Projekt

Sterowniki robotów

Dokumentacja

Manipulator 3DoF
M3DoF

Skład grupy:
Kacper Drążyk, 263462
Hanna Grześkowiak, 263492

Termin: srTN17

Prowadzący:
dr inż. Wojciech Domski

27 czerwca 2024

Spis treści

1 Opis projektu **2**

2 Założenia projektowe **2**

3 Konfiguracja mikrokontrolera **3**
3.1 Konfiguracja pinów 5
3.2 UART 5
3.3 USB 5
3.4 TIMER 5
3.5 PWM 6

4 Urządzenia zewnętrzne **6**
4.1 Sterownik silnika krokowego - DRV8825 6
4.2 Silniki krokowe 17HS4401 6
4.3 Kontroler 8BitDo Pro 2 7

5 Elektronika **7**
5.1 Zasilanie 7
5.2 Napęd osi 7
5.3 Sterowanie chwytaka 7
5.4 Wyłączniki krańcowe 7

6 Konstrukcja mechaniczna **8**
6.1 Przekładnia cykloidalna 8

7 Opis działania programu **9**
7.1 Przestrzeń robocza 9
7.2 Notacja Denavita-Hartenberga i Jakobian 10
7.3 Algorytm Newtona-Raphsona 10
7.4 Wyliczenie geometryczne 11
7.5 Krzywa prędkości silników 12
7.6 Sterowanie padem 13

8 Zadania niezrealizowane **13**
8.1 Referencja osi 13
8.2 Sterowniki silników krokowych 13

9 Kod źródłowy **13**

10 Podsumowanie **14**

Bibilografia **15**

1 Opis projektu

Celem projektu było stworzenie i sterowanie manipulatorem [3] o trzech stopniach swobody wraz z liczeniem zadania kinematyki odwrotnej na mikrokontrolerze. Do wyliczenia kinematyki odwrotnej wykorzystane zostały dwa algorytmy: za pomocą jakobianu z wykorzystaniem metody Newtona-Raphsona [2] i z geometrycznie wyliczonych kątów. Sterowanie odbywa się z komputera z możliwością zadawania współrzędnych końcowych efektora oraz otwierania i zamykania chwytaka lub przy pomocy kontrolera.

Rysunek 1: Manipulator

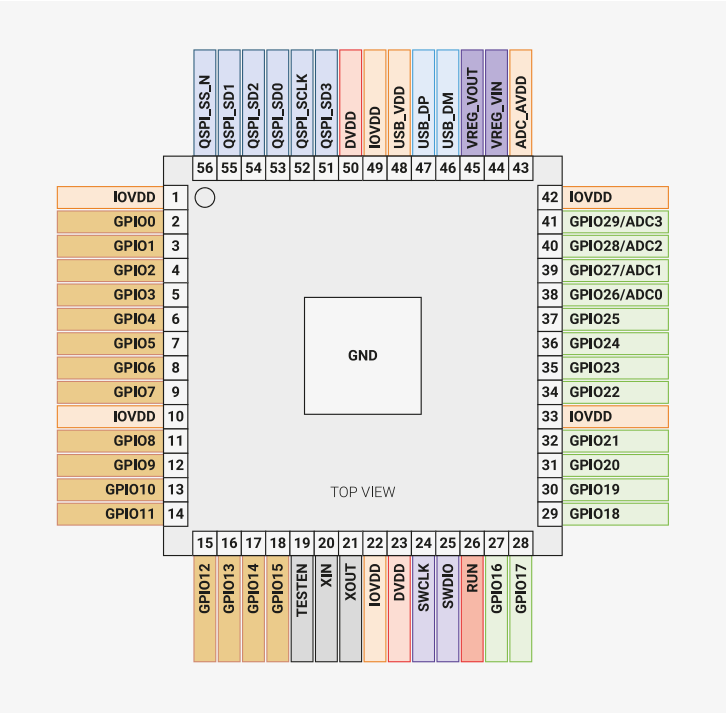
2 Założenia projektowe

Finalny projekt powinien spełniać następujące założenia:

