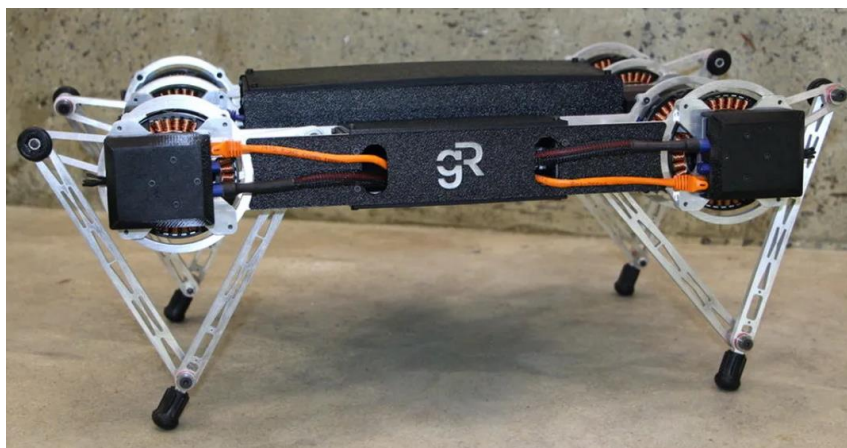
Istotną rolę w projektowaniu robotów odgrywa biomimetyka, czyli naśladowanie rozwiązań występujących w naturze. Przykładem tego są roboty czterokończynowe przypominające psa, wyprodukowane przez amerykańską firmę Boston Dynamics z serii „Spot”, widoczny na rysunku 2.1.



*Rys.2.1. Robot z serii „Spot” od Boston Dynamics [14]*

Zaletą takiej konstrukcji jest niższa cena oraz mniejsza masa wynikająca z posiadania mniejszej liczby kończyn w porównaniu do innych konfiguracji, natomiast wadą jest niska stabilność robota, która wymaga użycia sterowania z wykorzystaniem kinematyki odwrotnej oraz żyroskopu w celu uzyskania płynnego ruchu. Dodatkowo ciężar robota rozkłada się jedynie na cztery nogi, co powoduje konieczność zastosowania napędów oferujących większe momenty obrotowe.

Kolejnym przykładem quadrupeďa jest „Minitaur” widoczny na rysunku 2.2.



*Rys.2.2. Robot „Minitaur” [30]*

Realizacja jego ruchu odbywa się poprzez wykonywanie skoków kontrolując odpowiednio przednią i tylną parą kończyn. Wadą takiego sposobu poruszania się jest przede wszystkim występowanie dużych wstrząsów występujących przy lądowaniu, co stawia duże wymagania dotyczące wytrzymałości konstrukcji robota oraz użytych materiałów, aby robot nie uległ uszkodzeniu po jego krótkim użytkowaniu. Dodatkowym problemem jest konieczność użycia silników prądu stałego, oferujących duży moment, aby umożliwić wykonanie gwałtownego skoku. Wykonanie odpowiedniego sterowania do tego typu robota jest złożonym procesem wymagającym wykonywania wielu obliczeń co powoduje, że konieczne jest użycie mikrokontrolera o większej mocy obliczeniowej.

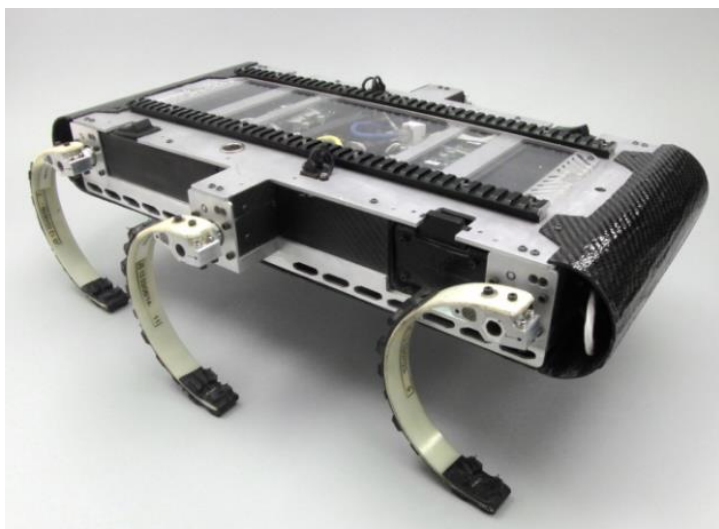
### 2.1.2. Roboty sześciokończynowe (Hexapod)

Konstrukcja sześciokończynowa jest powszechnie stosowana we wszelakich projektach robotów kroczących, a zawdzięcza to dzięki temu, że podczas ruchu trzy nogi zawsze wspierają masę robota, co wpływa na jego stabilność i prostotę w tworzeniu oprogramowania. Dodatkowo masa robota podtrzymywana jest przez większą liczbę nóg w porównaniu do quadraped'a, co przekłada się na mniejsze wymagania dotyczące silników. Na rysunku 2.3 przedstawiono przykładowy wygląd omawianego typu robota zainspirowanego skorpionem stworzonego przez studenta Politechniki Łódzkiej [16].



*Rys.2.3. Przykładowy robot typu hexapod zainspirowany skorpionem [16]*

Innym przykładem hexapod'a jest „RHex” widoczny na rysunku 2.4.



*Rys.2.4. Robot „RHex” [31]*

Każda z sześciu nóg posiada własny silnik obracający nogą. Kontrolowanie kierunku ruchu robota wykonuje się poprzez kontrolę prędkości obracania się poszczególnych napędów, dzięki czemu sterowanie tego typu robotem jest stosunkowo proste. Problemem przy tego typu realizacji ruchu jest dobór odpowiednich materiałów konstrukcyjnych kończyn, aby nie uległy one uszkodzeniu podczas kroczenia. Materiał ten musi być na tyle sztywny, aby zapewnić stabilność robota, jednakże zastosowanie zbyt sztywnego budulca powodowałyby ciągłe usterki kończyn.

### **2.1.3. Roboty ośmiokończynowe (Octopod)**

Ośmiokończynowe roboty nie są tak popularne jak quadrupe'y, czy hexapod'y, ze względu na dużą liczbę nóg, która przekłada się na wysokie koszty oraz większą masę całkowitą konstrukcji. Octopody oferują te same zalety, co sześciokończynowe roboty, a ponadto zapewniają redundancję, dzięki czemu usterka pojedynczej kończyny nie będzie skutkować unieruchomieniem robota. Na rysunku 2.5. ukazano przykładową realizację robota w tejże konfiguracji.

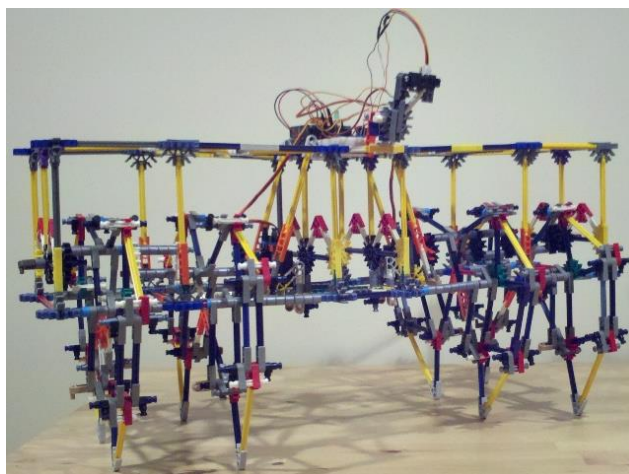




*Rys.2.5. Przykładowy robot ośmio-kończynowy [15]*

Realizacja ruchu tego typu konstrukcji jest podobna jak w przypadku robota widocznego na rysunku 2.3. Różnica polega na tym, że w tym przypadku robot podparty jest zawsze na co najmniej czterech nogach, dzięki czemu możliwe jest wykorzystanie serwonapędów o mniejszej mocy.

Kolejnym przykładem octopod'a jest widoczny na rysunku 2.6 robot o nazwie „K'nexabeast”.



*Rys.2.6. Robot typu octopod „K'nexabeast” [32]*

Robot ten napędzany jest przez zaledwie dwa silniki - każdy napędza po cztery kończyny. Sprawia to, że wykonanie programu sterującego jest niezwykle proste a obrót robota dokonuje się jedynie poprzez regulację prędkości dwóch napędów. Głównym wyzwaniem podczas budowy takiego robota jest odpowiednie zaprojektowanie jego nóg, łącząc je odpowiednio z poszczególnym napędem.

## 2.2. Koła Mecanum

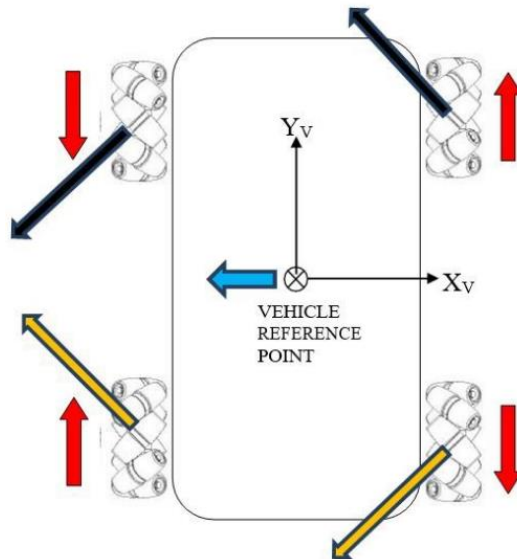
Koła Mecanum zawierają zestaw wałków obrotowych zamocowanych wzdłuż jego obwodu przechylonych pod kątem  $45^\circ$  względem osi obrotu koła [7]. Dzięki takiemu rozwiązaniu możliwy jest ruch we wszystkich kierunkach oraz obrót robota bez posiadania osi skrętnej kół, jedynie za pomocą kontroli kierunku obrotu odpowiednich napędów. Na rysunku 2.7 pokazano przykładowe koła Mecanum.



*Rys.2.7. Wygląd kół Mecanum [17]*

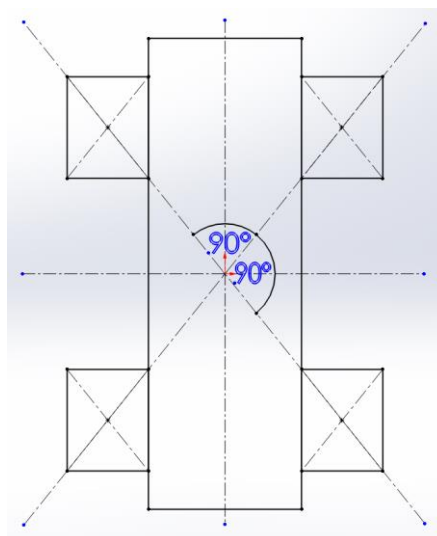
### 2.2.1. Zasada działania i sterowanie

Dzięki temu, że wałki obrotowe są zamocowane pod kątem  $45^\circ$  siła wypadkowa wynikająca z obrotu kół skierowana jest prostopadłe do osi obrotu wałków [7]. Dzięki odpowiedniej konfiguracji kierunku napędów możliwe jest zrównoważenie sił działających w poszczególnej osi i uzyskanie np. ruchu w bok robota, tak jak to pokazano na rysunku 2.8.



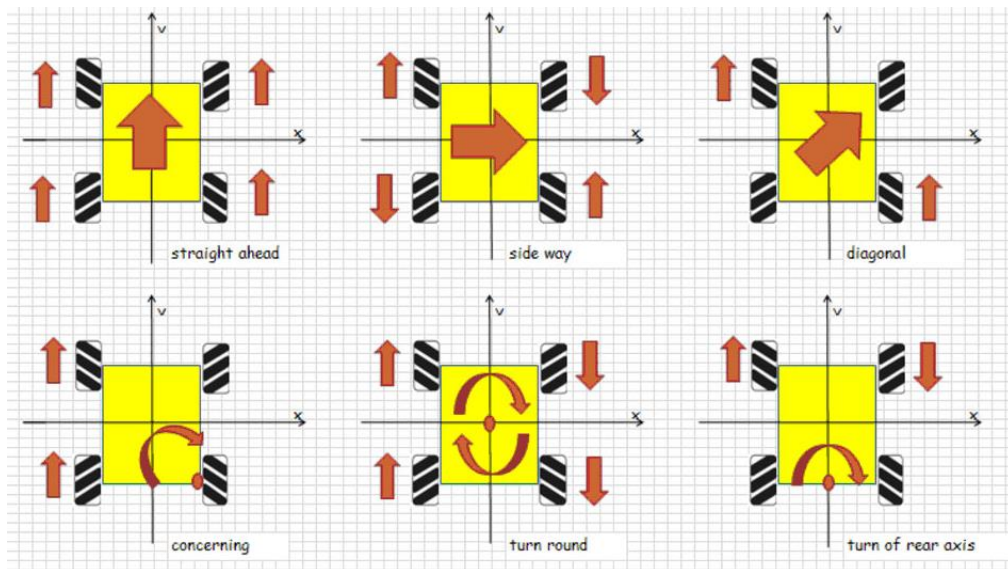
Rys.2.8. Siły wypadkowe poszczególnych kół podczas ruchu liniowego [18]

Dodatkowo istotne jest aby osie obrotowe wałków przecinały się pod kątem prostym tworząc wzór w kształcie litery „X” zgodnie z rysunkiem 2.9. Ma to na celu zmniejszenie poślizgu podczas ruchu i dzięki takiemu rozwiązaniu możliwe jest kontrolowanie siłą wypadkową napędzającą robota jedynie poprzez kontrolę kierunku obrotu napędów.



Rys.2.9. Przecięcie się osi obrotowych wałków (na podstawie [7])

Na rysunku 2.10. przedstawiono kilka przypadków ruchu wypadkowego robota w zależności od kierunków obrotu poszczególnych napędów.



Rys.2.10. Realizacja ruchu robota z kołami Mecanum [19]

## 2.3. Roboty z kołami Mecanum

Dzięki temu, że koła Mecanum umożliwiają poruszanie się w dowolnym kierunku bez konieczności zmiany orientacji całego pojazdu, są one często wykorzystywane przy konstrukcji magazynowych robotów AGV.



Rys.2.11. Przykładowy robot AGV z kołami Mecanum [33]

Istnieją również roboty posiadające więcej niż cztery koła Mecanum, tak jak widoczne to jest na rysunku 2.12.



