

Uniwersytet Mikołaja Kopernika Wydział Fizyki, Astronomii
i Informatyki Stosowanej

Andrzej Wąsowski

nr albumu: 279117

Automatyka i robotyka

Praca inżynierska

Sterowanie maszyną numeryczną w
układzie Delta z wykorzystaniem
oprogramowania LinuxCNC

Opiekun pracy dyplomowej

dr inż. Krystian Erwiński

Katedra Automatyki i Systemów Pomiarowych

Toruń 2024

*UMK zastrzega sobie prawo własności niniejszej pracy inżynierskiej w celu udostępniania dla potrzeb
działalności naukowo-badawczej lub dydaktycznej*

*Dziękuję Panu dr inż. Krystianowi Erwińskiemu
oraz mgr inż. Gabrielowi Karaskowi za wszelką
pomoc przy realizacji pracy inżynierskiej.*

Spis treści

Abstract	5
1. Wstęp	6
1.1 . Cel i zakres pracy	7
2. Układy sterowania numerycznego.....	8
2.1. Wprowadzenie.....	8
2.2. Architektura sterowania numerycznego.....	9
2.3. Jądro sterowania numerycznego	10
2.4. Interfejs komunikacyjny.....	13
2.4.1. EtherCAT.....	13
2.4.2. CANopen	14
2.5. Serwonapęd	15
2.6. Rodzaje układów sterowania.....	16
3. Budowa stanowiska	18
3.1. Komponenty maszyny numerycznej typu Delta liniowa	20
3.2. Kinematyka maszyny typu Delta liniowa	26
4. Struktura oprogramowania sterującego	30
4.1. Oprogramowanie LinuxCNC	30
4.2. Konfiguracja oprogramowania sterującego	35
4.3. Konfiguracja oprogramowania serwonapędów.....	38
5. Badania	41
5.1. Strojenie regulatorów serwonapędów osi	41
5.2. Testy jakości obróbki	47
6. Podsumowanie.....	51
Bibliografia	52
Załącznik nr 1. Listingi plików konfiguracyjnych oprogramowania LinuxCNC.....	55
Załącznik nr 2. Schemat elektryczny układu sterowania numerycznego	66

Abstract

The objective of this thesis was to implement numerical control of a linear delta-type machine using a desktop computer with LinuxCNC software as a controller. The existing mechanical structure was equipped with the necessary actuators and sensors, which were wired inside a control cabinet. The servo drives consisted of stepper motors with attached encoders and Nanotec C5-E controllers, which were controlled in a closed-loop system. Communication between the machine components and the controller was carried out using EtherCAT bus with the CANopen protocol as an application layer. Additionally, a CiA402 device profile was used as an interface for the servo drives. The Hardware Abstraction Layer (HAL), a crucial component of the LinuxCNC software, served as an interface between the hardware and software components. The cascade PID controllers of the servo drives were tuned to achieve minimal following error. A study was conducted on the milling process, which involved tuning the parameters of linear delta-type kinematics.

1. Wstęp

Komputerowe sterowanie numeryczne CNC (ang. Computerized Numerical Control) stanowiło znaczący przełom w przemyśle wytwórczym. W przeciwieństwie do tradycyjnych maszyn sterowanych ręcznie, maszyny CNC są sterowane cyfrowymi instrukcjami najczęściej wygenerowanymi z wykorzystaniem oprogramowania do projektowania wspomaganego komputerowo CAD (ang. Computer Aided Design). Wprowadzenie technologii CNC zrewolucjonizowało przemysł produkcyjny poprzez zwiększenie precyzji, wydajności i powtarzalności procesu wytwórczego. Maszyny CNC takie jak frezarki, tokarki czy drukarki 3D wykonują elementy, często o złożonych formach, w sposób precyzyjny, powtarzalny i zautomatyzowany. Umożliwia to masowe wytwarzanie produktów wysokiej jakości przy minimalnej ingerencji człowieka.

Jednym z zasadniczych komponentów maszyn sterowanych numerycznie jest sterownik CNC, który coraz częściej zbudowany jest w oparciu o komputer klasy PC. Jest to możliwe dzięki implementacji algorytmów sterowania numerycznego w formie programu komputerowego działającego w systemie operacyjnym komputera sterującego. Często są to systemy operacyjne czasu rzeczywistego umożliwiające realizację algorytmów sterowania w sposób deterministyczny czasowo. Do popularności takiego podejścia przyczynił się rozwój wolnego i otwartego oprogramowania obejmującego zarówno otwarte jądra czasu rzeczywistego (Linux RTAI, Linux RT-Preempt) jak również samo oprogramowanie sterowania numerycznego np. LinuxCNC.

Istnieje wiele rodzajów konstrukcji maszyn CNC. Jednym z coraz częściej stosowanych rozwiązań jest kinematyka równoległa, w której wrzeciono z narzędziem jest połączone z podstawą za pomocą wielu niezależnych ramion. Ramiona te poruszają się równolegle do siebie a ich skoordynowany ruch skutkuje realizacją przez narzędzie zadanej trajektorii ruchu. Szczególnym przypadkiem jest kinematyka delta, w której stosuje się trzy niezależne ramiona. Zaletą kinematyki równoległej jest większa prędkość i przyspieszenie efektora oraz większe pole robocze w stosunku do maszyn o kinematyce kartezjańskiej.

Edukacyjny potencjał konstrukcji takiej maszyny był motywacją do budowy stanowiska maszyny numerycznej w układzie Delta, której układ sterowania bazował na oprogramowaniu LinuxCNC.

1.1 . Cel i zakres pracy

Celem pracy było sterowanie maszyną numeryczną w układzie Delta typu Kossel. Wykorzystana została istniejąca struktura mechaniczna z napędami osi, lecz została wyposażona w stół roboczy, czujniki bazowania, czujniki krańcowe osi oraz napęd wrzeciona. W ramach pracy zaprojektowano i wykonano szafę sterowniczą maszyny. Skonfigurowano wszystkie elementy wykonawcze. W szczególności przeprowadzono procedurę konfiguracji napędów serwokrokowych osi oraz strojenie ich regulatorów.