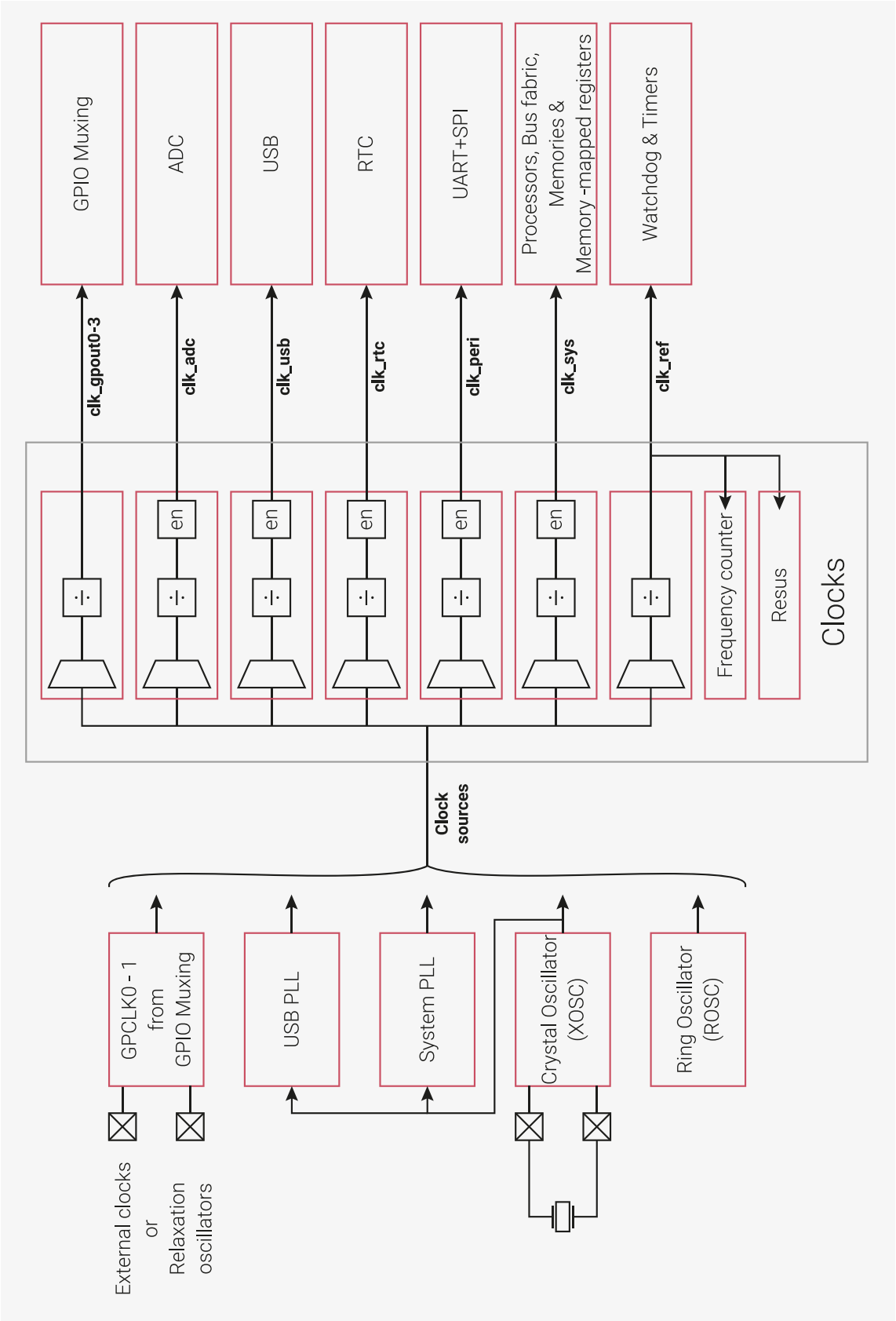
- manipulator o trzech stopniach swobody z chwytakiem,
- obliczanie kinematyki odwrotnej na mikrokontrolerze za pomocą jakobianu [4] z wykorzystaniem metody Newtona-Rhapsona [1] i z gemetrycznie wyliczonych kątów,
- interfejs linii poleceń umożliwiający kontrolę manipulatora,
- sterowanie we współrzędnych efektora oraz przegubów.

3 Konfiguracja mikrokontrolera

W projekcie został wykorzystany moduł deweloperski Raspberry Pi Pico W oparty o mikrokontroler RP2040 [8]. Sterownik komunikuje się z komputerem za pośrednictwem interfejsu USB oraz ze sterownikami silników krokowych za pośrednictwem interfejsu UART. Dzięki modułowi Infineon CYW43439 możliwa jest też komunikacja bezprzewodowa z kontrolerem 8BitDo Pro 2.



Rysunek 2: Konfiguracja wyjść mikrokontrolera [5].



Rysunek 3: Schemat blokowy konfiguracji zegarów mikrokontrolera [5]

3.1 Konfiguracja pinów

Numer pinu	PIN	Tryb pracy	Funkcja/etykieta
2	GPIO0	SIO	J0 DIR
3	GPIO1	PWM	J0 STEP
4	GPIO2	SIO	SLEEP
5	GPIO3	SIO	J0 RST
9	GPIO7	SIO	DIR2
11	GPIO8	SIO	J1 DIR
12	GPIO9	PWM	J1 STEP
13	GPIO10	SIO	J1 RST
14	GPIO11	SIO	J2 DIR
15	GPIO12	PWM	J2 STEP
16	GPIO13	SIO	MICROSTEPPING
17	GPIO14	SIO	J2 RST
20	XIN		Oscylator
21	XOUT		Oscylator
29	GPIO18	SIO	J2 Endstop
30	GPIO19	SIO	J1 Endstop
31	GPIO20	SIO	J0 Endstop
32	GPIO21	PWM	GRIPPER
35	GPIO23	SIO	RT6150 PS
36	GPIO24		VBUS sense
37	GPIO25	SIO	LED
41	GPIO29		VSYS/3
46	USB_DM	-	USB diff
47	USB_DP	-	USB diff
51	QSPI_SD3	-	QSPI FLASH
52	QSPI_SCLK	-	QSPI FLASH
53	QSPI_SD0	-	QSPI FLASH
54	QSPI_SD2	-	QSPI FLASH
55	QSPI_SD1	-	QSPI FLASH
56	QSPI_SS_N	-	QSPI FLASH

Tabela 1: Konfiguracja pinów mikrokontrolera

3.2 UART

Interfejs UART jest wykorzystywany do komunikacji z komputerem za pośrednictwem portu USB. Konfiguracja interfejsu została opisana w tabeli 2.