*Rys.2.12. Robot AGV z dwunastoma kołami Mecanum [34]*

Ze względu na zmienne położenie środka ciężkości w robotach magazynowych, wynikających z ładunku, kluczowa jest odpowiednia kontrola kąt obrotu każdego z kół w celu uniknięcia poślizgu. Dodatkowym problemem w tego typu robotach jest to, że koła Mecanum wymagają częstszej konserwacji w porównaniu do ich standardowych odpowiedników.

## **2.4. Wnioski**

Po opisanu dostępnych rozwiązań zdecydowano, że najlepszą konfiguracją do projektu hybrydowego robota będzie hexapod z rysunku 2.3, ze względu na dobrą stabilność, mniejsze wymagania dotyczące momentów obrotowych silników poruszających nogami oraz niższe koszty w porównaniu do konfiguracji ośmiokończynowej. Nie zdecydowano się na konstrukcję widoczną na rysunku 2.4, ponieważ wymaga ona wykorzystania specjalistycznych materiałów konstrukcyjnych do wykonania kończyn.

Dzięki zastosowaniu sześciokończynowej konstrukcji program sterujący robotem będzie mniej skomplikowany, niż w przypadku pozostałych konfiguracji.

W celu realizacji ruchu z wykorzystaniem kół Mecanum użyte zostaną cztery koła. W zależności od żądanego kierunku ruchu robota, jednocześnie napędzać go będą przynajmniej dwa silniki co przekłada się na mniejsze wymagania dotyczące napędów. Dodatkowo podczas projektowania platformy robota należy uwzględnić odpowiedni rozstaw kół.



### **3. Założenia projektowe do realizacji hybrydowego robota kroczącego z kołami Mecanum**

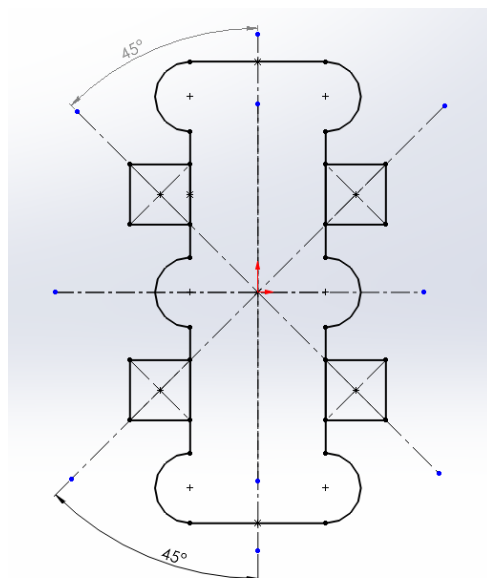
W trakcie projektowania i realizacji projektu hybrydowego robota kroczącego przyjęto poniższe założenia:

- części robota wykonane zostaną za pomocą druku 3D,
- wykorzystany zostanie mikrokontroler Arduino jako centralna jednostka sterująca,
- użyte zostaną koła Mecanum z pojazdu RC,
- zasilanie z wykorzystaniem baterii LiPO,
- sterowanie za pomocą aplikacji na telefon z wykorzystaniem technologii Bluetooth,
- dostępne dwa tryby poruszania się, którego wybór dokonuje się poprzez aplikację,
- cena całkowita robota powinna wynosić nie więcej niż 700 złotych,
- masa robota nie powinna przekraczać 3,5 kg,
- wymiary platformy robota nie powinny być większe niż: 40 x 30 cm

Na podstawie analizy dostępnych rozwiązań rozpatrzonych w podrozdziale 2.1. zdecydowano się na konstrukcję typu „hexapod” stanowiącego kompromis pomiędzy stabilnością konstrukcji i wymaganiami napędów, a ceną końcową robota.

#### **3.1. Założenia dotyczące konstrukcji robota**

Podczas projektowania platformy robota należy uwzględnić odpowiedni rozstaw kół omówiony w rozdziale 2.2.1. Dodatkowo mocowania kończyn muszą umożliwiać ich swobodny obrót w celu realizacji ruchu kroczącego. Na rysunku 3.1. przedstawiono wstępną koncepcję platformy robota w widoku z góry.



*Rys.3.1. Koncept platformy robota [opracowanie własne]*

Każda kończyna składać się będzie z trzech serwonapędów umożliwiających pełen zakres ruchu kończyny. Z racji tego, że zdecydowano się na konfigurację sześciokończynową robota wymagania dotyczące momentów obrotowych serwonapędów będą mniejsze, niż w przypadku quadraped'a. Wykorzystane koła Mecanum pochodzą z samochodu RC, zostały one przeczyszczone i wykorzystano smar zapewniający płynność obrotu wałków.

## 3.2. Wnioski

W oparciu o powyższe założenia, projektowany robot hybrydowy łączyć będzie cechy hexapoda z zaletami kół Mecanum, dzięki czemu efektem końcowym będzie stworzenie mobilnego robota zdolnego do efektywnego poruszania się po różnych powierzchniach. Wykorzystanie druku 3D oraz wtórne wykorzystanie kół Mecanum przekłada się na redukcję kosztów całkowitych robota.

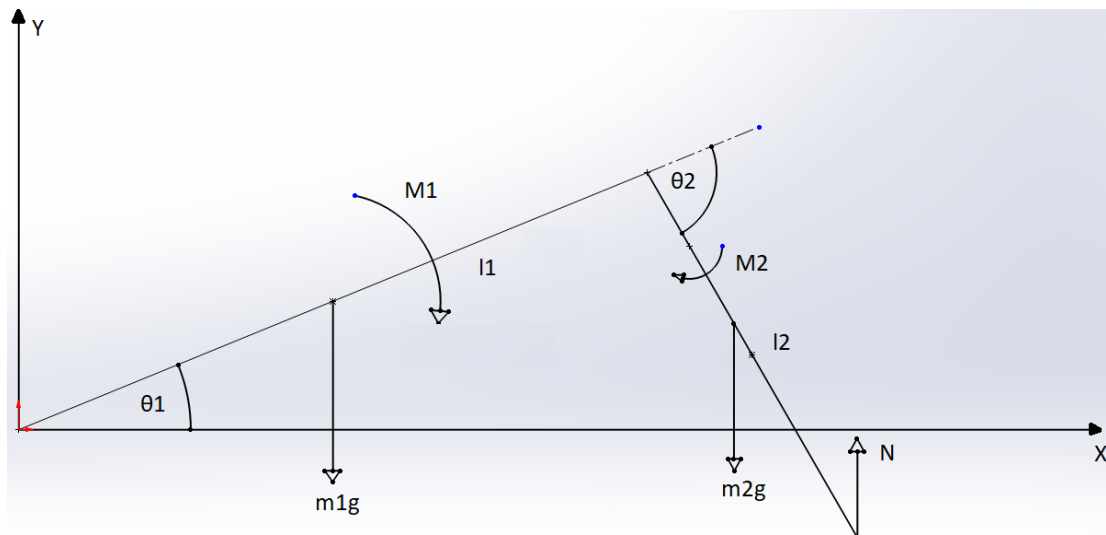


## 4. Obliczenia wymagań napędów

Wszelakie obliczenia stanowią istotny etap w procesie projektowania i jest on niezbędny w celu doboru odpowiednich napędów. Celem niniejszego rozdziału jest nie tylko przedstawienie matematycznych podstaw przyjętych rozwiązań, ale także uzasadnienie wyborów, które kierują doboorem konkretnych napędów robota.

### 4.1. Obliczenia dotyczące serwonapędów

Obliczenia dotyczące wymagań serwonapędów wykonane zostały na podstawie literatury [8].



Rys. 4.1. Siły działające na pojedynczą nogę robota (na podstawie [8])

Równania równowagi dla pojedynczej kończyny są następujące [8]:

$$M_1 + \frac{1}{2}m_1gl_1 \cos \theta_1 + m_2g[l_1 \cos \theta_1 + \frac{1}{2}l_2 \cos(\theta_2 - \theta_1)] = [l_1 \cos \theta_1 + l_2 \cos(\theta_2 - \theta_1)] \quad (4.1)$$

$$M_2 + \frac{1}{2}m_2gl_1 \cos(\theta_2 - \theta_1) = Nl_2 \cos(\theta_2 - \theta_1) \quad (4.2)$$

gdzie:  $M_1$  – moment obrotowy oddziałujący na pierwszą część nogi,

$M_2$  – moment obrotowy oddziałujący na drugą część nogi,

$m_1$  – masa części pierwszej,

$m_2$  – masa części drugiej,

- $g$  – przyspieszenie grawitacyjne,
- $l_1$  – długość części pierwszej,
- $l_2$  – długość części drugiej,
- $\theta_1$  – nachylenie między częścią pierwszą a osią  $x$ ,
- $\theta_2$  – nachylenie częścią drugą a przedłużeniem części pierwszej,
- $N$  – siła oddziaływania podłoża.

Dodatkowo masa całego robota zawsze będzie wspierana przez co najmniej trzy nogi, czyli ostatecznie otrzymujemy:

$$N = \frac{1}{3}mg \quad (4.3)$$

Masa robota wynosi około 3 kg,  $m_1=26$  g,  $m_2=24$  g,  $l_1=6,2$  cm,  $l_2=12,8$  cm,  $\theta_1$  domyślnie w pozycji stojącej wynosi  $25^\circ$ ,  $\theta_2-\theta_1$  wynosi  $60^\circ$ . Na podstawie powyższych parametrów oraz wzorów otrzymujemy, że  $N=9,81$  N,  $M_1 \approx 1$  Nm,  $M_2 \approx 0,64$  Nm.

## 4.2. Obliczenia dotyczące napędów kół Mecanum

Kolejnym etapem obliczeń jest wyznaczenie minimalnych wymogów dotyczących napędów do kół Mecanum wykonanych na podstawie pozycji [29]. Do obliczeń przyjęto masę robota równą 3kg, a współczynnik tarcia statycznego pomiędzy plastikowymi kołami, a drewnianym podłożem zgodnie ze źródłem [29] wynosi 0,4. Dodatkowo założono, że docelowe przyspieszenie wynosi  $0,1 \frac{m}{s^2}$ .

W pierwszej kolejności korzystamy ze wzoru na pierwszą zasadę dynamiki, aby określić siłę potrzebną do uzyskania docelowego przyspieszenia:

$$F_N = m * a \quad (4.4)$$

gdzie:  $F_N$  – siła wymagana do napędzenia robota,

$m$  – masa robota,

$a$  – przyspieszenie robota.

Czyli w omawianym przypadku otrzymano:

$$F_N = 3 * 0,1 = 0,3 \text{ N}$$

Następnym etapem jest obliczenie siły tarcia oddziałującego między kołami a podłożem wyliczonego ze wzoru:

$$F_T = \mu * m * g \quad (4.5)$$

gdzie:  $\mu$  - współczynnik tarcia statycznego.

Podstawiając dane do powyższego wzoru mamy:

$$F_T = 0,4 * 3 * 9,81 = 11,77 \text{ N}$$

Aby osiągnąć przyspieszenie  $1 \frac{m}{s^2}$  silnik oddziaływać musi z siłą równą sumie  $F_N$  i  $F_T$ , czyli:

$$F = F_N + F_T = 0,3 + 11,77 = 12,07 \text{ N}$$

Aby obliczyć wymaganą wartość momentu obrotowego należy wymnożyć wyliczoną powyżej siłę z promieniem koła:

$$M = F * r \quad (4.6)$$

Promień użytego koła Mecanum jest równy 3,8 cm, więc:

$$M = 12,07 * 0,0385 = 0,465 \text{ Nm}$$

Jako, że robota zawsze napędzają przynajmniej dwa silniki otrzymujemy, że moment obrotowy pojedynczego silnika powinien wynosić co najmniej 0,2325 Nm.

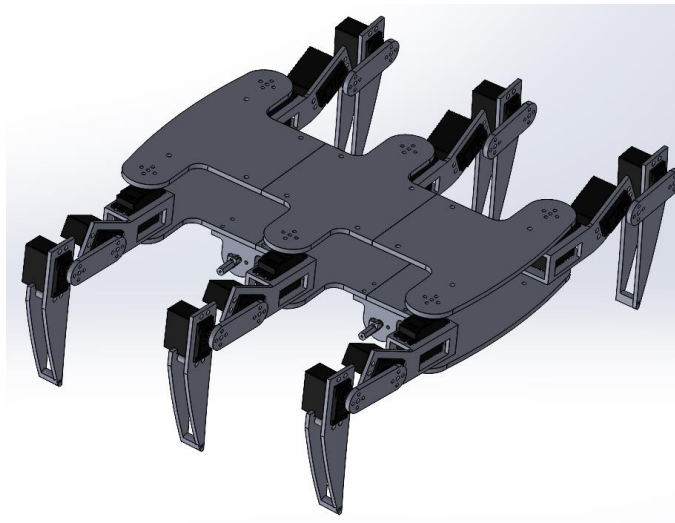
### 4.3. Wnioski

Należy pamiętać, że wyliczone wartości nie są do końca zgodne z warunkami rzeczywistymi, a wynika to z uproszczeń zastosowanych podczas obliczeń (np. współczynnik tarcia statycznego koła z podłożem w rzeczywistości może być inny). Mimo wszystko mając na uwadze otrzymane wyniki można przejść do kolejnego etapu projektowego, w tym doboru odpowiednich podzespołów oraz wykonania modelu 3D robota.



## 5. Trójwymiarowy model geometryczny robota wykonany w SolidWorks

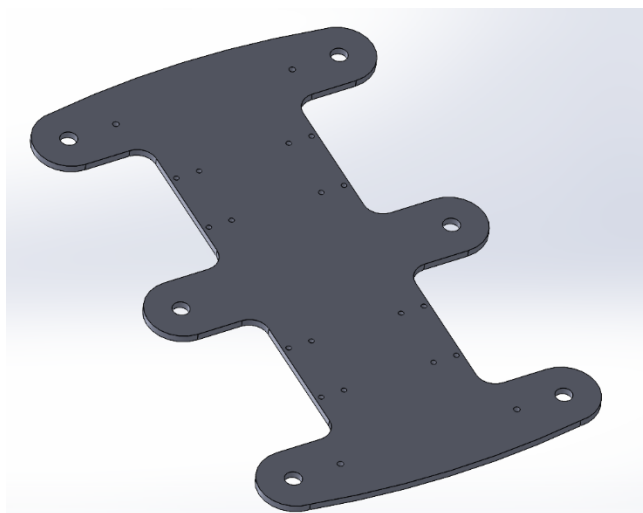
W celu fizycznej implementacji robota konieczne jest stworzenie jego trójwymiarowego modelu, na podstawie którego wydrukowane zostaną poszczególne części. W tym rozdziale przedstawione zostaną modele 3D poszczególnych komponentów wraz z ich opisami. Model ramy robota wraz z jego kończynami przedstawiono na rysunku 5.1.