Sterowanie zrealizowano w systemie Linux z jądrem czasu rzeczywistego RT-Preempt w formie bloków programowych HAL oprogramowania LinuxCNC. Przeprowadzono badania i kalibrację urządzenia oraz opracowano dokumentację techniczną wraz ze schematem elektrycznym.

2. Układy sterowania numerycznego

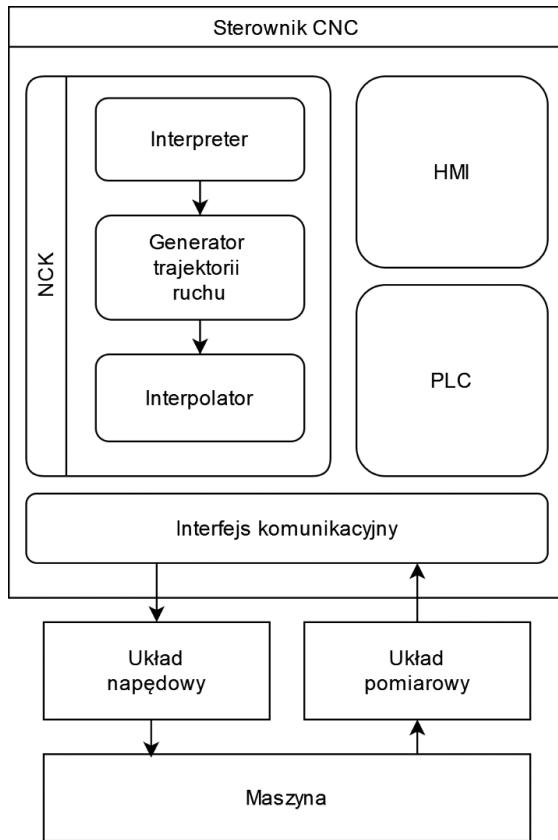
2.1. Wprowadzenie

Początek stosowania maszyn numerycznych poprzedza wprowadzenie ręcznych obrabiarek ogólnego przeznaczenia, które nastąpiło po opracowaniu silnika parowego pod koniec osiemnastego wieku. Z kolei Jacquard był pionierem metody automatycznego tkania tkanin na krośnie za pomocą kart dziurkowanych, kładąc podwaliny pod sterowanie numeryczne (NC). Po II wojnie światowej, w 1947 roku, Siły Powietrzne Stanów Zjednoczonych i firma Parsons jednocześnie opracowały technikę przesuwania suportów obrabiarek wzdłuż dwóch osi wykorzystując napędy sterowane za pomocą kart dziurkowanych zawierających kolejne współrzędne narzędzia wykonawczego. Rozwiązanie to zostało wykorzystane w obróbce części samolotów. Innowacja ta została później rozwinięta w laboratorium serwomechanizmów MIT, gdzie w marcu 1952 roku opracowano pierwszą obrabiarkę NC, 3-osiową frezarkę. W tym okresie, do budowy układów sterowania, stosowano lampy próżniowe, w wyniku czego system NC był większy niż same obrabiarki. Dalsze prace badawcze i rozwojowe doprowadziły do praktycznego zastosowania maszyn numerycznych. W Stanach Zjednoczonych komercyjne frezarki NC zostały uruchomione przez firmy takie jak Giddings & Lewis, Kearney & Tracker oraz Pratt & Whitney. Koncepcja NC, szeroko rozpowszechniona w amerykańskich czasopismach naukowych, dotarła również do Japonii. Do 1957 roku Japonia opracowała własną tokarkę NC. [1]

Sterowanie numeryczne jest systemem, który automatyzuje obrabiarkę poprzez zaprogramowanie poleceń wysyłanych do serwonapędów. Umożliwia to precyzyjny i powtarzalny ruch narzędzia bądź obrabianego przedmiotu. Początkowo sterowanie było zrealizowane za pomocą elektrycznych połączeń tranzystorów, obwodów i układów logicznych. Znaczący skok w ewolucji tych maszyn nastąpił wraz z integracją komputerów, przechodząc od sterowania numerycznego (NC) do CNC. Zamiast obwodów elektrycznych sterowanie numeryczne zaimplementowano programowo na komputerze. Integracja ta pozwoliła na bardziej złożone, precyzyjne i łatwe do zmiany instrukcje maszynowe, znacznie rozszerzając możliwości i zastosowania tych maszyn. W latach 60. technologia CNC zaczęła przekształcać produkcję, czyniąc zautomatyzowaną i precyzyjną obróbkę szeroko dostępną i opłacalną. Dziś termin NC i CNC jest tożsamy i oznacza sterowanie numeryczne. Maszyna numeryczna jest więc maszyną wykorzystującą sterowanie numeryczne. [2]

2.2. Architektura sterowania numerycznego

Sterowanie numeryczne obejmuje szereg kilku współpracujących ze sobą elementów. W ogólności składa się ono z: nadzawanego sterownika CNC, urządzeń wykonawczych takich jak serwonapędy osi i wrzeciona, urządzeń pomocniczych takich jak układy wejść-wyjść oraz czujniki, a także części mechanicznej maszyny.



Rysunek 1. Typowa architektura sterowania CNC

Centralną częścią układu sterowania jest sterownik CNC, który ze względu na funkcjonalność można podzielić na interfejs operatora HMI (ang. Human-Machine Interface), interfejsy komunikacyjne oraz jądro sterowania numerycznego NCK (ang. Numerical Control Kernel) [1] odpowiadające za generowanie pozycji zadanych elementu wykonawczego na podstawie programu. Z reguły, w skład sterownika wchodzi również programowalny logiczny sterownik PLC (ang. Programmable Logic Controller).

Jądro NCK interpretuje instrukcje sterujące, zdefiniowane w jednym z wariantów języka G-Code, generuje trajektorię zaprogramowanego ruchu i wysyła cyklicznie polecenia sterujące. Polecenia te trafiają do układów napędowych serwo, które obracając się powodują przemieszczenie narzędzia. Szereg czujników przesyła informacje zwrotną do sterownika o rzeczywistej pozycji napędów tworząc pętlę sterowania.