Parametr	Wartość
Baud Rate	115200
Word Length	8 Bits
Parity	None
Stop Bits	1

Tabela 2: Konfiguracja peryferów UART

3.3 USB

Interfejs USB został wykorzystany do sterowania ramieniem z komputera. Komunikacja przebiega w trybie USB CDC (imitacja portu szeregowego).

3.4 TIMER

Zegar jest wykorzystywany do sterowania chwytakiem z kontrolera. Co dokładnie 40ms następuje korekcja pozycji zamknięcia chwytaka zgodnie z wartością zadaną. Konfiguracja znajduje się w tebeli 3.

Parametr	Wartość
Period	40ms
Mode	exact

Tabela 3: Konfiguracja timera.

3.5 PWM

Peryferium PWM odpowiada za sygnał sterujący silników. W przypadku silników krokowych są to impulsy, a serwo jest sterowane przez wypełnienie sygnału. Część parametrów PWM zmienia się w czasie. Zostało to opisane w tabelach 4 i 5. Kanały zależą od wybranych wyjść. W tym przypadku są to 0B, 4B, 6A i 2B [5].

Parametr	Wartość
clkdiv	$\sim$
wrap	$\sim$
level	$wrap/2$

Tabela 4: Konfiguracja PWM dla silników krokowych. Wypełnienie zawsze wynosi 50%, a clkdiv i wrap są zmienne.

Parametr	Wartość
clkdiv	38.1451
wrap	65532
level	$\sim$

Tabela 5: Konfiguracja PWM dla silników serwa. Częstotliwość wynosi 50Hz, a wypełnienie jest zmienne.

4 Urządzenia zewnętrzne

Zastosowanie gotowych, sprawdzonych rozwiązań pozwoliło na ograniczenie nakładu pracy i podniesienie jakości projektu. Zintegrowane sterowniki silników krokowych mogłyby zostać zastąpione przez proste układy tranzystorowe, ale takie rozwiązanie wymagałoby zastosowania większej ilości elementów, a jakość sterowania uległaby najprawdopodobniej pogorszeniu.

4.1 Sterownik silnika krokowego - DRV8825

Sterowniki działają we współpracy z silnikami krokowymi Nema 17HS4401. Do komunikacji z mikrokontrolerem wykorzystywane są linie STEP, DIR i EN. Każdy sterownik wymaga ustawienia limitu prądu uzwojenia silnika zgodnie z dokumentacją. Służy do tego potencjometr umieszczony na module. Dla zastosowanych silników limit wynosi 1.7 A, co oznacza napięcie $V_{ref} = 0.85\,\mathrm{V}$.

Konfiguracja sterowników została zaprezentowana w tabeli 6.

Parametr	Wartość
J0 microstepping	32
J1 microstepping	32
J2 microstepping	32

Tabela 6: Konfiguracja microsteppingu sterowników.

4.2 Silniki krokowe 17HS4401

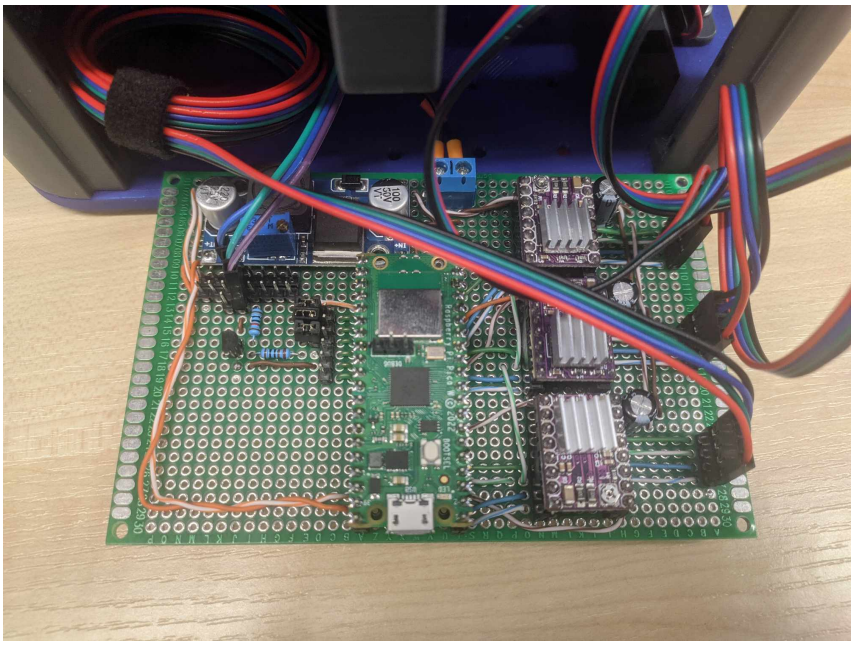
Zastosowane silniki krokowe są kompatybilne ze standardem Nema 17, a ich najczęstszym przeznaczeniem są drukarki 3D. Ich zaletami są niewielkie rozmiary, wysoka rozdzielczość (nawet 6400 pozycji na obrót) oraz możliwość zastosowania otwartej pętli sterowania. Z uwagi na stosunkowo niewielki moment obrotowy konieczne było zastosowanie dodatkowych przekładni.