*Rys.5.1. Model 3D robota [opracowanie własne]*

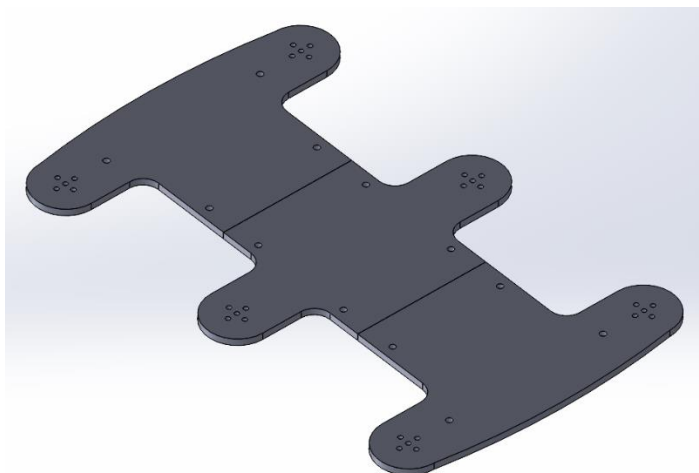
### 5.1. Platforma robota

Platforma robota, do której przymocowane zostaną nogi oraz mocowania kół składa się z dwóch głównych części – górnej oraz dolnej. Kształt ich jest jednakowy, a różnią się jedynie otworami. Do platformy dolnej, widocznej na rysunku 5.2., przykręcone zostaną mocowania do silników DC, a dodatkowo dopasowane zostaną łożyska zapewniające swobodę ruchu nóg. Jej wymiary wynoszą 36,5 x 23 cm



*Rys.5.2. Model dolnej części platformy [opracowanie własne]*

Górna część platformy, przedstawiona na rysunku 5.3, posiada odpowiednie otwory, za pomocą których przymocowane zostaną serwonapędy kończyn. Obie platformy połączone zostaną czterema śrubami zapewniając sztywność konstrukcji. Wymiary dolnej platformy wynoszą 36,5 x 23 cm.

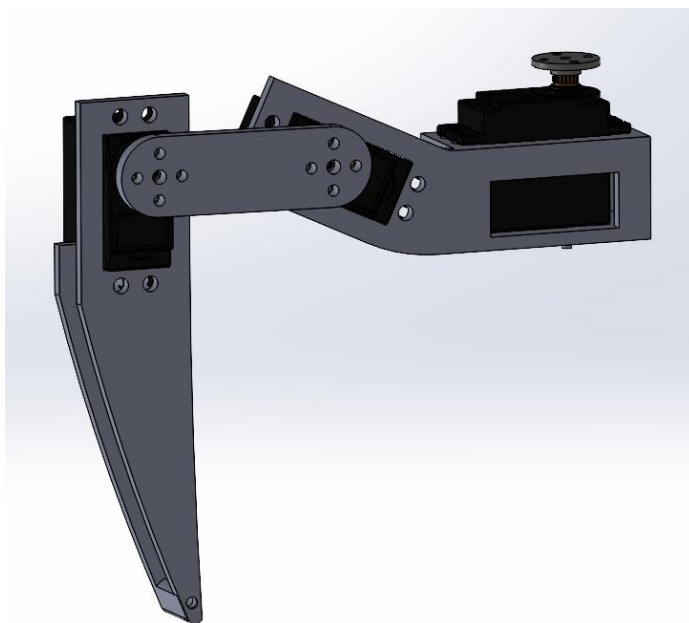


*Rys.5.3. Model górnej części platformy [opracowanie własne]*

Dodatkowo z racji tego, że platforma osiąga długość około 36,5 cm konieczne było podzielenie jej na części, aby możliwe było zmieszczenie wydruku w obszarze roboczym drukarki. Oddzielne otwory zostały uwzględnione w celu złożenia poszczególnych platform.

## 5.2. Konstrukcja pojedynczej kończyny

Robot posiadać będzie sześć sztuk nóg, złożenie pojedynczej sztuki ukazano na rysunku 5.4. Każda noga składa się z czterech wydrukowanych części oraz wykorzystano śruby M3 w celu ich skręcenia.

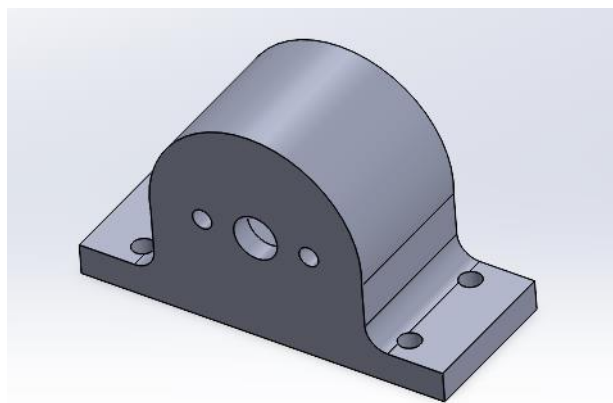


*Rys.5.4. Złożenie nogi [opracowanie własne]*

Podczas projektowania nóg uwzględniono właściwą grubość poszczególnych części, w celu zminimalizowania ich zniekształceń podczas ruchu. Szczególną uwagę poświęcono elementom mającym kontakt z podłożem, bowiem konieczne było osiągnięcie odpowiednio dużego tarcia pomiędzy elementem a podłożem, by osiągnąć płynny ruch robota. Serwonapędy, odpowiadające za ruch nogi, połączono z poszczególnymi elementami za pomocą orczyków.

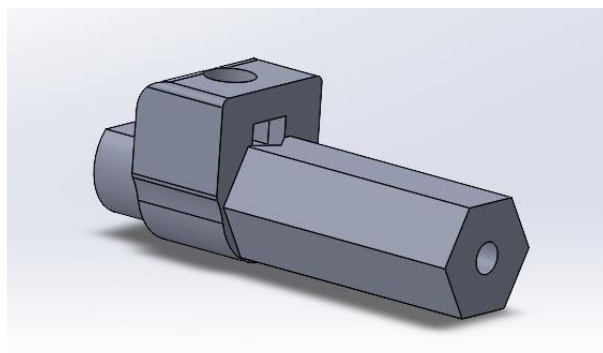
## 5.3. Mocowanie napędów kół Mecanum

Do przymocowania kół Mecanum do platformy robota zaprojektowano dwie części; Pierwszą z nich jest sam element przykręcany do platformy robota widoczny na rysunku 5.5 będący jednocześnie mocowaniem silnika DC.



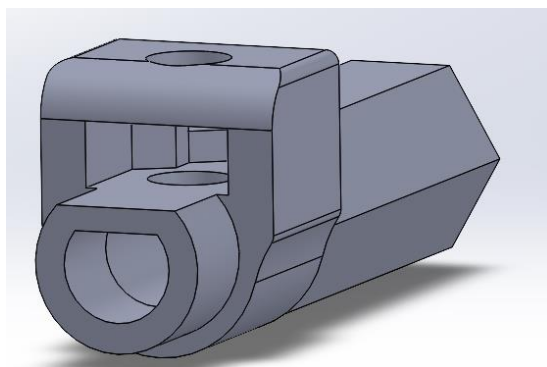
*Rys.5.5. Mocowanie silnika DC platformy [opracowanie własne]*

Drugą częścią jest adapter do koła łączący je z wałem napędowym silnika. W celu połączenia koła z adapterem zaprojektowano otwór, do którego wkręcona zostanie odpowiednia śruba. Sam element jest widoczny na rysunku 5.6.



*Rys.5.6. Adapter do koła - tył [opracowanie własne]*

W celu unieruchomienia wału silnika w adapterze wykorzystano śrubę oraz nakrętkę M3, których przeznaczone miejsce widoczne jest na rysunku 5.7.

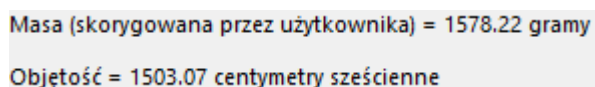


*Rys.5.7. Adapter do koła - front [opracowanie własne]*



## 5.4. Obliczenia masowe

Wykorzystano odpowiednie narzędzie w programie SolidWorks do wyznaczenia objętości modeli przeznaczonych do druku. Otrzymano, że ich objętość wynosi  $1503,07 \text{ cm}^3$ . Następnie sprawdzono gęstość żywicy, z której elementy zostaną wydrukowane, która według producenta wynosi  $1,05 \text{ g/cm}^3$ . Ostatecznie wyliczono, że całkowita masa wydrukowanych części jest równa 1578 g.



Masa (skorygowana przez użytkownika) = 1578.22 gramy  
Objętość = 1503.07 centymetry sześcienne

Rys.5.8. wyniki obliczeń masowych w SolidWorks [opracowanie własne]

## 5.5. Dokumentacja techniczna

Wszelka dokumentacja techniczna poszczególnych elementów oraz rysunek złożeniowy zostały zamieszczone w załącznikach pracy.

## 5.6. Wnioski

Wykorzystanie druku 3D, jako procesu technologicznego do produkcji części, oferuje dużą swobodę podczas projektowania modelu robota, ponieważ możliwe jest bezproblemowe wydrukowanie skomplikowanych kształtów, takich jak np. adapter widoczny na rysunku 5.7.

Możliwość dodania gęstości materiału dla elementów jest użyteczna w celu obliczenia masy poszczególnych elementów, co jest przydatne przy dalszym doborze elementów.

Podczas montażu części zakłada się wykorzystanie śrub i nakrętek M3 oraz M4, dlatego ważne jest aby odpowiednio zaprojektować otwory z uwzględnieniem ich tolerancji.



## 6. Dobór podzespołów

Dobór odpowiednich elementów jest istotnym etapem realizacji projektu. Wybór poszczególnych części podyktowany został założeniami znajdującymi się w rozdziale 3 oraz obliczeniami z rozdziału 4.

### 6.1. Mikrokontroler

W ramach realizacji projektu przyjęto mikrokontroler Arduino UNO jako centralną jednostkę sterującą. Decyzja ta oparta została na podstawie popularności i dostępności platformy Arduino, niskiej ceny oraz prostocie programowania co znacząco ułatwi implementację algorytmów sterowania i komunikację z różnymi modułami.



*Rys.6.1. Mikrokontroler Arduino UNO [20]*

## 6.2. Serwonapęd MG996R

Wybór serwomechanizmów MG996R jako napędu do nóg robota został podyktowany kilkoma czynnikami takimi jak:

- wystarczający moment obrotowy,
- odpowiednia precyzja ruchu,
- metalowe tryby przekładające się na odporność na uderzenia,
- kompatybilność z mikrokontrolerem Arduino,
- zasilanie 6V,
- dostępność na rynku i niska cena



Rys.6.2. Serwonapęd MG996R [21]

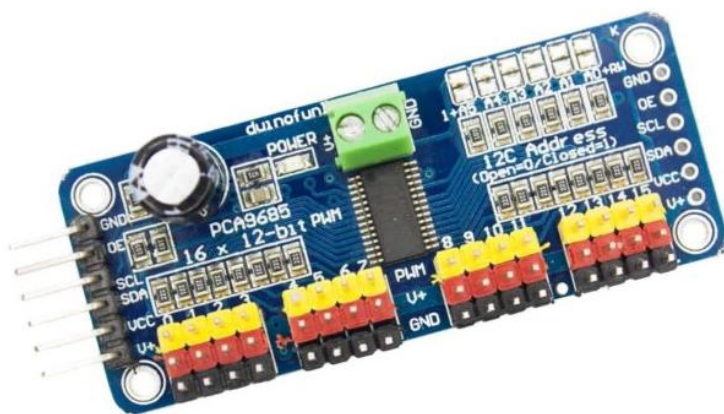
Poniżej w tabeli 6.1 przedstawiono specyfikację MG996R

Tab. 6.1. Parametry serwonapędu MG996R

Parametr	Wartość
Napięcie robocze	4,8-7,2 V
Prąd roboczy	300 mA
Moment obrotowy	1,1 Nm
Prędkość	0,14s/60° przy 6V
Waga	55 g
Kąt obrotu	180°
Napięcie sygnału sterującego	5V
Wymiary	40 x 20 x 38 mm

### 6.3. Sterownik do serwonapędów PCA9685

W ramach realizacji ruchu kroczącego robota wymagane będzie jednocześnie sterowanie aż osiemnastoma serwonapędami. W celu uproszczenia połączeń elektrycznych wykorzystane zostaną dwa sterowniki PCA9685. Jedną z zalet jest możliwość szeregowego podłączenia kilku takich modułów. Dodatkowo producent zapewnia gotowe biblioteki dedykowane sterowaniu serwonapędami. Na rysunku 6.3. przedstawiono omawiany sterownik.



Rys.6.3. Moduł PCA9685 [22]

W tabeli 6.2 przedstawiono wybrane parametry PCA9685.