2.3. Jądro sterowania numerycznego

Interpreter

Interpreter to rodzaj programu, który wykonuje instrukcje zapisane w postaci kodu źródłowego programu. Interpreter czyta kod, zazwyczaj linijka po linijce, a następnie wykonuje go. Najczęściej używanym językiem programowania maszyn CNC jest G-Code określony przez standardy RS-274D, ISO 6983 lub DIN 66025. Język G-Code (potocznie zwany G-kodem lub G-kodami) stanowi zestaw instrukcji sterujących ruchem maszyny, określając jego parametry takie jak ścieżka, prędkość czy kierunek.

```
%  
.0000  
N10 (PROGRAM PRODUCED - 19 FEB 19)  
N20 G90 G71  
N30 G40 G80  
'(OP 1 FINISH PASS TOOL 4 FLAT - 20MM)  
'(EFFECTIVE DIAMETER 20)  
N40 T0404 M03 'Select tool and offset  
N50 S999 H04 M06 'Next tool is 00, Next XY is 42.667, 37.412  
N60 G0 X42.667 Y37.412  
N70 G43 Z25. H04 M09  
N80 Z5.  
N90 G1 Z-18. F200  
N100 X13.333 Y52.889 F400  
N110 G2 Y97.111 R25.  
N120 G1 X72. Y128.066  
N130 G2 X128. R60.  
N140 G1 X186.667 Y97.111  
N150 G2 Y52.889 R25.  
N160 G1 X128. Y21.934  
N170 G2 X72. R60.  
N180 G1 X42.667 Y37.412  
N190 G0 Z25.  
N200 X100. Y100.  
N210 Z5.  
N220 G1 Z-18. F200  
N230 G2 X121.653 Y62.505 R25. F400  
N240 X78.347 R25.  
N250 X100. Y100. R25.  
N260 G0 Z25.  
N270 M09  
N280 M30  
%  
cadcamlessons.com
```

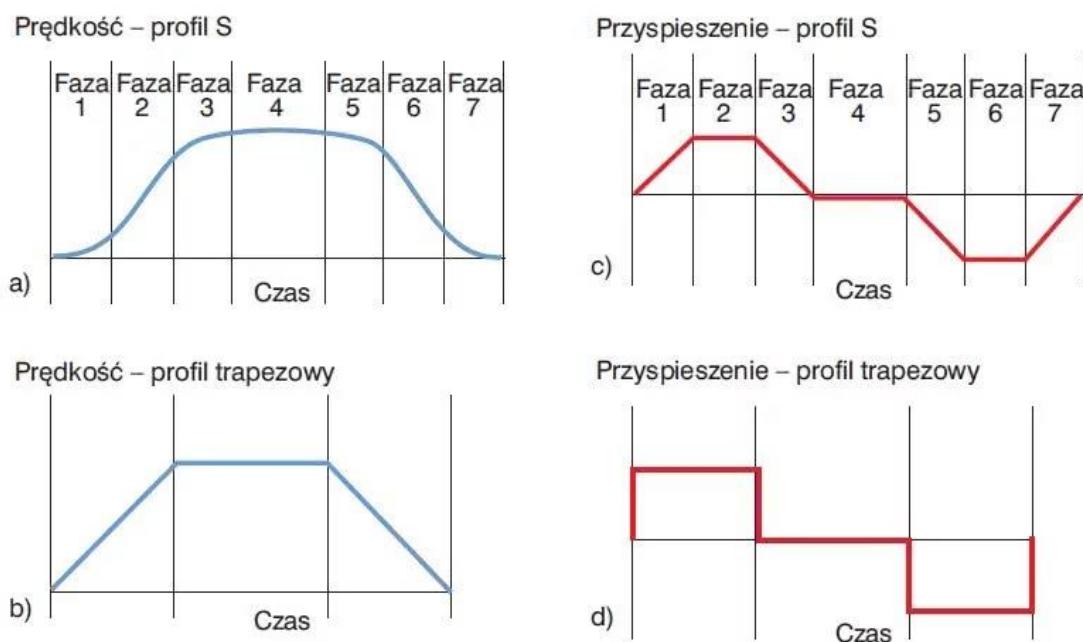
Rysunek 2. Język programowania G-code. Źródło: [3]

W sterowniku CNC interpretacja polega na odczytaniu i przetłumaczeniu G-kodu na komendę ruchu jaka zostanie przekazana do dalszych bloków: generatora trajektorii i interpolatora, a następnie wysłana do napędów. Przykładowa instrukcja G1 określa ruch prostoliniowy do punktu docelowego z określoną w parametrze prędkością posuwu. Instrukcje G2 i G3 oznaczają z kolei ruch po łuku o określonym środku i promieniu zgodnie (G2) lub przeciwnie (G3) do ruchu wskazówek zegara. Pozostałe instrukcje definiują szereg innych funkcji technologicznych takich jak załączanie wrzeciona, offsety (z ang. przesunięcia), cykle technologiczne (np. wiercenie), pętle i instrukcje warunkowe. Standaryzacja języka G-Code nie jest pełna i istnieje wiele wariantów z instrukcjami specyficznymi dla danego rodzaju sterowania. [4]

Generator trajektorii ruchu

W rzeczywistym układzie istnieją ograniczenia fizyczne takie jak maksymalna prędkość czy przyspieszenie osi. Nie jest możliwe, aby napęd osiągnął prędkość docelową w zerowym czasie. Z tego powodu generator trajektorii profiluje trajektorię tak, aby zapewnić osiągnięcie planowanej ścieżki ruchu przy określonym profilu prędkości i przyspieszenia.

Istnieje wiele rodzajów profilowania trajektorii. Często używanymi rodzajami są trapezoidalne profilowanie prędkości (Rysunek 3, podpunkty b i d) oraz profilowanie po krzywej S (Rysunek 3, podpunkty a i c).



Rysunek 3. Rodzaje profilowania prędkości. Źródło: [5]

Trapezowe profilowanie prędkości składa się z trzech oddzielnych faz: przyspieszania, stałej prędkości i zwalniania. Ten typ profilowania jest często stosowany w układach, w których kluczowe znaczenie ma szybkie osiągnięcie maksymalnej prędkości. Jest on prostszy w realizacji od profilowania po krzywej S. Do wad należą gwałtowne zmiany przyspieszenia między fazami, które mogą powodować mechaniczne naprężenia i oscylacje.