4.3 Kontroler 8BitDo Pro 2

Kontroler firmy 8BitDo został stworzony do współpracy z kompuerami, urządzeniami mobilnymi i konsolami do gier. Obsługuje wiele standardów, m.in. X-input stosowany w kontrolerach do konsol Xbox firmy Microsoft. Dzięki bibliotece bluepad32 możliwe jest połączenie go z modułem Raspberry Pi Pico W oraz konfiguracja funkcji przypisanych do przycisków.

5 Elektronika

Do stworzenia sterownika robota została wykorzystana płytka prototypowa (rys. 4). Cały układ elektroniczny można podzielić na kilka modułów, które zostały oznaczone na schemacie (załącznik 1).



Rysunek 4: Zdjęcie układu elektronicznego.

5.1 Zasilanie

Do zasilania robota został wykorzystany akumulator litowo-polimerowy 3S. Napięcie z akumulatora jest bezpośrednio przekazywane na sterowniki silników krokowych, a dodatkowa szyna 5V dla pozostałych układów jest zasilana przez moduł przetwornicy DC-DC step-down oparty o układ LM2596.

5.2 Napęd osi

Osie robota są napędzane przez silniki krokowe NEMA 17 17HS4401. Z każdym z nich współpracuje moduł stepstick DRV8825. Na płytce została uwzględniona konfiguracja microsteppingu przez zworki.

5.3 Sterowanie chwytaka

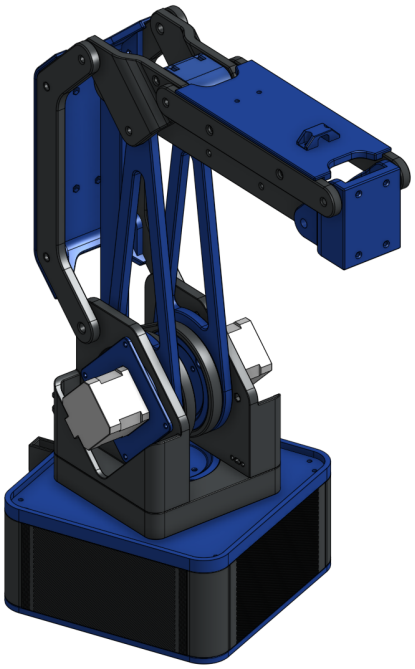
Serwo modelarskie MG996R, które porusza chwytakiem wymaga zasilania oraz sygnału logicznego na poziomie 5V. Napięcie sygnałów logicznych mikrokontrolera RP2040 to 3.3V, więc został dodany układ "low side switch"z rezystorem podciągającym w oparciu o tranzystor BC547.

5.4 Wyłączniki krańcowe

Na płytce zostały wyprowdzone wejścia pod wyłączniki krańcowe do przeprowadzania referencji położenia osi robota, ale nie zostały wykorzystane.

6 Konstrukcja mechaniczna

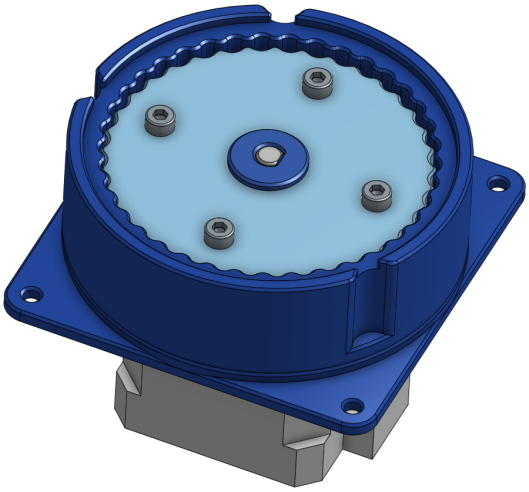
Manipulator ma trzy przeguby obrotowe. Jego konstrukcja różni się od typowego ramienia 3R tym, że silnik trzeciej osi również znajduje się w podstawie. Z każdym silnikiem współpracuje przekładnia cykloidalna o przełożeniu 1:37. Całość została wykonana w technologii druku 3D z dodatkiem standardowych elementów (łożyska ślizgowe oraz pręty hartowane), aby poprawić sztywność konstrukcji.



Rysunek 5: Model manipulatora stworzony w oprogramowaniu CAD.

6.1 Przekładnia cykloidalna

Wykorzystane silniki krokowe mają zbyt mały moment obrotowy, aby bezpośrednio napędzać osie, więc zostały zastosowane dodatkowe przekładnie. Przekładnie cykloidalne są często stosowane w robotyce z uwagi na bardzo dobry stosunek przełożenia do rozmiarów, ale komercyjne rozwiązania wykraczają poza budżet projektu. Dzięki technologii druku 3D możliwe było stworzenie przekładni idealnie dopasowanych pod to zastosowanie. Ich parametry mechaniczne są gorsze, ale zasada działania nie różni się od wersji profesjonalnych.



Rysunek 6: Model przekładni cykloidalnej stworzony w oprogramowaniu CAD. Wierzchnia pokrywa została ukryta, aby pokazać budowę wewnętrzną.