Tab. 6.2. Parametry sterownika PCA9685

Parametr	Wartość
Napięcie części logicznej	3,3 – 5,0 V
Napięcie zasilania serwomechanizmów	Do 6 V
Interfejs komunikacyjny	I2C
Wymiary	62,5 x 25,4 x 9 mm

## 6.4. Silnik DC MT80 70

Idealnym napędem do kół Mecanum byłby silnik prądu stałego wraz enkoderem, jednakże cena takiego silnika wynosi około 70 złotych, więc w przypadku zastosowania czterech takich silników zostałby przekroczony budżet. W ramach kompromisu zdecydowano się wykorzystać silnik MT80, oferujący 70 obrotów na minutę, ze względu na niską cenę oraz wystarczająco wysoki moment obrotowy umożliwiający ruch robota w trybie jazdy. Ze względu na ograniczony budżet zrezygnowano również z enkodera co skutkować będzie w niewielkim poślizgiem podczas jazdy oraz brakiem możliwości płynnej jazdy. Sam silnik widoczny jest na rysunku 6.4., a w tabeli 6.3. pokazano parametry wybranego silnika



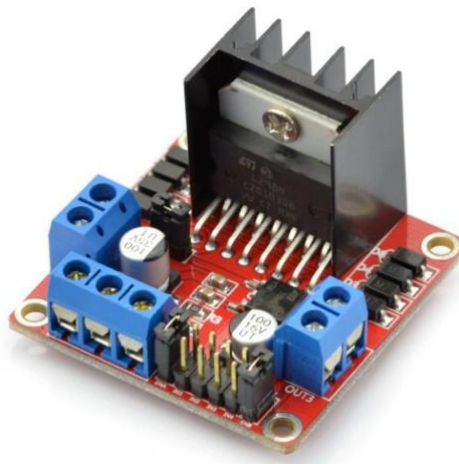
Rys.6.4. Silnik DC MT80 70 [23]

Tab. 6.3. Parametry silnika MT80 70

Parametr	Wartość
Napięcie robocze	3,0-9,0V
Prąd szczytowy	600 mA
Moment obrotowy	0,3 Nm
Obroty na minutę na biegu jałowym przy 6V	70
Waga	80 g
Średnica wału	4 mm
Długość wału	9,5 mm
Wymiary	25 x 43 mm

## 6.5. Sterownik do silników DC L298N

Na wybór sterownika L298N, widocznego na rysunku 6.5, składa się kilka czynników. Przede wszystkim umożliwia dwukierunkową kontrolę obrotu silników, co jest kluczowe w realizacji ruchu z wykorzystaniem kół Mecanum. Prostota w implementacji, zwłaszcza w połączeniu z platformą Arduino sprawia, że jest łatwy w obsłudze, a jego niska cena jest kolejnym atutem. Jedyną wadą tego rozwiązania jest stary typ mostka skutkujący dużym spadkiem napięcia (nawet do 1,8 V) między zasilaniem układu, a zasilaniem silników.



Rys.6.5. Sterownik L298N [24]

W tabeli 6.4. ukazano niektóre parametry L298N.

Tab. 6.4. Parametry sterownika L298N

Parametr	Wartość
Napięcie zasilania silników	12 V
Napięcie zasilania części logicznej	5 V
Maksymalny prąd wyjściowy	2 A
Wymiary	44 x 44 x 30 mm

## 6.6. Bateria LiPo REDOX 11,1V 3S 900mAh 20C

Wykorzystanie baterii 3S, widocznej na rysunku 6.6, oferuje napięcie około 11,1 V, co jest wystarczające do zakładanych zastosowań. Dodatkowo niska waga (65g) jest kolejnym atutem. Zwrócić należy jednak uwagę na to, że jednoczesne działanie osiemnastu serwonapędów może wymagać dużego poboru prądu. Zgodnie z tabelą 6.1 prąd roboczy pojedynczego silnika wynosi około 300 mA, czyli otrzymujemy całkowity pobór prądu:

$$300mA * 18 = 5,4A$$

Aby obliczyć prąd rozładowania akumulatora korzystamy ze wzoru:

$$C * Ah = maxA \quad (6.1)$$

gdzie: C – ocena C akumulatora podana przez producenta,

Ah – pojemność akumulatora,

maxA – maksymalny prąd rozładowania akumulatora.

Czyli w omawianym przypadku:

$$20C * 0,9Ah = 18A$$

Według powyższych obliczeń bateria 20C o pojemności 900 mAh w pełni zaspokaja omówione potrzeby.



Rys.6.6. Bateria LiPo REDOX 3S 900 mAh 20C [25]



## 6.7. Przetwornica step-down SZBK07

W celu obniżenia napięcia z baterii 3S wykorzystano między innymi przetwornice step-down SZBK07, widoczną na rysunku 6.7.



Rys.6.7. Sterownik L298N [26]

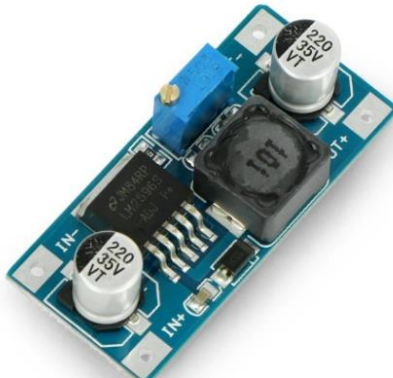
Jednoczesna praca wielu silników może pobierać dużo prądu z baterii, a wybrana przetwornica oferuje maksymalny prąd wyjściowy 20A, co jest w zupełności wystarczający w realizowanym projekcie. Posiada ona również regulację napięcia wyjściowego w zakresie od 1,2 do 36V oraz zawiera wbudowany włącznik. Więcej danych technicznych pokazano w tabeli 6.5.

Tab. 6.5. Parametry przetwornicy SZBK07

Parametr	Wartość
Zakres napięcia wejściowego	6 – 40 VDC
Zakres napięcia wyjściowego	1,2 – 36 VDC
Maksymalny prąd wyjściowy	20 A
Maksymalna moc wyjściowa	300 W
Wymiary	65 x 54 x 30 mm

## 6.8. Przetwornica step-down LM2596S

W celu zasilania między innymi sterownika serwonapędów konieczne jest osiągnięcie niższego napięcia, niż te wykorzystywane do zasilania silników DC. Osiągnięto to poprzez wykorzystanie kolejnych przetwornic widocznych na rysunku 6.8.



Rys.6.8. Przetwornica LM2596 [27]

W tabeli 6.6 widoczne są jej parametry.

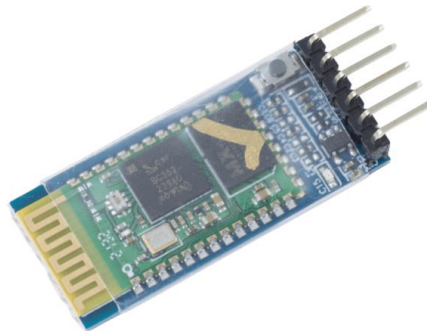
Tab. 6.6. Parametry sterownika L298N

Parametr	Wartość
Zakres napięcia wejściowego	3,2 – 35 VDC
Zakres napięcia wyjściowego	1,5 – 35 VDC
Maksymalny prąd wyjściowy	3 A
Wymiary	44 x 44 x 30 mm

Jedna taka przetwornica wykorzystywana jest do zasilania 9 serwonapędów, które zgodnie z tabelą 6.1. pobierać będą maksymalnie około 2,7 A.

## 6.9. Moduł BT HC-05

W celu realizacji komunikacji pomiędzy aplikacją na telefon, a mikrokontrolerem Arduino wykorzystano moduł HC-05, który umożliwia zarówno odbiór jak i nadawanie danych poprzez pasmo Bluetooth. Sam moduł widoczny jest na rysunku 6.7. a jego parametry podano w tabeli 6.9.



Rys.6.9. Moduł HC-05 [28]

Tab. 6.7. Parametry sterownika L298N

Parametr	Wartość
Napięcie zasilania	3,6 – 5 V
Napięcie złącza Rx	3,3 V
Moc nadajnika	+4 dBm
Pobór prądu	80 mA (parowanie); 8 mA (transmisja)
Zasięg	10 m

## 6.10. Wnioski

Właściwy dobór elementów jest niezwykle istotny w procesie tworzenia robota. Wybór mikrokontrolera Arduino, sterownika silników L298N oraz modułu Bluetooth umożliwia płynną integrację podzespołów, a dzięki wykorzystaniu sterowników PCA9685, wraz z dedykowaną biblioteką do kontroli serwonapędów, programowanie ruchu stanie się dużo prostsze. Uwzględniono również aspekt finansowy, aby zmieścić się w założonym budżecie.



## 7. Drukowanie części robota z wykorzystaniem druku 3D

W celu stworzenia konstrukcji robota na podstawie modelu 3D wykorzystano technologię druku 3D. Decyzja ta podjęta została za względu na łatwość tworzenia ewentualnych części zamiennych, szybki proces ich tworzenia oraz możliwość druku skomplikowanych kształtów. Platforma robota została wykonana z wykorzystaniem drukarki FDE, a do wydruku nóg użyto drukarki MSLA. Podczas korzystania z obydwu rodzaju drukarek korzystano z wiedzy nabytej ze źródeł [4, 5].

### 7.1. Drukarka MSLA

Do wydruku 3D z żywicy wykorzystano drukarkę Anycubic Photon M3, którą pokazano na rysunku 7.1. Drukarki żywicowe charakteryzują się większą precyzją w porównaniu do ich filamentowych odpowiedników. Sam wydruk z reguły jest jednak bardziej kruchy i mniej odporny na odkształcenia, jednak parametry te głównie zależą od wykorzystanego rodzaju żywicy. Dodatkowo dzięki zastosowaniu technologii MSLA sam proces drukowania jest dużo szybszy.



*Rys.7.1. Drukarka Anycubic Photon M3 [archiwum własne]*

Wadą tego typu drukarek jest sama żywica, której opary są toksyczne, a w kontakcie ze skórą powoduje oparzenia. Kluczowe w związku z tym jest zachowanie odpowiednich środków ostrożności takich jak zapewnienie odpowiedniej wentylacji pomieszczenia, czy też noszenie rękawic oraz okularów ochronnych. Kolejną niedogodnością jest konieczność dodatkowej obróbki druku po wyjęciu z drukarki.

Na rysunku 7.2. pokazano części wydrukowane na drukarce MSLA składające się na nogi robota.



*Rys.7.2. Wydrukowanie części nóg [archiwum własne]*

## **7.2. Drukarka FDM**

W celu uzyskania platformy robota skorzystano z drukarki filamentowej 3D Creality Ender 3 V2 Neo widoczną na rysunku 7.3.



*Rys.7.3. Drukarka 3d Creality Ender 3 V2 Neo [archiwum własne]*

Zaletą druku filamentowego jest to, że jest on bardziej odporny na pękanie w wyniku zginania i z tego powodu zdecydowano się na wykorzystać ten typ wydruku do konstrukcji platformy robota. Wspomniana drukarka posiada również duży obszar roboczy, dzięki czemu nie był konieczny podział platformy na wiele części. W przeciwieństwie do drukarek żywicznych materiał do druku nie jest toksyczny i jedynym zagrożeniem dla użytkownika jest jedynie rozgrzany plastik i dysza drukarki.

Na rysunku 7.4. pokazano drukarkę podczas drukowania elementu platformy.



*Rys.7.4. Proces drukowania platformy [archiwum własne]*

Gotowe wydruki pochodzące z drukarki FDM pokazano na rysunku 7.5.



*Rys.7.5. Gotowe wydruki [archiwum własne]*





## 7.4. Obróbka druku z żywicy

Po otrzymaniu druku z drukarki żywicznej konieczna jest jego obróbka. Proces ten składa się z oczyszczania oraz utwardzania. Oczyszczanie polega na usunięciu nadmiaru żywicy, który pozostaje na powierzchni wydruku po zakończonym procesie drukowania. Wykorzystuje się do tego alkohol izopropylowy, w którym dana część jest zanurzana i dokładnie czyszczona. Utwardzanie jest kolejnym etapem mającym na celu osiągnięcie odpowiedniej twardości materiału. Najczęściej stosuje się do tego promieniowanie UV, które utwardza żywicę. Do obydwu wyżej opisanych czynności wykorzystano stację „Anycubic Wash & Cure” przeznaczoną do czyszczenia i utwardzania wydruków widoczną na rysunku 7.8.



Rys.7.8. Anycubic Wash & Cure [archiwum własne]

## 7.5. Złożenie robota

Złożenie całkowite robota pokazano na rysunkach 7.9 oraz 7.10. Masa całkowita robota wynosi 2,9 kg, a jego wymiary w pozycji stojącej wynoszą 40 x 50 cm.



*Rys.7.9. Złożenie robota – front [archiwum własne]*



*Rys.7.10. Złożenie robota – bok [archiwum własne]*

## 7.6. Wnioski

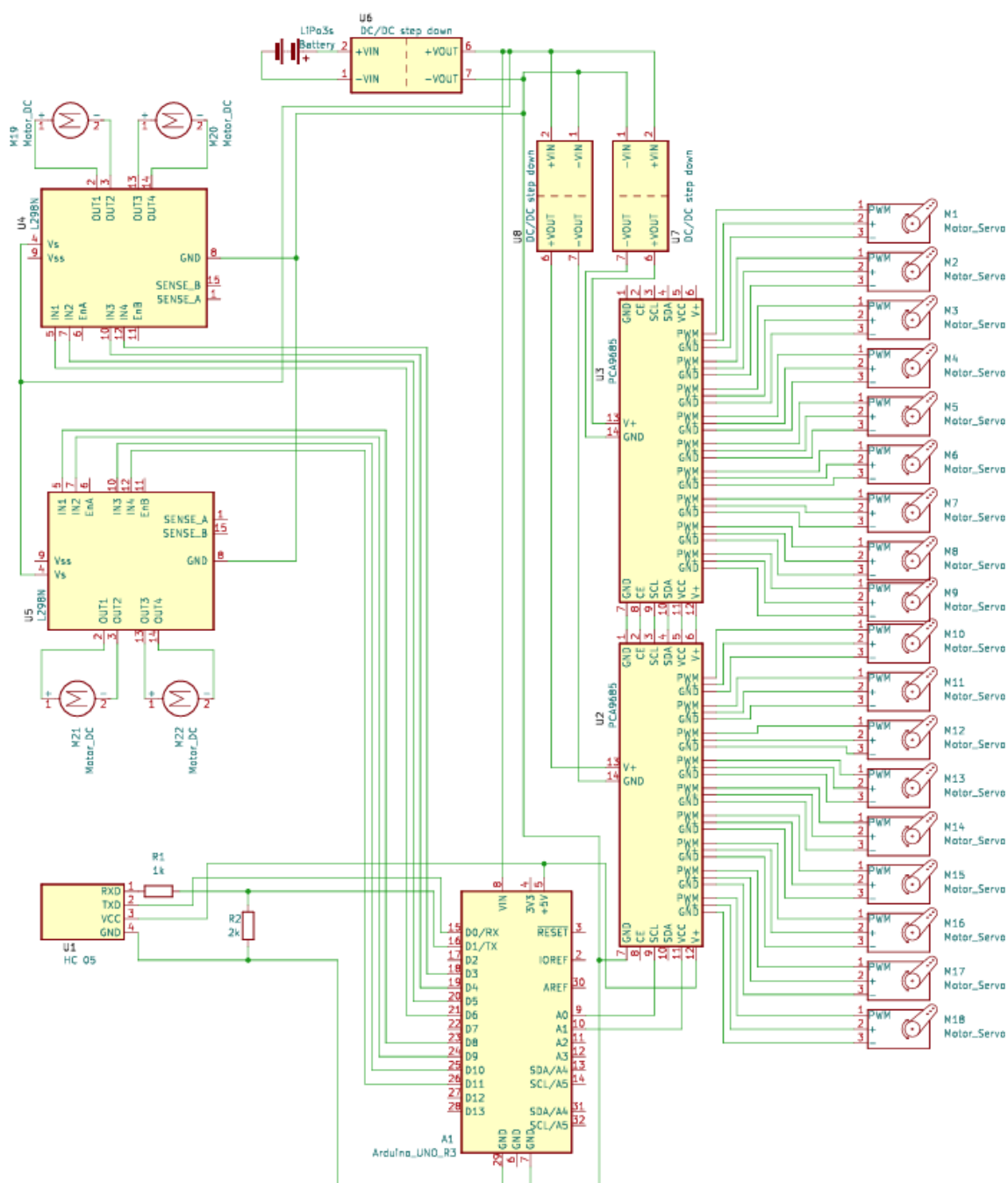
Technologia druku 3D jest niezwykle przydatna przy tworzeniu prototypowych części oraz ich zamienników. Płynne przejście między projektowaniem modelu trójwymiarowego a rozpoczęciem wydruku znacząco wpływa na szybkość pracy i jej efektywność. Ważne jest zachowanie odpowiednich środków bezpieczeństwa zwłaszcza w przypadku korzystania z drukarek żywicowych. Wydruki z żywicy są zazwyczaj kruche, więc trudne może być wywiercenie w nich otworów. Kluczowe zatem jest wcześniejsze odpowiednie zaprojektowanie wszystkich elementów. Duże wymiary platformy robota skutkowały koniecznością jej podziału na mniejsze części, aby móc zmieścić wydruk na platformie roboczej drukarki 3D.