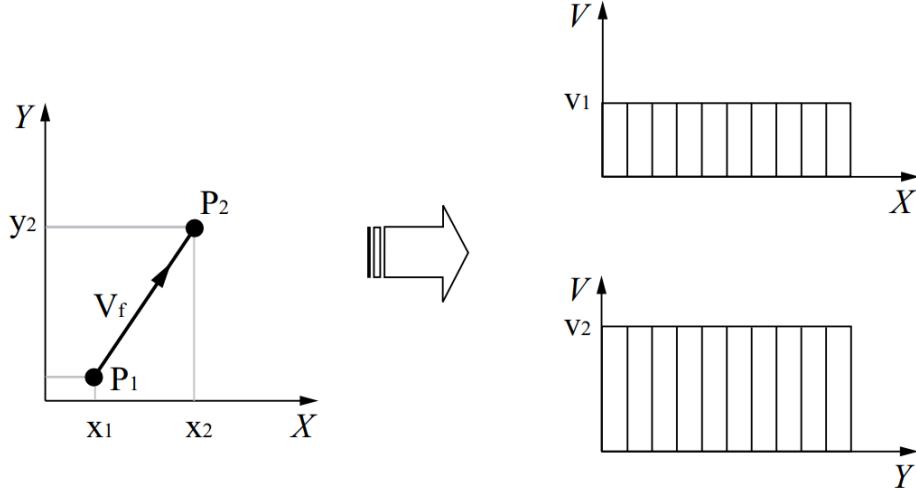
Profilowanie prędkości po krzywej S wprowadza stopniowe zmiany w fazach przyspieszania i zwalniania. Profil ten ma bardziej gładki przebieg od profilu trapezowego i lepiej nadaje się do zastosowań wymagających wysokiej precyzji. Stopniowe zmiany przyspieszenia zmniejszają naprężenia i wibracje, co przekłada się na płynniejszą pracę i większą żywotność układu.

Interpolator

Interpolacja to matematyczna metoda znajdowania pośrednich punktów (wartości) pomiędzy dwoma znanymi punktami funkcji. W CNC interpolacja to proces wyznaczania położen pośrednich pomiędzy punktami skrajnymi określonego kształtu (linii, łuku, krzywej) w programie technologicznym. Dodatkowo w procesie interpolacji możliwe jest wygładzanie toru, szczególnie w przypadku kształtów zdefiniowanych za pomocą linii prostych. [6]

Interpolator odpowiada za obliczenie współrzędnych pośrednich punktów dla każdej z osi pomiędzy punktem początkowym i końcowym ruchu. Ma to na celu skoordynowanie ruchów wszystkich osi i precyzyjne określenie ścieżki ruchu. Najprostszym rodzajem interpolacji jest interpolacja liniowa w której narzędziem przemieszcza się od punktu startowego do punktu końcowego wzdłuż prostej linii. Interpolator liniowy oblicza jak każda z osi powinna się poruszać aby w efekcie uzyskać płynny ruch narzędzia wzdłuż tej linii.

Przykładowo gdy narzędzie musi przemieścić się z punktu P₁ do P₂ przy prędkości posuwu V_f na płaszczyźnie XY, tak jak pokazano na rys. 4, interpolator podzieli cały ruch na poszczególne przemieszczenia wzdłuż osi X i Y w oparciu o określoną prędkość posuwu. [1]



Rysunek 4. Idea działania interpolatora. Źródło: [1] s.86

Innym rodzajem interpolacji jest interpolacja kołowa, gdzie punkty pośrednie leżą na łuku o określonym promieniu oraz środku. Interpolacja może być również realizowana dla toru ruchu opisanego krzywymi wielomianowymi np. B-Spline lub NURBS. Interpolacja realizowana jest z reguły w kartezjańskim układzie współrzędnych, zarówno na płaszczyźnie jak i w przestrzeni. Dla bardziej zaawansowanych maszyn, interpolacji podlega też orientacja narzędzia jak ma to miejsce w maszynach 5-osiowych.

2.4. Interfejs komunikacyjny

Jednostki napędowe otrzymują od sterownika cyklicznie kolejno zadawane pozycje. W tym celu stosuje się interfejs, który pozwala na komunikację z napędem. Obecnie w prostych maszynach stosuje się interfejs impulsowy (ang. Pulse Train). Prędkość silnika zależy wówczas od częstotliwości impulsów. Często dodatkowo stosuje się sygnał DIR (ang. Direction) określający kierunek obrotów.



Rysunek 5. Ciąg impulsów jako interfejs napędowy. Źródło: [7]

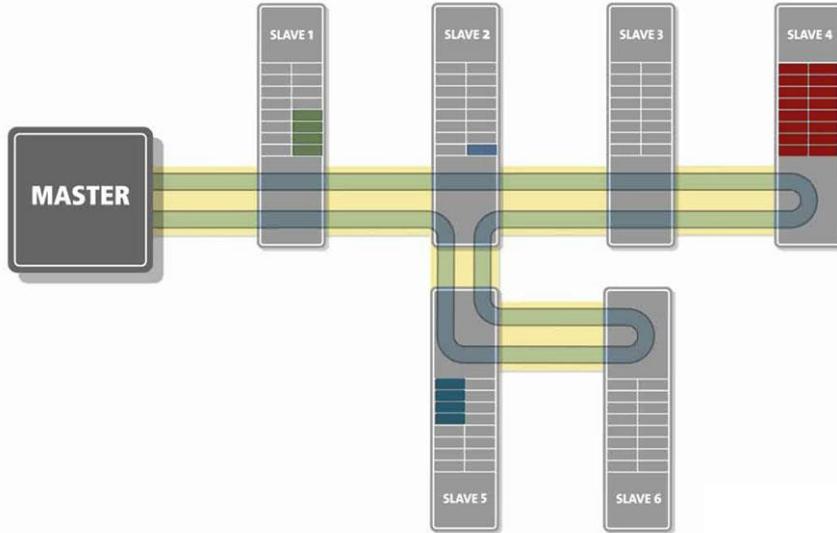
Bardziej rozbudowanym interfejsem jest komunikacja poprzez magistrale szeregowe wykorzystujące przemysłowe protokoły komunikacyjne. Dawniej były to protokoły bazujące na magistralach szeregowych takich jak RS-485 lub CAN. Obecnie najczęściej stosuje się protokoły takie jak Profinet, EtherCAT oraz Ethernet/IP bazujące na warstwie fizycznej i łączących magistrali Ethernet.