7 Opis działania programu

Interfejs programu daje użytkownikowi menu dostępnych akcji do wykonania manipulatorem oraz wyświetla podstawowe informacje o nim, takie jak aktualne współrzędne wewnętrzne i zewnętrzne oraz stan chwytaka. Za pomocą menu przy każdym powtórzeniu od użytkownika pobierana jest wybrana przez niego opcja akcji do wykonania. Do wyboru są:

1. Ruch relatywny we współrzędnych wewnętrznych - użytkownik podaje, o ile obrócić każdy z przegubów J0, J1 i J2,
2. Ruch relatywny we współrzędnych zewnętrznych - użytkownik wybiera algorytm KO i współrzędne docelowe X, Y i Z,
3. Ruch absolutny we współrzędnych zewnętrznych - użytkownik wybiera algorytm KO i współrzędne docelowe X, Y i Z,
4. Zamknięcie chwytaka
5. Otwarcie chwytaka
6. Powrót do pozycji wyjściowej - powoduje wyzerowanie kątów w programie i powrót manipulatora do pozycji wyjściowej jak na rys. 7
7. Wyzerowanie układu współrzędnych - powoduje wyzerowanie kątów w programie, do wykorzystania przy awaryjnym zatrzymaniu manipulatora i ręcznym ustawieniu do pozycji wyjściowej
8. Przejście na sterowanie padem

7.1 Przestrzeń robocza

Dla ruchu we współrzędnych zewnętrznych sprawdzane jest, czy zadane wartości mieszczą się w przestrzeni roboczej, czyli czy są w zasięgu maksymalnie wyciągniętego ramienia oraz czy kąty w przegubach wymagane do osiągnięcia zadanej pozycji nie wychodzą poza zakres minimalnych i maksymalnych obrotów w przegubach, które wynoszą:

$J0 : \{-135°, 135°\}$

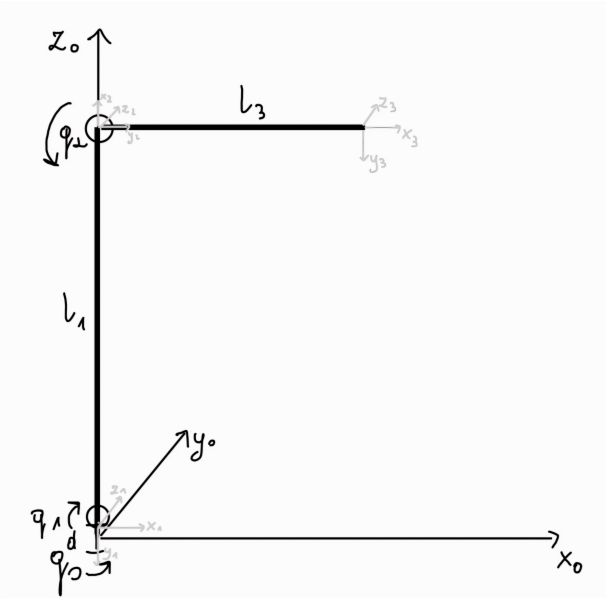
$J1 : \{-15°, 75°\}$

$J2 : \{-55°, 25°\}$

Jeżeli zadana pozycja docelowa nie znajduje się w przestrzeni roboczej, użytkownik dostaje informację o przekroczonych zakresach, a ruch nie jest wykonywany. Przy poprawnych współrzędnych docelowych algorytmy wyliczają kąty potrzebne do osiągnięcia pozycji docelowej, a następnie zadają ruch o różnicę docelowych i aktualnych współrzędnych wewnętrznych.

7.2 Notacja Denavita-Hartenberga i Jakobian

Dla kątów w przegubach do wyliczenia Jakobianu wykorzystana została notacja Denevita-Hartenberga (rys. 7), wyliczone wartości przekształceń dla kolejnych układów współrzędnych znajdują się w tabeli 7.



Rysunek 7: Notacja Denevita-Hartenberga.

	$R(z,\alpha)$	$T(z,a)$	$T(x,b)$	$R(x,\beta)$
0->1	q_0	d	0	$-\frac{\pi}{2}$
1->2	$q_1 - \frac{\pi}{2}$	0	l_1	0
2->3	$-q_2 + \frac{\pi}{2}$	0	l_3	0

Tabela 7: Tabela D-H dla manipulatora.

$$\begin{vmatrix} -s_1(l_1 s_2 + l_3 c_{23}) & c_1(l_1 c_2 - l_3 s_{23}) & l_3 s_{23} c_1 \\ c_1(l_1 s_2 + l_3 c_{23}) & s_1(l_1 c_2 - l_3 s_{23}) & l_3 s_1 s_{23} \\ 0 & -l_1 s_2 - l_3 s_{23} & l_3 s_{23} \end{vmatrix}$$

Tabela 8: Jakobian manipulatora.

gdzie $s_{23} = sin(q_2 - q_3)$.

7.3 Algorytm Newtona-Raphsona

Algorytm Newtona-Raphsona zaimplementowany zostal zgodnie w przykładem zaprezentowanym w [6]. Jego dzialanie opiera sie na rownaniu (1) wykonywanym powtarzalnie do momentu, kiedy odleglość między aktualnym i zadanym położeniem efektora jest mniejsza niż wartość przyjęta jako maksymalny dopuszczalny bląd.

$$q_n = q - \gamma J^{-1}(P_a - P_d) \tag{1}$$

- q_n, q - wartości przegubów w następnym i poprzednim kroku (w pierwszym kroku aktualne wartości)