## 8. Schemat elektryczny

Stworzenie schematu elektrycznego umożliwia wizualizację połączeń oraz jego odpowiednią implementację. Istotny on jest zwłaszcza w przypadku zastosowania większej liczby urządzeń elektrycznych w celu zminimalizowania ryzyka błędnego podłączenia elementów, a tym samym ich uszkodzenia.

### 8.1. Schemat połączeń opracowany w programie KiCAD

Schemat połączeń elementów omówionych dokładniej w rozdziale szóstym przedstawiono na rysunku 8.1. Wykorzystano dwa sterowniki PCA9685, każdy kontroluje po dziewięć serwonapędów. Zasilane są one z dwóch oddzielnych przetwornic, aby zapewnić im odpowiedni dopływ prądu. Sterowniki do silników DC oraz same Arduino zasilane zasilane są z większej przetwornicy podłączonej bezpośrednio do baterii. Moduł HC-05 zasilany jest z pinu 5 V Arduino, dodatkowo wykorzystano dzielnik napięcia, aby obniżyć je do poziomu 3,3 V na złączu RxD modułu BT.



Rys.8.1. Schemat połączeń elektrycznych [opracowanie własne]

Oznaczenia powyższego rysunku są następujące:

- A1 – mikrokontroler Arduino UNO,
- M1-M18 – serwonapędy MG996R,
- M19-M21 – silniki DC MT80 70,
- R – rezystor,
- U1 – moduł BT HC-05,
- U2-U3 – sterowniki do serwonapędów PCA9685,
- U4-U5 – sterowniki do silników DCL298N,
- U6 – przetwornica SZBK07,
- U7-U8 – przetwornice LM2596.

## 8.2. Wnioski

Duża liczba połączeń wynika z zastosowania wielu silników, trzech przetwornic i łącznie czterech sterowników. Dzięki klarownemu przedstawieniu struktury i funkcji poszczególnych elementów elektrycznych, udało się uzyskać pełny wgląd w działanie całego układu. Niezwykle przydatny okazał się być port  $V_{in}$  na płycie Arduino, dzięki któremu możliwe jest zasilanie mikrokontrolera za pomocą napięcia z zakresu 7 – 12 V.

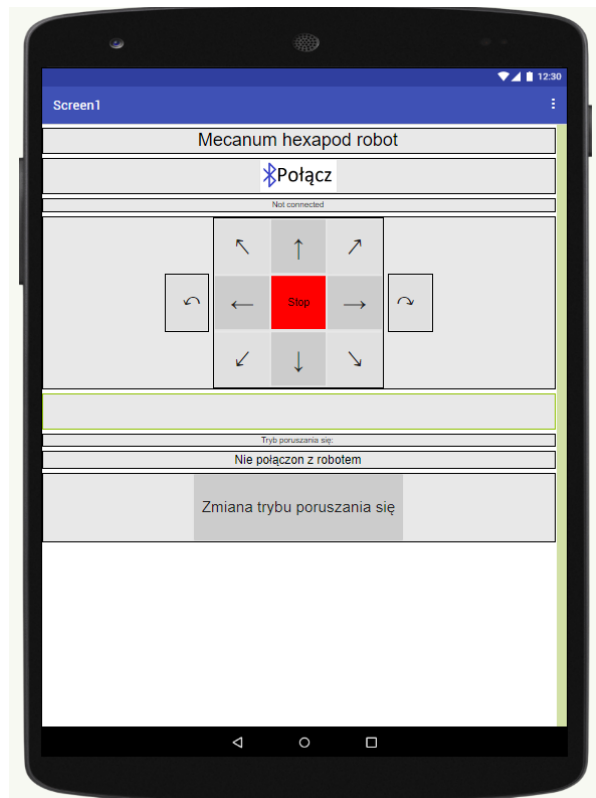


## 9. Aplikacja do sterowania robotem

By sterować robotem stworzono dedykowaną aplikację na telefon wraz z odpowiednim interfejsem. Do jej wykonania skorzystano z programu „MIT App Inventor”, którego główną zaletą jest możliwość podglądu w czasie rzeczywistym tworzonego programu. Interfejs jak i funkcja logiczna aplikacji zostały stworzone po wcześniejszym zapoznaniu się z literaturą [10, 11].

### 9.1. Oprawa graficzna aplikacji

Aplikacja posiada przycisk odpowiedzialny za rozpoczęcie protokołu komunikacji za pomocą technologii Bluetooth. Dodatkowo aplikacja na bieżąco informuje o stanie połączenia. Odpowiednie przyciski umożliwiają ruch robota w zadanym kierunku lub jego obrót. Aplikacja, widoczna na rysunku 8.1, wyświetla również obecny tryb poruszania się robota oraz umożliwia jego zmianę.

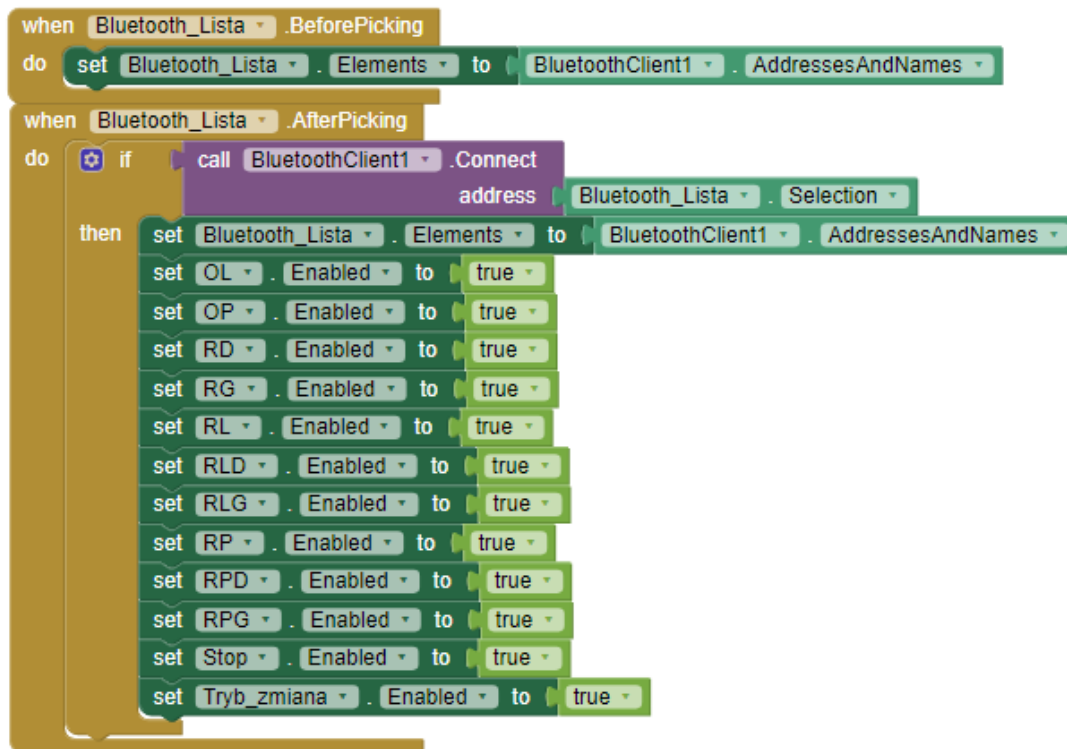


Rys.9.1. Podgląd aplikacji [opracowanie własne]

## 9.2. Część logiczna aplikacji

Aplikacja ta wykorzystuje programowanie blokowe, dzięki czemu jest prosta i intuicyjna w tworzeniu. Sama część logiczna składa się z kilku elementów:

- Obsługa połączenia Bluetooth (rys. 9.2.)

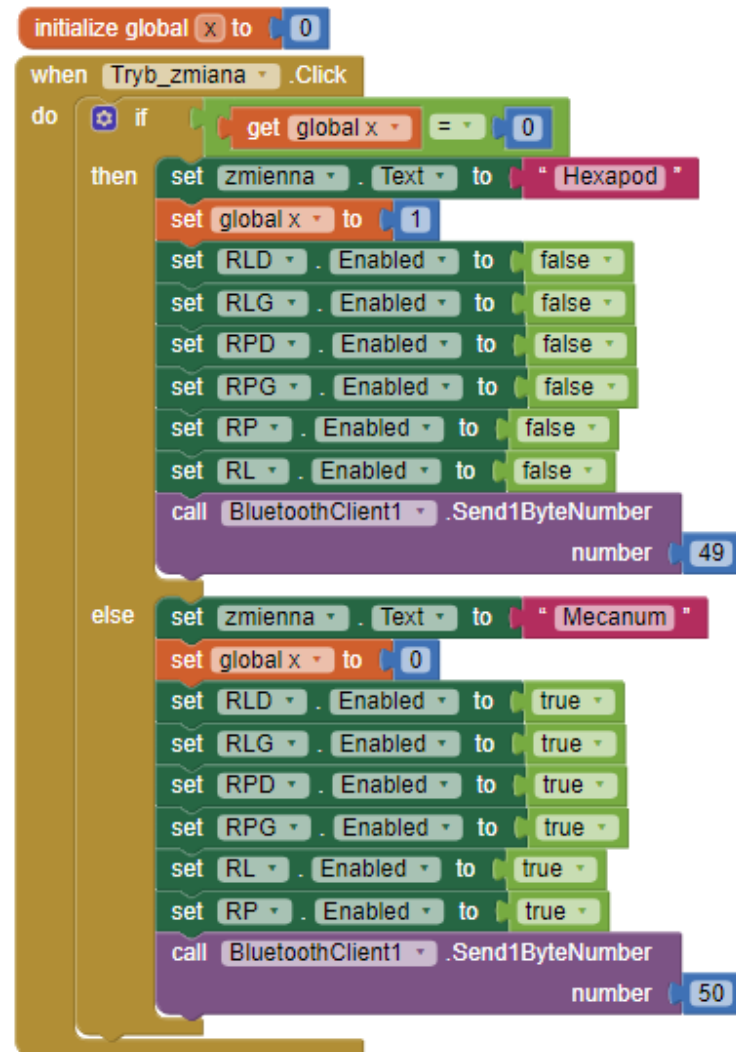


Rys.9.2. Obsługa Bluetooth [opracowanie własne]

Po wciśnięciu przycisku „Połącz” wyświetli się lista ze sparowanymi wcześniej urządzeniami, z której należy wybrać moduł HC-05. Przyciski odpowiedzialne za ruch robota domyślnie są wyłączone i aktywują się dopiero po pomyślnym połączeniu z modułem HC-05.



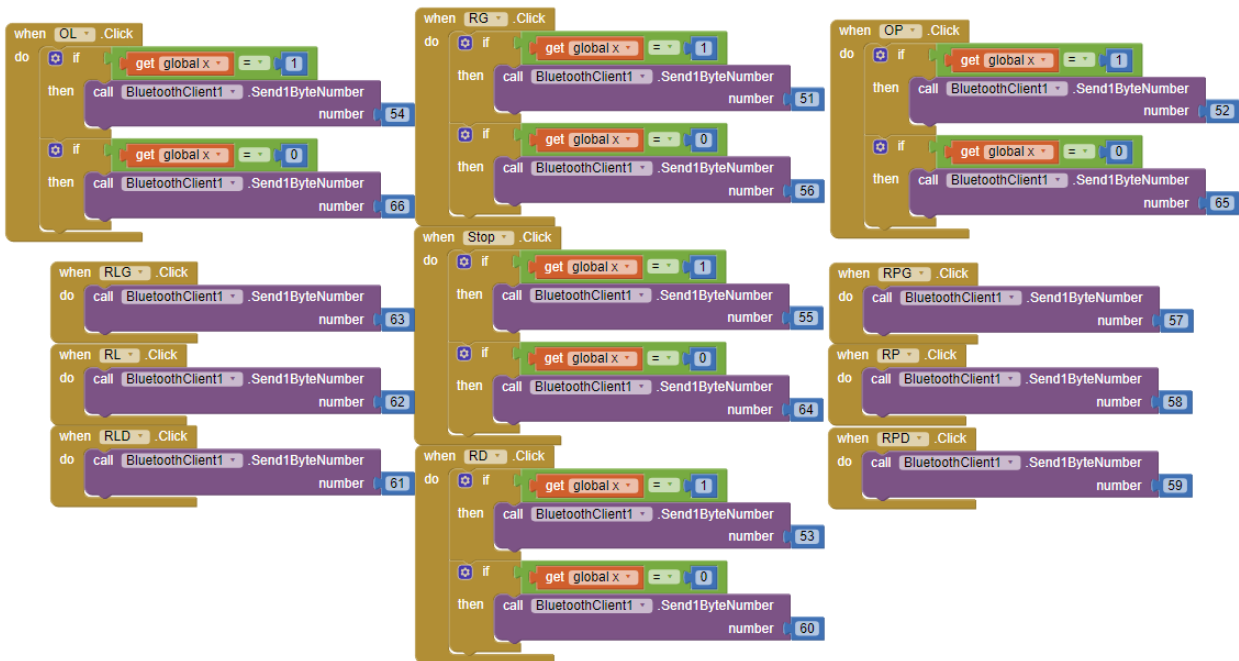
- Zmiana trybu poruszania się (rys 9.3.)