2.4.1. EtherCAT

EtherCAT, skrót od Ethernet for Control Automation Technology, to szybki protokół komunikacyjny zaprojektowany specjalnie do wydajnego sterowania w czasie rzeczywistym w automatyce przemysłowej, w szczególności w zakresie sterowania ruchem napędów elektrycznych. Protokół ten umożliwia szybką komunikację między centralnym sterownikiem, takim jak sterownik CNC, a różnymi urządzeniami, w tym napędami lub modułami wejść-wyjść. Charakteryzuje się wysoką przepustowością danych i wysokim determinizmem czasowym komunikacji.

Protokół EtherCAT różni się od typowych magistrali tym, że wykorzystuje metodę, w której pojedynczy pakiet danych jest przesyłany sekwencyjnie do każdego urządzenia w sieci, w przeciwieństwie do wysyłania danych do każdego węzła. Gdy pakiet ten przechodzi przez sieć, każde urządzenie interpretuje dane przeznaczone specjalnie dla niego i natychmiast

dodaje swoją odpowiedź do pakietu. Podejście to jest powszechnie określane jako "przetwarzanie w czasie rzeczywistym". Gdy pakiet danych przejdzie przez wszystkie urządzenia i powróci do kontrolera, kontroler uzyskuje odpowiedź od wszystkich urządzeń w sieci (Rysunek 6). Przetwarzanie pakietu danych „w locie” umożliwia osiągnięcie czasu pojedynczego cyklu poniżej 100 µs. Przekłada się to na typową częstotliwość aktualizacji danych procesowych w zakresie od 1 do 30kHz. [8]



Rysunek 6. Mechanizm komunikacji w sieci EtherCAT. Źródło: [9]

Warstwa fizyczna i łącza danych sieci EtherCAT jest kompatybilna ze standardem Ethernet (IEEE 802.3). Po stronie sterownika, którym często jest komputer klasy PC, można stosować standardowe karty sieciowe. Po stronie urządzeń podrzędnych konieczne jest zastosowanie specjalistycznego układu sieciowego ESC (EtherCAT Slave/SubDevice Controller). Jako warstwę aplikacyjną sieci EtherCAT najczęściej wykorzystuje się protokół CANopen.

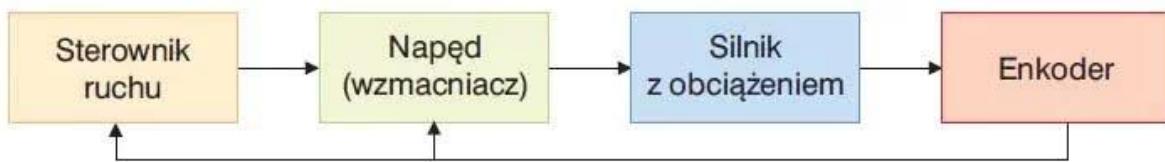
2.4.2. CANopen

CANopen to protokół komunikacyjny i specyfikacja profilów urządzeń (ang. device profiles) stosowana w dziedzinie automatyki. Każde urządzenie korzystające z protokołu CANopen określa zbiór obiektów tworzących słownik obiektów (ang. Object Dictionary). Jest to ustandaryzowany interfejs dostępu do danych urządzenia. Każdy obiekt w słowniku jest mapowany poprzez jego indeks oraz opcjonalnie subindeks. Obiekty danych procesowych PDO (ang. Process Data Object) zawierają dane przesypane cyklicznie w czasie rzeczywistym. Obiekty danych serwisowych SDO (ang. Service Data Object) używane są do wymiany danych niekrytycznych czasowo takich jak parametry konfiguracyjne urządzenia.

Profile urządzeń CiA (ang. CAN in Automation) określają standard zachowania i charakterystyki różnych klas urządzeń w sieci. Profil 402 definiuje interfejs dla napędów i sterowników ruchu. Określa on tryby pracy, obiekty danych procesowych PDO, takie jak słowo kontrolne (ang. control word) lub prędkość napędu, a także obiekty SDO służące do konfiguracji napędu. Kluczowym komponentem profilu 402 jest maszyna stanów definiująca dozwolone stany napędu oraz przejścia pomiędzy stanami. Obecny stan urządzenia określony jest w słowie statusowym (ang. status word). Zmiana stanu urządzenia następuje poprzez słowo kontrolne otrzymane od kontrolera (mastera) sieci.

2.5. Serwonapęd

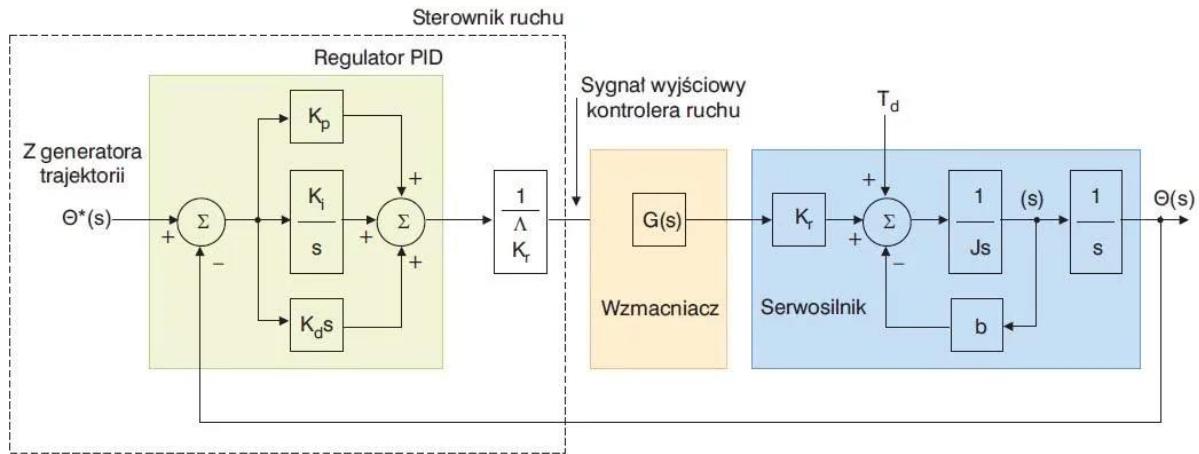
Serwonapęd to zespół urządzeń służących do precyzyjnego sterowania pozycją, prędkością i momentem obrotowym silnika. Serwonapęd składa się z: sterownika ruchu, wzmacniacza, silnika oraz enkodera.