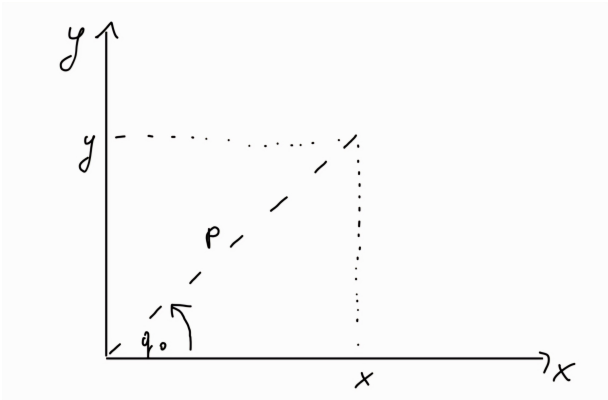
- γ - współczynnik zbieżności
- J - aktualna macierz Jakobiego (tab. 8)
- P_a - aktualne wspólrzędne efektora
- P_d - zadane współrzędne efektora

7.4 Wyliczenie geometryczne

Dla manipulatorów o większej ilości stopni swobody wyliczenie geometryczne kinematyki odwrotnej nie jest przez złożoność dostępną opcją, jednak dla prezentowanego manipulatora było możliwe przy wykorzystaniu podstawowych funkcji trygonometrycznych i twierdzenia cosinusów (2).

$$a^2 = b^2 + c^2 - 2abcos(\alpha) \tag{2}$$

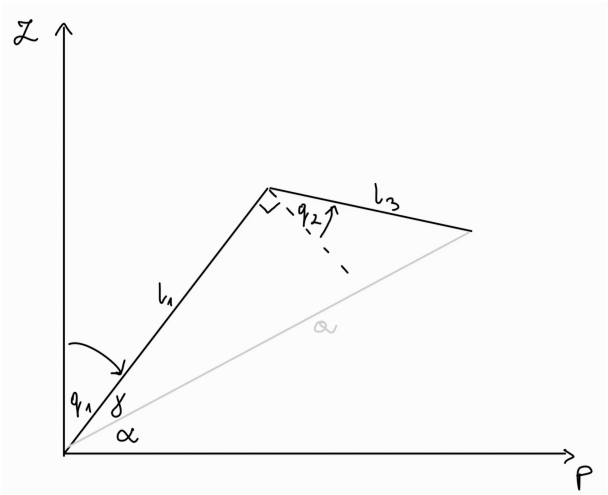
Wyliczenie kąta q_0 (3) wykonane zostało na rzucie przegubów l_1 i l_3 na płaszczyznę XY jako $p^2 = x^2 + y^2$.



Rysunek 8: Rzut manipulatora na płaszczyznę XY.

$$\begin{aligned} tan(q_0) &= \frac{x}{y} \\ q_0 &= arctan\left(\frac{x}{y}\right) \end{aligned} \tag{3}$$

Następnie na rzucie manipulatora na płaszczyznę PZ przy $a^2 = p^2 + z^2$ wyznaczone zostały kąty q_1 (6) jako dopełnienie sumy kątów α i γ oraz q_2 (7) z twierdzenia cosinusów w trójkącie $l_1 l_3 a$.



Rysunek 9: Rzut manipulatora na płaszczyznę PZ.

$$tan(\alpha) = \frac{z}{p}$$
$$\alpha = arctan(\frac{z}{p}) \tag{4}$$

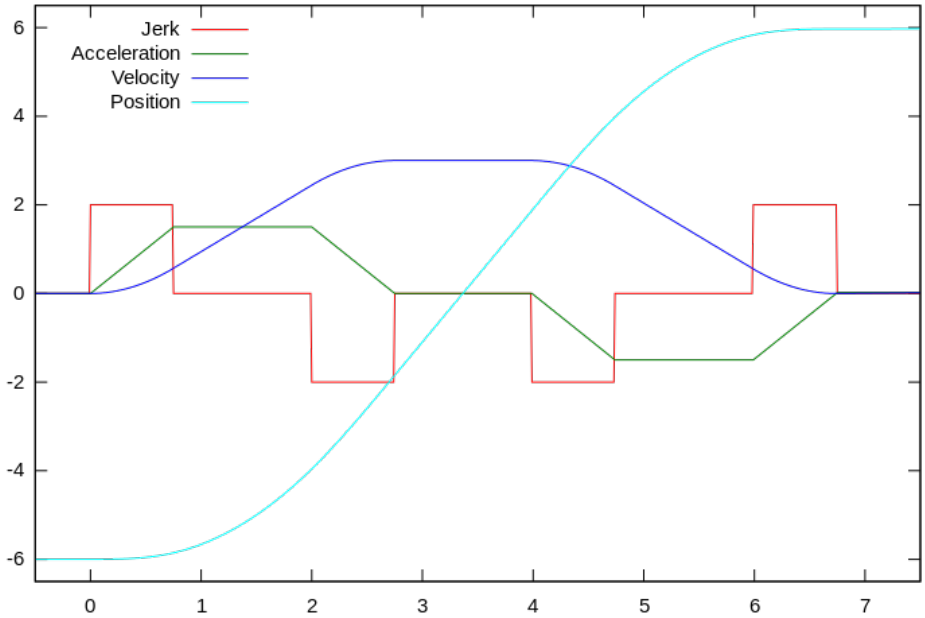
$$l_3^2 = l_1^2 + a^2 - l_1 acos(\gamma)$$
$$\gamma = arccos(\frac{l_1^2 + p^2 + z^2 - l_3^2}{2l_1\sqrt{p^2+z^2}} \tag{5}$$