Rys.9.3. Zmiana trybu poruszania się robota [opracowanie własne]

W celu osiągnięcia dwóch trybów poruszania się wykorzystano zmienną „x”, która przechowuje obecny tryb robota. W zależności od tego, który tryb jest obecnie aktywny, aktywowane lub wyłączane są odpowiednie przyciski odpowiedzialne za ruch robota.

- Wysyłanie komend ruchu robota (rys. 9.4.)



Rys.9.4. Wysyłanie komend ruchu [opracowanie własne]

Po wciśnięciu przycisku w aplikacji wysyłany jest do modułu HC-05 znak zapisany kodem ASCII, który kolejno jest interpretowany przez program robota omówiony w rozdziale 10.

## 9.3. Wnioski

Możliwość podglądu tworzonej aplikacji w czasie rzeczywistym jest dużą zaletą programu MIT App Inventor i znacząco przyspiesza proces powstawania aplikacji. Komunikacja z modulem HC-05 umieszczonym na robocie odbywa się poprzez wysyłanie znaków w kodzie ASCII co należy uwzględnić podczas pisania programu na mikrokontrolerze Arduino. Sam program posiada wbudowane funkcje obsługi Bluetooth oraz wewnętrzne timery, dzięki czemu tworzenie aplikacji przebiega sprawnie. Tworzenie oprawy graficznej przypomina tworzenie stron internetowych za pomocą języka html – cały układ tworzy się poprzez tworzenie odpowiednich „kontenerów”, za pomocą których kontroluje się rozmieszczenie elementów.

## 10. Program sterowania robotem

Program do sterowania robotem napisany został z pomocą wiedzy ze źródeł [2, 3] z użyciem programu Arduino IDE w języku C++. Wszystkie funkcje ruchu robota zostały zapisane jako osobne funkcje (np. ruch do przodu w trybie chodzenia). Po uruchomieniu robota w pierwszej kolejności wykonywana jest funkcja „setup”, której zadaniem jest przygotowanie robota do ruchu (ustawienie odpowiednich pinów jako wyjścia, ustalenie częstotliwości komunikacji z serwami). W głównej funkcji programu znajduje się obsługa komunikacji Bluetooth oraz wywołanie odpowiedniej funkcji ruchu w zależności od danych otrzymanych z aplikacji na telefon. Kod źródłowy zamieszczono w załącznikach pracy.

### 10.1. Sterowanie częścią kroczącą robota

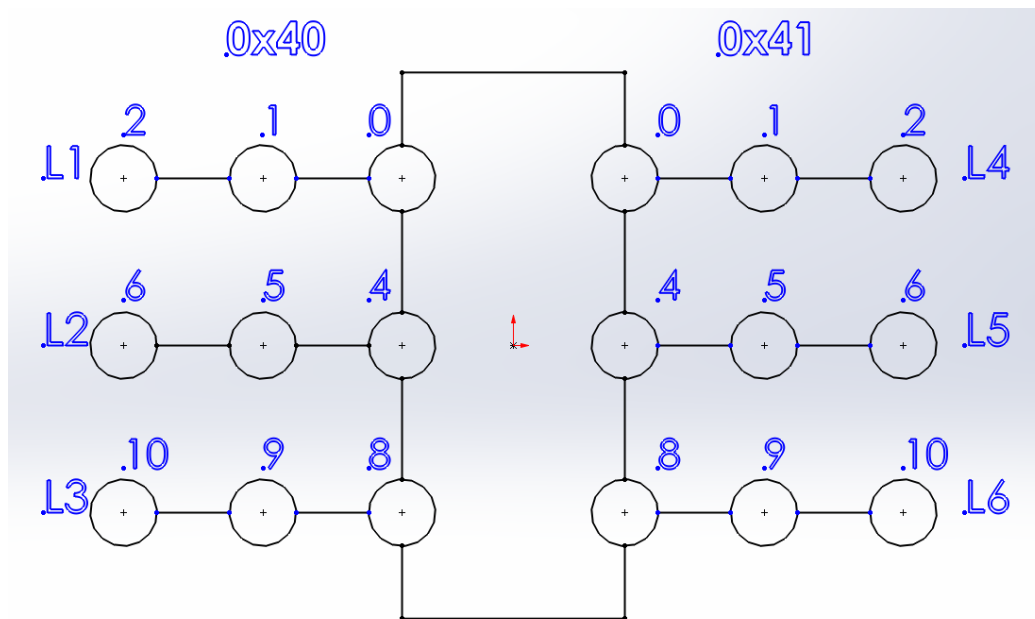
Sterowanie serwonapędami odbywa się z wykorzystaniem sterowników PCA9685, do których dostępne są dedykowane biblioteki oraz poradniki od firmy Adafruit [13]. Możliwe jest ich szeregowe podłączenie, należy wtedy je jednak odpowiednio zaadresować w programie co pokazano na wydruku 10.1.

*Program 10.1. Adresowanie modułów PCA9685.*

```
Adafruit_PWMServoDriver pwm1 = Adafruit_PWMServoDriver(0x40);  
Adafruit_PWMServoDriver pwm2 = Adafruit_PWMServoDriver(0x41);
```

Adresowanie modułów polega na zlutowaniu odpowiednich zworek na samym module. Pierwszy moduł ma adres 0x40, drugi 0x41, a w przypadku podłączenia trzeciego miałby on adres 0x42 [12].

Realizacja ruchu robota została podzielona na trzy etapy, a nogi podzielone na dwie grupy składające się z trzech elementów. Na rysunku 10.1 przedstawiono numerację kończyn oraz poszczególnych serwonapędów, a wynika ona ze sposobu podłączenia poszczególnych silników do modułów PCA9685.



Rys. 10.1. Oznaczenia kończyn i serwonapędów [opracowanie własne]

Podział ruchu zostanie przedstawiony na przykładzie funkcji umożliwiającej ruch robota do przodu. Kończyny nr 1, 3 i 5 należą do pierwszej grupy, a nogi 2, 4, 6 do drugiej. Pierwszy etap ruchu polega na podniesieniu nóg z grupy pierwszej, przesunięciu ich do przodu z pozycji stojącej spoczynkowej i opuszczeniu. Nogi z grupy drugiej jedynie są przesuwane w przeciwnym kierunku. Etap 1 w programie przedstawiono na wydruku 10.2.

Program 10.2. Etap pierwszy funkcji „move\_frwd”

```

if (FM1<=10){
    pwm1.setPWM(0, 0, angleToPulse(hl1s1 + FM1) ); //ruch nogi 1 do przodu
    pwm1.setPWM(1, 0, angleToPulse(hl1s2 + 2*FM1+offset) ); //podniesienie nogi 1
    pwm1.setPWM(2, 0, angleToPulse(hl1s3 - 2*FM1) ); //podniesienie nogi 1

    pwm1.setPWM(8, 0, angleToPulse(hl3s1 + FM1) ); //ruch nogi 3 do przodu
    pwm1.setPWM(9, 0, angleToPulse(hl3s2 + 2*FM1+offset)); //podniesienie nogi 3
    pwm1.setPWM(10, 0, angleToPulse(hl3s3 - 2*FM1) ); //postawienie nogi 3

    pwm2.setPWM(4, 0, angleToPulse(hl5s1 - FM1) ); //ruch nogi 5 do przodu
    pwm2.setPWM(5, 0, angleToPulse(hl5s2 - 2*FM1+offset)); //podniesienie nogi 5
    pwm2.setPWM(6, 0, angleToPulse(hl5s3 + 2*FM1) ); //postawienie nogi 5

    pwm1.setPWM(4, 0, angleToPulse(hl2s1 - FM1) ); //ruch nogi 2 do tyłu
    pwm2.setPWM(0, 0, angleToPulse(hl4s1 + FM1) ); //ruch nogi 4 do tyłu
    pwm2.setPWM(8, 0, angleToPulse(hl6s1 + FM1) ); //ruch nogi 6 do tyłu

    FM1++;
    FM1++;
}
if (FM1<=20 && FM1>10){
    pwm1.setPWM(0, 0, angleToPulse(hl1s1 + FM1) ); //ruch nogi 1 do przodu
    pwm1.setPWM(1, 0, angleToPulse((hl1s2+20) - ((2*FM1)-20))); //podniesienie
nogi 1
    pwm1.setPWM(2, 0, angleToPulse((hl1s3-20) + ((2*FM1)-20))); //podniesienie
nogi 1
    pwm1.setPWM(8, 0, angleToPulse(hl3s1 + FM1) ); //ruch nogi 3 do przodu

```

```

    pwm1.setPWM(9, 0, angleToPulse((hl3s2+20) - ((2*FM1)-20)) ); //postawienie
nogi 3
    pwm1.setPWM(10, 0, angleToPulse((hl3s3-20) + ((2*FM1)-20)) ); //postawienie
nogi 3

    pwm2.setPWM(4, 0, angleToPulse(hl5s1 - FM1) ); //ruch nogi 5 do przodu
    pwm2.setPWM(5, 0, angleToPulse((hl5s2-20) + ((2*FM1)-20)) ); //postawienie
nogi 5
    pwm2.setPWM(6, 0, angleToPulse((hl5s3+20) - ((2*FM1)-20)) ); //postawienie
nogi 5
    pwm1.setPWM(4, 0, angleToPulse(hl2s1 - FM1) ); //ruch nogi 2 do tyłu
    pwm2.setPWM(0, 0, angleToPulse(hl4s1 + FM1) ); //ruch nogi 4 do tyłu
    pwm2.setPWM(8, 0, angleToPulse(hl6s1 + FM1) ); //ruch nogi 6 do tyłu

    FM1++;
    FM1++;
}

```

Etap drugi polega na przesunięciu kończyn z grupy pierwszej do tyłu i wykonaniu kroku grupy drugiej do przodu. Kod przedstawiono na wydruku 10.3.

Program 10.3. Etap drugi funkcji „move\_frwd”

```

if (FM2<=20){
    pwm1.setPWM(4, 0, angleToPulse(hl2s1-20 + FM2) ); //ruch nogi 2 do przodu
    pwm1.setPWM(5, 0, angleToPulse(hl2s2 + FM2 + offset) ); //podniesienie nogi 2
    pwm1.setPWM(6, 0, angleToPulse(hl2s3 - FM2) ); //podniesienie nogi 2

    pwm2.setPWM(0, 0, angleToPulse(hl4s1+20 - FM2) ); //ruch nogi 4 do przodu
    pwm2.setPWM(1, 0, angleToPulse(hl4s2 - FM2 - offset) ); //podniesienie nogi 4
    pwm2.setPWM(2, 0, angleToPulse(hl4s3 + FM2) ); //podniesienie nogi 4

    pwm2.setPWM(8, 0, angleToPulse(hl6s1+20 - FM2) ); //ruch nogi 6 do przodu
    pwm2.setPWM(9, 0, angleToPulse(hl6s2 - FM2 - offset) ); //podniesienie nogi 6
    pwm2.setPWM(10, 0, angleToPulse(hl6s3 + FM2) ); //podniesienie nogi 6

    pwm1.setPWM(0, 0, angleToPulse((hl1s1+FM1) - FM2) ); //ruch nogi 1 do tyłu
    pwm1.setPWM(8, 0, angleToPulse((hl3s1+FM1) - FM2) ); //ruch nogi 3 do tyłu
    pwm2.setPWM(4, 0, angleToPulse((hl5s1-FM1) + FM2) ); //ruch nogi 5 do tyłu
    FM2++;
    FM2++;
}
if (FM2<=40 && FM2>20){
    pwm1.setPWM(4, 0, angleToPulse(hl2s1-20 + FM2) ); //ruch nogi 2 do przodu
    pwm1.setPWM(5, 0, angleToPulse((hl2s2+20) - (FM2-20)) ); //postawienie nogi 2
    pwm1.setPWM(6, 0, angleToPulse((hl2s3-20) + (FM2-20)) ); //postawienie nogi 2

    pwm2.setPWM(0, 0, angleToPulse(hl4s1+20 - FM2) ); //ruch nogi 4 do przodu
    pwm2.setPWM(1, 0, angleToPulse((hl4s2-20) + (FM2-20)) ); //postawienie nogi 4
    pwm2.setPWM(2, 0, angleToPulse((hl4s3+20) - (FM2-20)) ); //postawienie nogi 4

    pwm2.setPWM(8, 0, angleToPulse(hl6s1+20 - FM2) ); //ruch nogi 6 do przodu
    pwm2.setPWM(9, 0, angleToPulse((hl6s2-20) + (FM2-20)) ); //postawienie nogi 6
    pwm2.setPWM(10, 0, angleToPulse((hl6s3+20) - (FM2-20)) ); //postawienie nogi 6

    pwm1.setPWM(0, 0, angleToPulse((hl1s1+FM1) - FM2) ); //ruch nogi 1 do tyłu
    pwm1.setPWM(8, 0, angleToPulse((hl3s1+FM1) - FM2) ); //ruch nogi 3 do tyłu
    pwm2.setPWM(4, 0, angleToPulse((hl5s1-FM1) + FM2) ); //ruch nogi 5 do tyłu

    FM2++;
    FM2++;
}

```

Etap trzeci, przedstawiony na wydruku 10.4, polega na wykonaniu odwrotnych czynności z etapu drugiego, czyli grupa pierwsza wykonuje krok do przodu a grupa druga przesuwana jest do tyłu.