Rysunek 7. Schemat blokowy serwonapędu. Źródło: [5]

Sterownik ruchu wysyła sygnały sterujące, które zostają wzmocnione a następnie dostarczone do silnika co przekłada się na jego ruch. Enkoder odpowiada za pomiar rzeczywistej zmiany pozycji wału silnika dostarczając do sterownika dane o położeniu i prędkości. Sterownik znając rzeczywistą pozycję może skorygować sygnał sterujący. W ten sposób serwonapędy mogą pracować w pętli zamkniętej regulacji. Serwonapędy stosowane są w automatyce i robotyce tam, gdzie potrzebna jest wysoka precyzja ruchu. Ich zastosowania obejmują między innymi maszyny CNC.

Serwonapęd stanowi układ regulacji, w którym regulatorem jest sterownik ruchu, elementem wykonawczym wzmacniacz, zaś obiektem regulacji serwosilnik. W sterowniku często stosowany jest regulator proporcjonalno-różniczkującco-całkujący PID (ang. Proportional-Integral-Differential). Na podstawie pozycji zadanej z generatora trajektorii $\Phi^*(s)$ i aktualnej pozycji wału silnika obliczany jest uchyb sterowania $E(s)$. Regulator w oparciu o sygnał uchybu generuje sygnał sterujący w celu osiągnięcia zadanej pozycji i prędkości, a także momentu wału silnika. Serwosilnik przetwarza sygnał sterujący na rzeczywistą pozycję wału silnika $\Phi(s)$. Taki układ regulacji uwzględnia wpływ momentu bezwładności własnej silnika i momentu obciążenia T_d silnika.



Rysunek 8. Uproszczony model serwonapędu w domenie Laplace'a. Źródło: [5]

Dawniej serwonapędy ograniczały się do napędów prądu stałego (DC) po czym zaczęto stosować je dla silników synchronicznych z magnesami trwałymi PMSM (ang. permanent magnet synchronous motor). Obecnie w mniej zaawansowanych zastosowaniach stosuje się napędy serwo-krokowe w których silnik krokowy z enkoderem sterowany jest metodą zorientowaną polowo.

2.6. Rodzaje układów sterowania

Układy sterowania są dzielone na zamknięte i otwarte. Te pierwsze mają z góry określona strukturę nie pozwalającą na zmianę funkcjonalności. Wielu znanych producentów układów automatyki tacy jak Siemens, Heidenhain, Fanuc czy Mitsubishi produkuje zamknięte układy. Przykładowo Siemens produkuje sterowniki zintegrowane Sinumerik do obrabiarek wieloosiowych. [10]



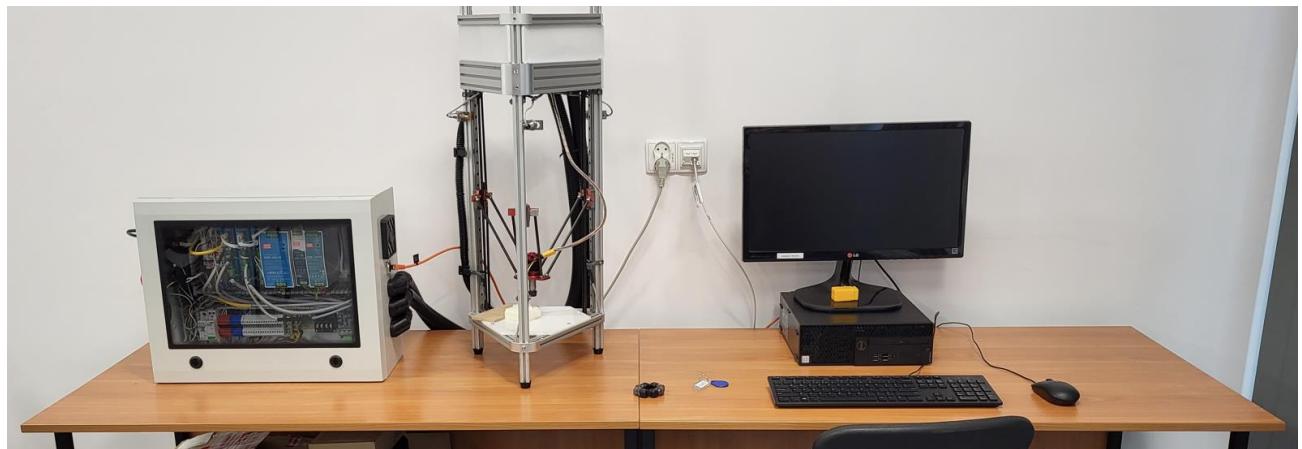
Rysunek 9. Sterownik CNC Sinumerik 808D ADVANCED. Źródło: [11]

Przykładem częściowo otwartego układu sterowania jest oprogramowanie TwinCAT firmy Beckhoff. Jest to elastyczne oprogramowanie zawierające w sobie funkcjonalność PLC oraz możliwość sterowania ruchem. Obsługuje ono wiele interfejsów napędowych takich jak pulse train czy własnościowy EtherCAT. Oprogramowanie TwinCAT może zostać uruchomione na komputerach klasy przemysłowej IPC (ang. Industrial Personal Computer) dając większą swobodę i elastyczność. [10] Układy sterowania otwartego mogą być swobodnie modyfikowane dla różnych konfiguracji maszyn. Jednym z nich jest sterowanie bazujące na wolnym i otwartym oprogramowaniu FOSS (ang. Free and Open-Source Software) takim jak LinuxCNC oraz Machinekit. Oprogramowanie to może pracować na komputerach IPC oraz zwykłych komputerach PC czyniąc je łatwo dostępnymi.