$$q_1 = \frac{\pi}{2} - arctan(\frac{z}{p}) - arccos(\frac{l_1^2 + p^2 + z^2 - l_3^2}{2l_1\sqrt{p^2+z^2}} \tag{6}$$

$$a^2 = l_1^2 + l_3^2 - 2l_1 l_3 cos\left(\frac{\pi}{2} + q_2\right)$$
$$a^2 = l_1^2 + l_3^2 + 2l_1 l_3 sin(q_2)$$
$$sin(q_2) = \frac{p^2 + z^2 - l_1^2 - l_3^2}{2l_1 l_3} \tag{7}$$
$$q_2 = arcsin\left(\frac{p^2 + z^2 - l_1^2 - l_3^2}{2l_1 l_3}\right)$$

7.5 Krzywa prędkości silników

Aby zwiększyć płynność ruchu manipulatora i ograniczyć drgania należy zastosować odpowiedni profil prędkości. Dwa typowe rozwiązania to krzywa trapezoidalna lub sigmoidlana. W pierwszej wykres prędkości jest ciągły, a przyspieszenie zmienia się skokowo, natomiast w drugiej obie wartości są ciągłe, a nieciągłość występuje dopiero przy zrywie (pochodnej przyspieszenia). Przykład zmian tych wartości w trakcie ruchu obrazuje rysunek 10.



Rysunek 10: Wykres zmian położenia, prędkości, przyspieszenia i zrywu dla ruchu o ciągłym wykresie przyspieszenia. Źródło: Wikipedia.

Istnieją dwie metody implementacji tych krzywych: odczyt wartości z tabeli i obliczanie w locie. Z uwagi na prostotę implementacji w projekcie zostało zastosowane pierwsze rozwiązanie. Po zadaniu parametrów generowane jest 5000 próbek rozłożonych równomiernie na krzywej trapezoidalnej i zapisywane w pamięci RAM.

Rysunek 11: Kontroler wykorzystany do sterowania manipulatorem wraz z opisem funkcji przypisanych do przycisków.

7.6 Sterowanie padem

Do sterowania padem została wykorzystana biblioteka Bluepad32 [7]. Konfiguracja peryferiów nie jest widoczna w interfejsie biblioteki, więc nie została uwzględniona w dokumentacji. Sposób obsługi manipulatora został opisany na rysunku 11.

8 Zadania niezrealizowane

W trakcie pracy nad manipulatorem stało się jasne, że początkowe założenia projektowe będą musiały zostać zmodyfikowane.

8.1 Referencja osi

Plan projektu zakładał realizację referencji pozycji wszstkich osi. Z uwagi na dużą złożoność mechaniczną tych układów i niewielką trudność w implementacji po stronie oprogramowania to zadanie zostało pominięte.

8.2 Sterowniki silników krokowych

Początkowo projek zakładał wykorzystanie bardziej zaawansowanych sterowników Trinamic TMC2208 wraz z dodatkowymi interfejsami UART. W fazie tworzenia pierwszego prototypu zostały one zamienione na sterowniki DRV8825, które dobrze się sprawdziły w tym zadaniu i nie było potrzeby zmiany.

9 Kod źródłowy

Kod źródłowy oraz fragmenty dokumentacji są dostępne w publicznym repozytorium na platformie Gitlab: https://git.kcir.pwr.edu.pl/kdrazyk/sr-manipulator.

10 Podsumowanie

W ramach projektu udało się stworzyć manipulator o trzech stopniach swobody sterowanego we współrzędnych wewnętrznych oraz zewnętrznych z komputera oraz za pomocą pada.

Praca nad projektem pozwoliła na zdobycie wiedzy o manipulatorach i rozwinięcie doświadczenia teoretycznego oraz praktycznego. W tarkcie prac nad manipulatorem konieczne było szukanie coraz lepszych rozwiązań, w celu poprawy płynności ruchu oraz dokładności osiąganego celu. Specyfika konstrukcji wymagała szerszego spojrzenia oraz dopasowania standardowych algorytmów do ruchu dla kątów w przegubie J2 zależnych od J1. Udało się zaimplementować dodatkowe funkcje ułatwiające użytkowanie, takie jak sterowanie padem oraz powrót programowy i zdalny do pozycji wyjściowej w razie awaryjnego zatrzymania.

Projekt ma duży potencjał na rozwój. Modułowość całej konstrukcji zapewnia łatwość wymiany elementów w razie ich zużycia lub zepsucia, a także umożliwia montaż wymiennych efektorów w przyszłości w celu rozszerzenia funkcjonalności.

Literatura

[1] S. Akram, Q. U. Ann. Newton raphson method. *International Journal of Scientific & Engineering Research*, 6(7):1748–1752, 2015.

[2] Engineering Educator Academy. Numerical inverse kinematics using the newton raphson method. `https://youtu.be/_fVgZASOOCw?si=kxeQWgF6YWZEtJ2x`, Sier. 2021.

[3] K. Tchoń i inni. Manipulatory i roboty mobilne: modele, planowanie ruchu, sterowanie. Akad. Oficyna Wyd PLJ., 2000.