Program 10.4. Etap drugi funkcji „move\_frwd”

```

if (FM3<=20){
    pwm1.setPWM(0, 0, angleToPulse(hl1s1-20 + FM3) ); //ruch nogi 1 do przodu
    pwm1.setPWM(1, 0, angleToPulse(hl1s2 + FM3 + offset) ); //podniesienie nogi 1
    pwm1.setPWM(2, 0, angleToPulse(hl1s3 - FM3) ); //podniesienie nogi 1

    pwm1.setPWM(8, 0, angleToPulse(hl3s1-20 + FM3) ); //ruch nogi 3 do przodu
    pwm1.setPWM(9, 0, angleToPulse(hl3s2 + FM3 + offset) ); //podniesienie nogi 3
    pwm1.setPWM(10, 0, angleToPulse(hl3s3 - FM3) ); //podniesienie nogi 3

    pwm2.setPWM(4, 0, angleToPulse(hl5s1+20 - FM3) ); //ruch nogi 5 do przodu
    pwm2.setPWM(5, 0, angleToPulse(hl5s2 - FM3 - offset) ); //podniesienie nogi 5
    pwm2.setPWM(6, 0, angleToPulse(hl5s3 + FM3) ); //podniesienie nogi 5

    pwm1.setPWM(4, 0, angleToPulse((hl2s1+20) - FM3) ); //ruch nogi 2 do tyłu
    pwm2.setPWM(0, 0, angleToPulse((hl4s1-20) + FM3) ); //ruch nogi 4 do tyłu
    pwm2.setPWM(8, 0, angleToPulse((hl6s1-20) + FM3) ); //ruch nogi 6 do tyłu

    FM3++;
    FM3++;
}
if (FM3<=40 && FM3>20){
    pwm1.setPWM(0, 0, angleToPulse(hl1s1-20 + FM3) ); //ruch nogi 1 do przodu
    pwm1.setPWM(1, 0, angleToPulse((hl1s2+20) - (FM3-20)) ); //postawienie nogi 1
    pwm1.setPWM(2, 0, angleToPulse((hl1s3-20) + (FM3-20)) ); //postawienie nogi 1

    pwm1.setPWM(8, 0, angleToPulse(hl3s1-20 + FM3) ); //ruch nogi 3 do przodu
    pwm1.setPWM(9, 0, angleToPulse((hl3s2+20) - (FM3-20)) ); //postawienie nogi 3
    pwm1.setPWM(10, 0, angleToPulse((hl3s3-20) + (FM3-20)) ); //postawienie nogi

    pwm2.setPWM(4, 0, angleToPulse(hl5s1+20 - FM3) ); //ruch nogi 5 do przodu
    pwm2.setPWM(5, 0, angleToPulse((hl5s2-20) + (FM3-20)) ); //postawienie nogi 5
    pwm2.setPWM(6, 0, angleToPulse((hl5s3+20) - (FM3-20)) ); //postawienie nogi 5

    pwm1.setPWM(4, 0, angleToPulse((hl2s1+20) - FM3) ); //ruch nogi 2 do tyłu
    pwm2.setPWM(0, 0, angleToPulse((hl4s1-20) + FM3) ); //ruch nogi 4 do tyłu
    pwm2.setPWM(8, 0, angleToPulse((hl6s1-20) + FM3) ); //ruch nogi 6 do tyłu
}

```

Po wykonaniu trzeciego etapu, poprzez zastosowanie odpowiednich warunków, zapętlane zostają etapy 2 i 3, dzięki czemu otrzymywany jest ruch robota. Inne funkcje odpowiedzialne za ruch również zostały podzielone na trzy etapy.

## 10.2. Sterowanie napędami kół Mecanum

Sterowanie trybu jazdy odbywa się poprzez kontrolę kierunku obrotu poszczególnych kół. Dzięki zastosowaniu silników DC wystarczy jedynie zmienić kierunek przepływu prądu, co realizowane jest za pomocą sterownika L298N. Zasada sterowania dokładnie została opisana w podrozdziale 2.2.1. Na wydruku 10.5 przedstawiono funkcję umożliwiającą jazdę do przodu.

*Program 10.6. Realizacja ruchu do przodu*

```
void ride_frwd(){
    digitalWrite(m1_1, HIGH);
    digitalWrite(m1_2, LOW);

    digitalWrite(m2_1, HIGH);
    digitalWrite(m2_2, LOW);

    digitalWrite(m3_1, LOW);
    digitalWrite(m3_2, HIGH);

    digitalWrite(m4_1, LOW);
    digitalWrite(m4_2, HIGH);
} //End Ride Forward
```

Inne funkcje różnią się jedynie stanami wyjść poszczególnych pinów. Łącznie możliwy jest ruch w ośmiu kierunkach oraz obrót robota w lewo i prawo.

## 10.3. Zmiana trybu poruszania się

W zależności od obecnego trybu poruszania się robota, po wciśnięciu przycisku „zmiana trybu poruszania się” w aplikacji wywoływana jest odpowiednia funkcja. Gdy robot znajduje się w trybie chodzenia po wciśnięciu przycisku realizowana jest funkcja przedstawiona na wydruku 10.7 polegająca na wyśrodkowaniu nóg i ich uniesieniu.

*Program 10.7. Przejście w tryb jazdy*

```
void get_down(){
    while (i>0){
        pwm1.setPWM(1, 0, angleToPulse(hl1s2 + i) ); //podniesienie nogi 1
        pwm1.setPWM(5, 0, angleToPulse(hl2s2 + i) ); //podniesienie nogi 2
        pwm1.setPWM(9, 0, angleToPulse(hl3s2 + i) ); //podniesienie nogi 3

        pwm2.setPWM(1, 0, angleToPulse(hl4s2 - i) ); //podniesienie nogi 4
        pwm2.setPWM(5, 0, angleToPulse(hl5s2 - i) ); //podniesienie nogi 5
        pwm2.setPWM(9, 0, angleToPulse(hl6s2 - i) ); //podniesienie nogi 6
        i--;
    }
}
```

```

    pwm1.setPWM(0, 0, angleToPulse(hl1s1) );
    pwm1.setPWM(2, 0, angleToPulse(hl1s3) );
    pwm1.setPWM(4, 0, angleToPulse(hl2s1) );
    pwm1.setPWM(6, 0, angleToPulse(hl2s3) );
    pwm1.setPWM(8, 0, angleToPulse(hl3s1) );
    pwm1.setPWM(10, 0, angleToPulse(hl3s3) );
    pwm2.setPWM(0, 0, angleToPulse(hl4s1) );
    pwm2.setPWM(2, 0, angleToPulse(hl4s3) );
    pwm2.setPWM(4, 0, angleToPulse(hl5s1) );
    pwm2.setPWM(6, 0, angleToPulse(hl5s3) );
    pwm2.setPWM(8, 0, angleToPulse(hl6s1) );
    pwm2.setPWM(10, 0, angleToPulse(hl6s3) );
    delay(60);
}
}

```

Przejsie z trybu jazdy w tryb chodzenia realizowane jest dzięki funkcji z wydruku 10.8 i działa w analogiczny sposób co wyżej omówiony fragment kodu.

#### *Program 10.8. Przejsie w tryb chodzenia*

```

void get_up() {
    while (i<75){
        pwm1.setPWM(1, 0, angleToPulse(hl1s2 + i) ); //opuszczenie nogi 1
        pwm1.setPWM(5, 0, angleToPulse(hl2s2 + i) ); //opuszczenie nogi 2
        pwm1.setPWM(9, 0, angleToPulse(hl3s2 + i) ); //opuszczenie nogi 3

        pwm2.setPWM(1, 0, angleToPulse(hl4s2 - i) ); //opuszczenie nogi 4
        pwm2.setPWM(5, 0, angleToPulse(hl5s2 - i) ); //opuszczenie nogi 5
        pwm2.setPWM(9, 0, angleToPulse(hl6s2 - i) ); //opuszczenie nogi 6
        i++;
        //Wyśrodkowanie nóg
        pwm1.setPWM(0, 0, angleToPulse(hl1s1) );
        pwm1.setPWM(2, 0, angleToPulse(hl1s3) );
        pwm1.setPWM(4, 0, angleToPulse(hl2s1) );
        pwm1.setPWM(6, 0, angleToPulse(hl2s3) );
        pwm1.setPWM(8, 0, angleToPulse(hl3s1) );
        pwm1.setPWM(10, 0, angleToPulse(hl3s3) );
        pwm2.setPWM(0, 0, angleToPulse(hl4s1) );
        pwm2.setPWM(2, 0, angleToPulse(hl4s3) );
        pwm2.setPWM(4, 0, angleToPulse(hl5s1) );
        pwm2.setPWM(6, 0, angleToPulse(hl5s3) );
        pwm2.setPWM(8, 0, angleToPulse(hl6s1) );
        pwm2.setPWM(10, 0, angleToPulse(hl6s3) );
        delay(60);
    }
}

```



## 10.4. Komunikacja z wykorzystaniem Bluetooth

Komunikacja odbywa się tylko w jednym kierunku - z aplikacji do modułu HC-05, w związku z czym program jedynie oczekuje na otrzymanie informacji, która jest zapisywana w odpowiedniej zmiennej. Odbywa się to za pomocą funkcji widocznej na wydruku 10.9, która znajduje się w głównej funkcji programu.

*Program 10.9. Odczytywanie informacji z aplikacji*

```
if(Serial.available()>0){ //Sprawdź czy otrzymano dane z modułu HC-05
    status=Serial.read(); //Przypisanie informacji z modułu HC-05 do zmiennej
}
```

## 10.5. Wnioski

Pomimo tego, że podczas programowania ruchu kroczącego robota wykorzystano kinematykę prostą, proces ten był czasochłonny ze względu na złożoność ruchu i konieczność kontrolowania jednocześnie osiemnastu serwonapędów. Istotna była także odpowiednia kalibracja wszystkich silników, ponieważ błąd w ich pozycjonowaniu mógłby wpływać negatywnie na ruch robota. Zmiana trybu poruszania się polega na wyśrodkowaniu kończyn i ich podniesieniu/opuszczeniu w zależności od obecnego trybu. Kontrola kierunku obrotu kół, konieczna do realizacji trybu jazdy, odbywa się poprzez odpowiednie wysterowanie stanów poszczególnych pinów mikrokontrolera. Odpowiednie funkcje wywoływane są na podstawie zmiennej „status”, w której przechowywany jest znak pochodzący z aplikacji.



## 11. Testy hybrydowego robota krocącego z kołami Mecanum

Wykonano testy robota polegające na określeniu jego możliwości. We wszystkich testach prędkości wykonano pięć pomiarów, na których podstawie wyliczono średnią będącą wynikiem testu.

### 11.1. Testy trybu krocącego

Wykonano cztery testy dla trybu krocącego mające na celu określenie prędkości poruszania się robota w trybie kroczenia. W pierwszej kolejności wykonano pomiar prędkości ruchu oraz obrotu na płaskiej powierzchni (panele podłogowe) następnie testy powtórzono na wykładzinie. Trasa do przebycia w pierwszym teście wynosi dwa metry, w przypadku obrotu – 720 stopni. Zdecydowano się na takie wartości, aby zminimalizować błąd wynikający z reakcji osoby mierzącej wyniki, ponieważ pomiar czasu dokonano z wykorzystaniem stopera.

- Chód po powierzchni płaskiej

Tab. 11.1. Test prędkości chodu po powierzchni płaskiej

Nr testu	Czas [s]	Wyliczona prędkość [m/s]
1	19,8	0,1
2	20,4	
3	19,3	
4	20,0	
5	20,3	

- Obrót na powierzchni płaskiej

Tab. 11.2. Test prędkości obrotu po powierzchni płaskiej

Nr testu	Czas [s]	Wyliczona prędkość [°/s]
1	36,0	19,92
2	36,6	

3	35,8	
4	36,2	
5	36,1	

Testy wykonane na płaskiej powierzchni przebiegły bez zarzutów – robot pracował płynnie, a serwonapędy nie drgały podczas ruchu.

- Chód na wykładzinie

*Tab. 11.3. Testy prędkości chodu po wykładzinie*

Nr testu	Czas [s]	Wyliczona prędkość [m/s]
1	28,4	0,072
2	27,3	
3	28,9	
4	26,9	
5	27,6	

- Obrót na wykładzinie

*Tab. 11.4. Test prędkości obrotu po wykładzinie*

Nr testu	Czas [s]	Wyliczona prędkość [°/s]
1	41,0	17,31
2	42,2	
3	41,6	
4	40,9	
5	42,3	

Testy chodu na wykładzinie okazały się być dużym wyzwaniem dla robota. Nogi czasem się blokowały przez co przebycie trasy zajmowało dużo więcej czasu w porównaniu do testu na powierzchni płaskiej. Powodowało to również większy rozrzut wyników. Zaobserwowano również większą tendencję serwonapędów do drgań wynikających z ciągłego zaczepiania o podłoże.

## 11.2. Testy trybu jazdy

Dla trybu jazdy wykonano cztery testy:

1. prędkość jazdy prosto,
2. prędkość jazdy po skosie,
3. prędkość jazdy bokiem,
4. prędkość obrotu robota .

W trzech pierwszych przypadkach robot do przebycia miał 2m, w ostatnim teście mierzony był czas dwóch pełnych obrotów robota.

- Jazda prosto

Tab. 11.5. Test prędkości jazdy do przodu

Nr testu	Czas [s]	Wynik [m/s]
1	7,4	0,27
2	7,6	
3	7,4	
4	7,3	
5	7,1	

Tryb jazdy jest szybszy o około 37% w porównaniu do kroczenia, ale możliwy jest jedynie na płaskiej powierzchni.

- Jazda po skosie

Tab. 11.6. Test prędkości jazdy po skosie

Nr testu	Czas [s]	Wynik [m/s]
1	12,2	0,16
2	12,6	
3	13,4	
4	11,9	
5	12,6	

Jazda po skosie jest wolniejsza, niż jazda prosto a wynika to głównie z tego, że w przypadku tej konfiguracji ruchu robota napędzają jedynie dwa silniki.

Tab. 11.7. Test prędkości jazdy bokiem

Nr testu	Czas [s]	Wynik [m/s]
1	11	0,18
2	10,6	
3	11,4	
4	11,8	
5	10,8	

Jazda bokiem jest problematyczna dla robota głównie przez brak enkoderów, które kontrolowałyby prędkości wszystkich silników. Przy tym typie ruchu występuje poślizg przez co robot zbacza z kursu co negatywnie wpływa na czas przebycia wyznaczonej trasy.

- Prędkość obrotu

Tab. 11.8. Test prędkości obrotu

Nr testu	Czas [s]	Wynik [°/s]
1	10,8	68,05
2	10,6	
3	10,3	
4	10,5	
5	10,7	

Prędkość obrotu w trybie jazdy jest około 340% szybsza niż w trybie krocącym.