Sterownik CNC coraz częściej zbudowany jest w oparciu o komputer klasy PC. Jest to możliwe dzięki implementacji algorytmów sterowania numerycznego, w szczególności jądra sterowania numerycznego, (ang. Numerical Control Kernel, NCK) w formie programu komputerowego działającego w systemie operacyjnym komputera sterującego. Często są to systemy operacyjne czasu rzeczywistego umożliwiające realizację algorytmów sterowania w sposób deterministyczny czasowo. W tym celu konieczne jest zastosowanie jądra czasu rzeczywistego takiego jak otwarte jądra Linux RTAI oraz Linux RT Preempt. Sterownik CNC zbudowany w oparciu o PC dzieli czas procesora na zadania krytyczne i niekrytyczne czasowo. Do tych pierwszych należy jądro sterowania numerycznego oraz komunikacja z napędami. Do zadań niekrytycznych należy interfejs graficzny GUI, funkcjonalność pomocnicza (np. przetwarzanie układów peryferyjnych wejść/wyjść) i zadania poboczne takie jak akwizycja danych [12]. Zastosowanie jądra czasu rzeczywistego umożliwia również realizację wysoko deterministycznej komunikacji szeregowej z napędami osi maszyny. Jest to niezbędne do realizacji skoordynowanych przemieszczeń narzędzia roboczego realizującego zadaną trajektorię ruchu.

3. Budowa stanowiska

Stanowisko składa się z szafy sterowniczej, maszyny numerycznej Delta-Kossel oraz komputera stacjonarnego Dell OptiPlex 3070. Konstrukcja maszyny została zbudowana z modułarnego systemu Kossel przeznaczonego dla maszyn w układzie typu delta.

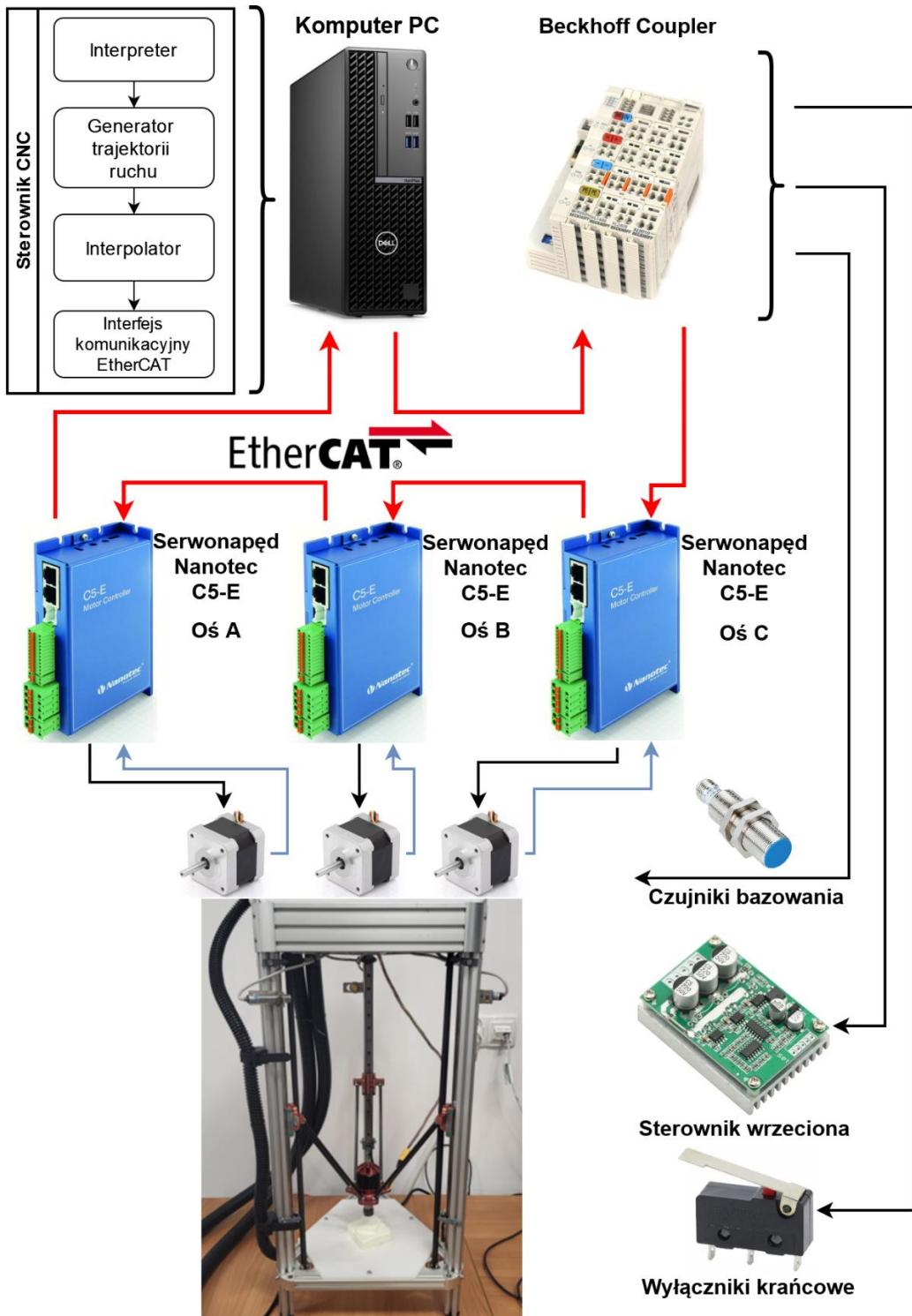


Rysunek 10. Stanowisko z maszyną numeryczną.



Rysunek 11. Maszyna numeryczna w układzie Delta typu Kossel.