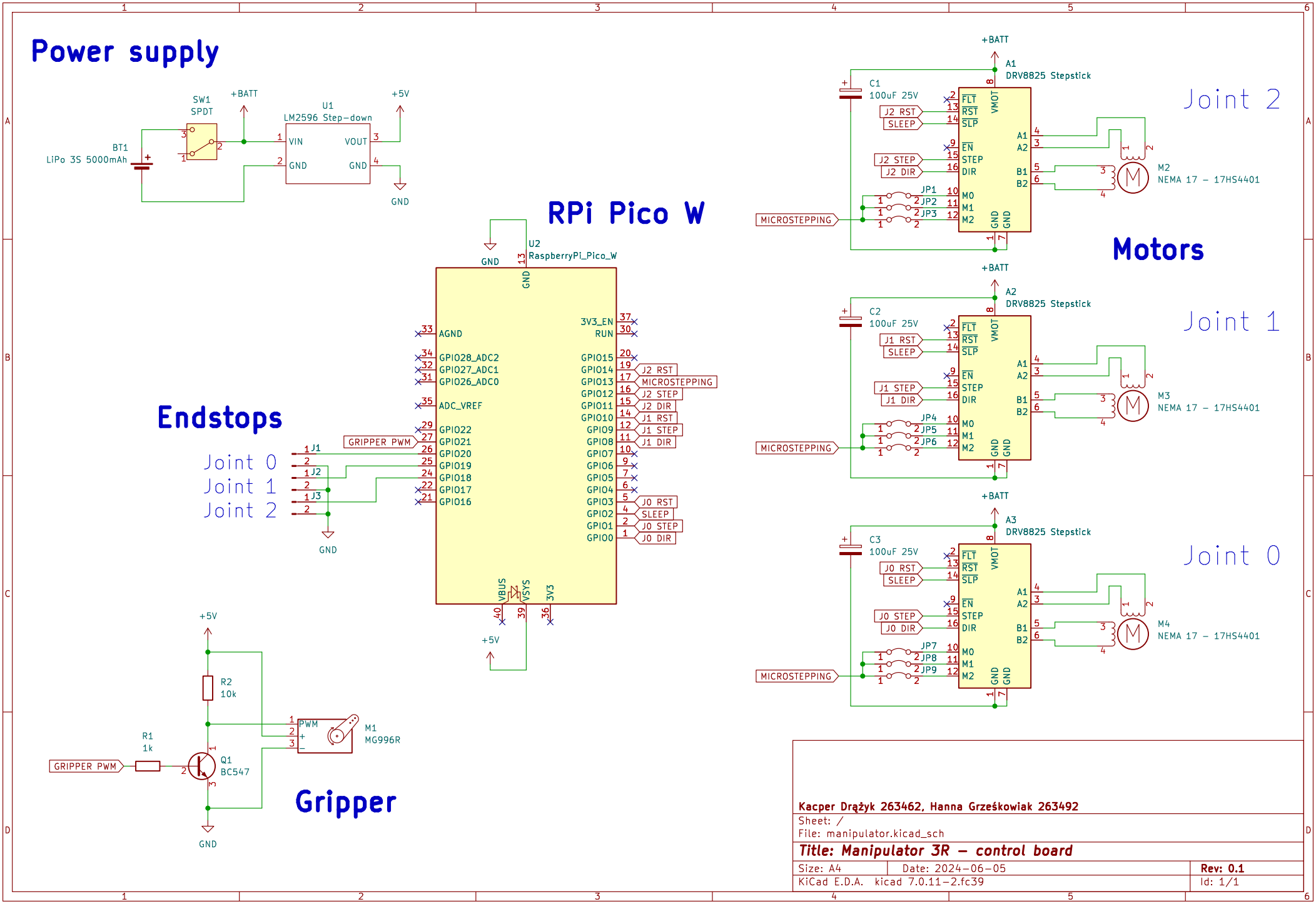
[4] M. Meredith, S. Maddock. Real-time inverse kinematics: The return of the jacobian. 2004.

[5] Paspberry Pi Ltd. Rp2040 datasheet. `https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf`, Luty 2024.

[6] Passion for Robotics. Podstawy Robotyki - Kinematyka Odwrotna (ALGORYTM Jacobianowy), Dec 2020.

[7] R. Quesada. Bluepad32. `https://github.com/ricardoquesada/bluepad32`. [Online, odwiedzone 01.03.2024].

[8] Raspberry Pi Ltd. Raspberry pi pico datasheet. `https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf`, Luty 2024.


Power supply
SW1
SPDT
+BATT
U1
LM2596 Step-down
+5V
BT1
LiPo 3S 5000mAh
VIN
VOUT
GND
RPi Pico W
U2
RaspberryPi_Pico_W
Endstops
Joint 0
Joint 1
Joint 2
GRIPPER PWM
Gripper
R2
10k
R1
1k
Q1
BC547
M1
MG996R
Motors
A1
DRV8825 Stepstick
C1
100uF 25V
M2
NEMA 17 – 17HS4401
A2
C2
M3
A3
C3
M4
MICROSTEPPING
J2 RST
SLEEP
J2 STEP
J2 DIR
J1 RST
J1 STEP
J1 DIR
J0 RST
J0 STEP
J0 DIR
Kacper Drążyk 263462, Hanna Grześkowiak 263492
Sheet: /
File: manipulator.kicad_sch
Title: Manipulator 3R – control board
Size: A4
Date: 2024-06-05
Rev: 0.1
KiCad E.D.A. kicad 7.0.11-2.fc39
Id: 1/1