### 11.3. Wnioski

Robot jest w stanie poruszać się w obydwu trybach, z tym że z powodu użycia tanich serwonapędów zdolność poruszania się w trybie kroczenia po nierównych powierzchniach takich jak np. dywan jest ograniczona. Robot czasem się blokuje i trzeba go ręcznie przeprawić, aby mógł kontynuować ruch. Zastosowanie droższych napędów zniwelowałoby również drgania występujące podczas ruchu nóg.

## 12. Koszt konstrukcji robota

Wykonano zestawienia kosztów poszczególnych części robota w celu określenia, czy cena jego konstrukcji mieści się w założeniach opisanych w rozdziale 3.

### 12.1. Zestawienie kosztów poszczególnych części

W tabeli 12.1 przedstawiono zestawienie cen poszczególnych części składowych robota. Pominięto koszt żywicy i filamentu wykorzystanych do druku 3D oraz ceny śrub, nakrętek i okablowania.

Tab. 12.1. Zestawienie kosztów części robota

Lp.	Produkt	Cena jednostkowa [zł]	Ilość	Cena całkowita [zł]
1	Arduino UNO	26,99	1	26,99
2	Moduł HC-05	20,00	1	20,00
3	PCA9685	23,50	2	47,00
4	L298N	11,71	2	23,42
5	SZKB07	45,00	1	45,00
6	LM2596	6,00	2	12,00
7	Bateria Li-PO 3S 900mAh	40,00	1	40,00
8	Serwonapęd MG996R	20,00	18	360,00
9	Silnik DC MT80 70	20,00	4	80,00
Łączny koszt				654,41

Łączny koszt wszystkich części do robota wyniósł 654,41 złotych, czyli zmieszczono się w założonym budżecie.

## 12.2. Wnioski

Zastosowanie odpowiednich silników z enkoderami, czy też precyzyjniejszych serwonapędów do nóg prowadziłyby do przekroczenia budżetu określonego w rozdziale 3. Skutkowało to koniecznością zastosowania pewnych kompromisów takich jak rezygnacja z enkoderów czy też zastosowanie tańszych i mniej precyzyjnych serwonapędów MG996R. Pomimo tego, że w zestawieniu pominięto koszty niektórych części (śruby, nakrętki, kable itp.) cena całkowita robota znajduje się blisko limitu postawionego w założeniach.



## 13. Podsumowanie

Wykonanie autorskiego projektu oraz fizyczna implementacja hybrydowego robota, łączącego ze sobą zalety maszyny kroczącej oraz kół Mecanum, okazał się być sporym wyzwaniem. Zachowanie odpowiedniego balansu pomiędzy masą robota, a parametrami silników okazał się szczególnie kłopotliwy. Zbyt duża masa robota zmusza do zastosowania większych silników prądu stałego co z kolei powoduje, że konieczne jest użycie serwonapędów o większym momencie obrotowym. Wszystko to wiąże się z wyższymi cenami części i aby zmieścić się w wyznaczonym budżecie należało zrezygnować z pewnych elementów. Szczególnie widoczny jest brak enkoderów podczas ruchu w trybie jazdy, który objawia się sporym poślizgiem robota. Zjawisko to jest widoczne zwłaszcza podczas jazdy bokiem.

Pomimo trudności osiągnięto zamierzony cel pracy, czyli stworzono hybrydowego robota zdolnego do poruszania się na dwa różne sposoby, sterowanego za pomocą aplikacji na telefon poprzez Bluetooth. Wykonano szczegółowy model trójwymiarowy, za pomocą którego wydrukowane zostały części składowe robota. Dobrano odpowiednie elementy na podstawie obliczeń matematycznych, sporządzono schemat elektryczny oraz przygotowano odpowiednie rysunki techniczne oraz rysunek złożeniowy. Z powodzeniem użyto modułu BT, dzięki któremu robot jest w stanie komunikować się z autorską aplikacją na telefon. Stworzono również program na mikrokontroler Arduino, który zarządza pracą wszystkich podzespołów, dzięki czemu możliwa jest ich efektywna integracja.

W celu dalszego rozwoju projektu przede wszystkim wyposażylbym robota w silniejsze silniki prądu stałego wraz z enkoderami. Dzięki temu możliwe byłoby osiągnięcie płynnej jazdy we wszystkich kierunkach przy minimalnym poślizgu. Kolejnym ulepszeniem byłyby serwonapędy, których wymiana na lepsze odpowiedniki zniwelowałaby niepożądane drgania. Dałoby to możliwość realizacji ruchu kroczącego z wykorzystaniem kinematyki odwrotnej co sprawiłoby, że ruch robota byłby płynniejszy i wyglądałby bardziej naturalnie. Możliwe byłoby również wykorzystanie czujników styku na końcówkach nóg robota, dzięki czemu dałoby się balansować jego pozycję. Skutkowałoby to znacznie większymi możliwościami do pokonywania trudności terenowych. W zależności od kierunku rozwoju można byłoby również zaopatrzyć robotyka w odpowiednie sensory takie jak np. kamera wizyjna a nawet manipulator, dający możliwość interakcji robota z otoczeniem. Kolejnym ulepszeniem byłoby zastosowanie żywic o właściwościach ABS, dzięki czemu sama konstrukcja robota byłaby dużo wytrzymalsza i bardziej odporna na odkształcenia, co zmniejszyłoby ryzyko usterki.



## Bibliografia

- [1] Cook D., Budowa robotów dla średniozaawansowanych, Helion, Gliwice, 2013
- [2] Monk S., Arduino dla początkujących: podstawy i szkice, Helion, Gliwice, 2019
- [3] Monk S., Anna Mizerska Tł., Konrad Matuk Tł., Arduino dla początkujących: kolejny krok, Helion, Gliwice, 2021
- [4] Redwood B., Schöffner F., Garret B., The 3D Printing Handbook: Technologies, design and applications, 3D Hubs, 2017
- [5] Kloski L. W., Kloski N., Getting Started with 3D Printing: A Hands-on Guide to the Hardware, Software, and Services Behind the New Manufacturing Revolution, Make Community, 2016
- [6] Tzefastas S. G., Introduction to mobile robot controls, Elsevier, 2013
- [7] Veer A., Shital S C., Adaptive robust control of Mecanum-wheeled mobile robot with uncertainties, Nonlinear dynamics, 2017, p.2147-2169
- [8] Sun C., Yuan M., Li F., Yang Z., Ding X., Design and Simulation Analysis of Hexapod Bionic Spider Robot, Journal of Physics: Conference Series, Volume 1168 (2019)
- [9] Orozco-Magdaleno E. C., Gomez-Bravo F., Castillo-Castaneda E., Carbone G., Evaluation of Locomotion Performances for a Mecanum-Wheeled Hybrid Hexapod Robot, IEEE/ASME transactions on mechatronics,, 2021, p.1657-1667
- [10] Top A., Gökbulut M., Android Application Design with MIT App Inventor for Bluetooth Based Mobile Robot Control, Wireless personal communications, 2022, p.1403-1429
- [11] Veiga J. J. D., Pokress S. C., MIT App Inventor: Enabling Personal Mobile Computing, arXiv preprint, [arXiv:1310.2830v2](https://arxiv.org/abs/1310.2830v2), 2.02.2024
- [12] <https://learn.adafruit.com/16-channel-pwm-servo-driver?view=all>, 14.02.2024
- [13] <https://learn.adafruit.com/16-channel-pwm-servo-driver/using-the-adafruit-library>, 14.02.2024
- [14] <https://bostondynamics.com/products/spot/>, 14.02.2024
- [15] <https://community.robotshop.com/robots/show/spooder-octopod-robot>, 14.02.2024
- [16] Stańczyk B., Grzelczyk D., Awrejcewicz J., Control of mobile walking robot, Pomiar Automatyka Robotyka, 2012, p.157-159
- [17] <https://docs.revrobotics.com/duo-build/ftc-starter-kit-Mecanum-drivetrain/Mecanum-wheel-setup-and-behavior>, 14.02.2024

- [18] <https://automaticaddison.com/force-vector-diagrams-for-an-omni-directional-mobile-robot/>, 14.02.2024
- [19] <https://www.roboteq.com/applications/all-blogs/5-driving-Mecanum-wheels-omnidirectional-robots>, 14.02.2024
- [20] [https://botland.com.pl/Arduino-seria-podstawowa-oryginalne-plytki/1060-Arduino-uno-rev3-a000066-7630049200050.html?cd=18298825651&ad=&kd=&gad\\_source=1&gclid=Cj0KCQiA5rGuBhCnARIsAN11vgQVa4CKUzqHQurKx2sU6oZDL7\\_DL178AyJvbUb2sFfZxuX6CcfUt2QaAgdVEALw\\_wcB](https://botland.com.pl/Arduino-seria-podstawowa-oryginalne-plytki/1060-Arduino-uno-rev3-a000066-7630049200050.html?cd=18298825651&ad=&kd=&gad_source=1&gclid=Cj0KCQiA5rGuBhCnARIsAN11vgQVa4CKUzqHQurKx2sU6oZDL7_DL178AyJvbUb2sFfZxuX6CcfUt2QaAgdVEALw_wcB), 14.02.2024
- [21] <https://elektroweb.pl/silnikiserwa/59-serwo-towerpro-mg-996r-180-55g-94-kgcm-mg996-5904162800384.html>, 14.02.2024
- [22] <https://sklep.msalamon.pl/produkt/pca9685-16-sto-kanalowy-sterownik-pwm-servo/>, 14.02.2024
- [23] <https://allegrolokalnie.pl/oferta/silnik-z-przekladnia-6v-mt80-70obrmin>, 14.02.2024
- [24] [https://botland.com.pl/sterowniki-silnikow-moduly/3164-l298n-dwukanalowy-sterownik-silnikow-modul-12v-2a-5904422359317.html?cd=18298825138&ad=&kd=&gad\\_source=1&gclid=Cj0KCQiA5rGuBhCnARIsAN11vgSy\\_PdxOESIKsY10zVpVnchhkzWoIwtNHPmsZQgAMW9hScam0Qb\\_a8aAm90EALw\\_wcB](https://botland.com.pl/sterowniki-silnikow-moduly/3164-l298n-dwukanalowy-sterownik-silnikow-modul-12v-2a-5904422359317.html?cd=18298825138&ad=&kd=&gad_source=1&gclid=Cj0KCQiA5rGuBhCnARIsAN11vgSy_PdxOESIKsY10zVpVnchhkzWoIwtNHPmsZQgAMW9hScam0Qb_a8aAm90EALw_wcB), 14.02.2024
- [25] <https://allegro.pl/oferta/bateria-li-po-11-1v-3s-900mah-18a-20c-redox-11480040396>, 14.02.2024
- [26] <https://allegro.pl/oferta/przetwornica-step-down-300w-20a-regulacja-u-i-12549786997>, 14.02.2024
- [27] [https://botland.com.pl/przetwornice-step-down/2967-przetwornica-step-down-lm2596-32v-35v-3a-5903351241397.html?cd=18298825138&ad=&kd=&gad\\_source=1&gclid=Cj0KCQiA5rGuBhCnARIsAN11vgTlo\\_t-iHZLK8zn9ab9IJ1yUgisLBAUhm0JGD298pnaXLaF4nDm3uQaAnsnEALw\\_wcB](https://botland.com.pl/przetwornice-step-down/2967-przetwornica-step-down-lm2596-32v-35v-3a-5903351241397.html?cd=18298825138&ad=&kd=&gad_source=1&gclid=Cj0KCQiA5rGuBhCnARIsAN11vgTlo_t-iHZLK8zn9ab9IJ1yUgisLBAUhm0JGD298pnaXLaF4nDm3uQaAnsnEALw_wcB), 14.02.2024
- [28] [https://sklep.msalamon.pl/produkt/modul-Bluetooth-hc-05/?utm\\_source=Google%20Shopping&utm\\_campaign=msalamon&utm\\_medium=cpc&utm\\_term=355&gad\\_source=1&gclid=Cj0KCQiA5rGuBhCnARIsAN11vgTv5TBAQG-OKEbZDDhxiyTp6MBaPS82dAcQeCvPzGrwEonnDaVo2MaAk-SEALw\\_wcB](https://sklep.msalamon.pl/produkt/modul-Bluetooth-hc-05/?utm_source=Google%20Shopping&utm_campaign=msalamon&utm_medium=cpc&utm_term=355&gad_source=1&gclid=Cj0KCQiA5rGuBhCnARIsAN11vgTv5TBAQG-OKEbZDDhxiyTp6MBaPS82dAcQeCvPzGrwEonnDaVo2MaAk-SEALw_wcB), 14.02.2024

- [29] <https://mae.ufl.edu/designlab/Class%20Projects/Background%20Information/Friction%20coefficients.htm>, 15.02.2024
- [30] <https://spectrum.ieee.org/ghost-robotics-minitaur-quadruped>, 21.02.2024
- [31] <https://kodlab.seas.upenn.edu/robots/rhex/>, 21.02.2024
- [32] <http://robotics.hobbizine.com/knexabeast.html>, 21.02.2024
- [33] <https://www.supoheer.com/large-agv-tractor/mecanum-wheel-agv.html>, 21.02.2024
- [34] <https://www.youtube.com/watch?v=MHXjseAhajk>, 21.02.2024

## Załączniki pracy

Folder „Rysunki” zawiera:

1. Rysunek techniczny „Adapter.pdf”
2. Rysunek techniczny „Łączenie nogi.pdf”
3. Rysunek techniczny „Piszczel.pdf”
4. Rysunek techniczny „Platforma dolna 1”
5. Rysunek techniczny „Platforma dolna 2”
6. Rysunek techniczny „Platforma górna 1”
7. Rysunek techniczny „Platforma górna 2”
8. Rysunek techniczny „Podstawa nogi lewa.pdf”
9. Rysunek techniczny „Podstawa nogi prawa.pdf”
10. Schemat połączeń elektrycznych „Schemat elektryczny.pdf”
11. Rysunek techniczny „Wzmocnienie puszczela lewe.pdf”
12. Rysunek techniczny „Wzmocnienie puszczela prawe.pdf”
13. Rysunek złożeniowy „Rysunek złożeniowy.pdf”

Folder „Programy” zawiera:

1. Aplikacja na telefon „Mecanum\_Hexapod.apk”
2. Folder „Hexapod\_Mecanum\_robot” na Arduino zawierający:
  - Kod do Arduino „Hexapod-Mecanum robot.ino”