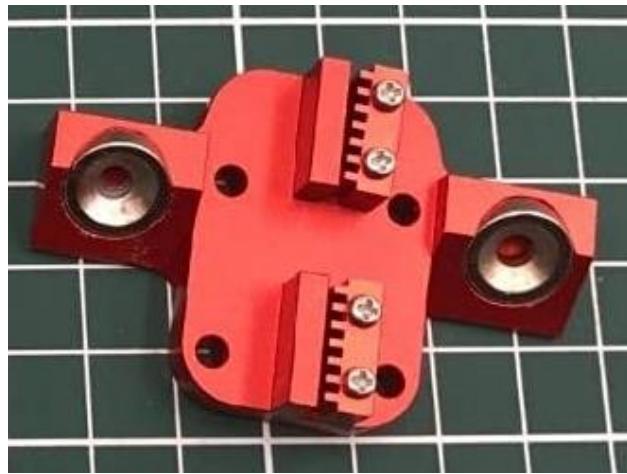
Poniżej przedstawiono schemat blokowy stanowiska. Sterownikiem maszyny jest komputer stacjonarny Dell OptiPlex 3070 z 6-rdzeniowym procesorem Intel Core i5-9500, dyskiem SSD 1TB oraz 16GB pamięci RAM. Komunikacja sterownika z napędami i urządzeniami peryferyjnymi odbywa się poprzez sieć EtherCAT o fizycznej topologii liniowej (ang. daisy-chain) i logicznej topologii pierścieniowej.



Rysunek 12. Schemat blokowy stanowiska

3.1. Komponenty maszyny numerycznej typu Delta liniowa

Maszyna w układzie delta liniowa składa się z nieruchomego stołu oraz trzech kolumn – osi liniowych maszyny w konfiguracji trójkąta równobocznego. Na każdej kolumnie zamocowana jest prowadnica po której porusza się wózek napędzany paskiem zębatym:



Rysunek 13. Wózki prowadnicy liniowej maszyny

Wózek połączony jest z efektorem za pomocą pary ramion wykonanych z aluminium o nominalnej długości 180mm i posiadających magnetyczne uchwyty służące jako złącza sferyczne maszyny:



Rysunek 14. Magnetyczne ramiona maszyny

Wszystkie ramiona łączą się z efektorem wykonanym z aluminium na którym zamocowane zostało wrzeciono.



Rysunek 15. Efektor maszyny połączony magnetycznie z ramionami

Za ruch każdego wózka osi odpowiada silnik krokowy Nanotec ST48118L3004-B o prądzie nominalnym 3A i rozdzielczości 1.8°/krok.



Rysunek 16. Silnik krokowy Nanotec ST48118L3004-B serii NEMA 17. Źródło: [13]

Do silnika dołączono optyczny enkoder inkrementalny NOE2-05-B14 o rozdzielczości 16000 inkrementów na obrót.



Rysunek 17. Enkoder inkrementalny NOE2-05-B14. Źródło: [14]

Silniki osi napędzane są przez sterownik Nanotec C5-E-1-21 przeznaczony do silników krokowych i silników bezszczotkowych prądu stałego BLDC (ang. brushless direct current) charakteryzujący się prądem nominalnym 6A i napięciem w zakresie 12-48VDC.

Sterownik Nanotec C5-E posiada swobodnie konfigurowalne wejścia i wyjścia, obsługę magistrali EtherCAT oraz interfejsu USB. Sterownik obsługuje pracę zarówno w pętli otwartej jak i zamkniętej.



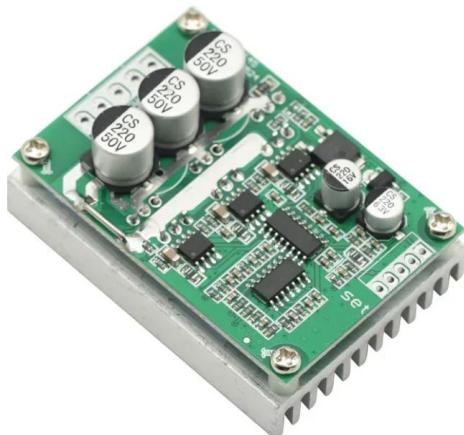
Rysunek 18. Serwonapęd Nanotec C5-E-1-21. Źródło: [15]

Napędem wrzeciona jest silnik bezszczotkowy prądu stałego FEICHAO C4250 560KV. Silniki BLDC charakteryzują się większą gęstością mocy w stosunku do silników szczotkowych o tych samych wymiarach.



Rysunek 19. Silnik bezszczotkowy FEICHAO C4250 560KV. Źródło: [16]

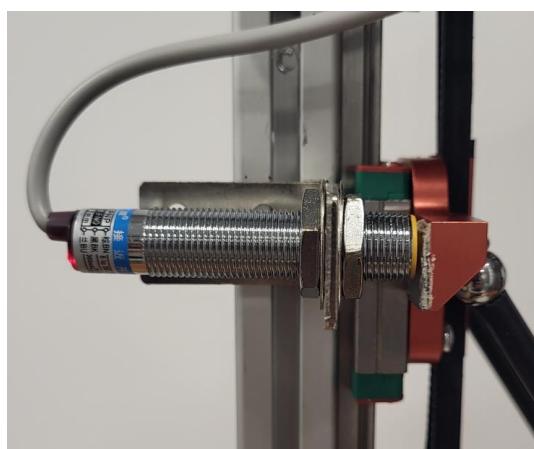
Silnik bezszczotkowy prądu stałego sterowany jest z wykorzystaniem komutacji elektronicznej. Z tego powodu wymagana jest znajomość położenia wału. Silniki BLDC mogą pracować w pętli zamkniętej z czujnikami Halla bądź w pętli otwartej z estymacją położenia wału. Zdecydowano się na drugą metodę i zastosowano elektroniczny sterownik prędkości (ang. Electronic Speed Controller) ZUHEGELA 500W 15A DC12-36V estymujący położenie wału za pomocą pomiaru siły elektromotorycznej indukowanej w uzwojeniach silnika.



Rysunek 20. Elektroniczny regulator prędkości ZUHEGELA 500W 15A DC12-36V. Źródło: [17]

Zadawanie prędkości do sterownika może odbywać się za pomocą potencjometru lub sygnału PWM o napięciu 5V. Wykorzystano zadawanie prędkości poprzez wyjście PWM modułu Beckhoff EL2502 dołączonego do sprzęgacza (coupler'a) EtherCAT. Sygnał PWM o napięciu 24V dostosowano do akceptowalnego przez sterownik napięcia 5V poprzez zastosowanie dzielnika napięcia.

Czujnikami bazowania osi są czujniki indukcyjne LJ12A3-4-Z/BX typu NPN zasilane napięciem 24VDC. Czujniki zamontowano na uchwycie kątowym przykręconym do profilu aluminiowego stanowiącego element konstrukcji maszyny.



Rysunek 21. Czujnik indukcyjny NPN LJ12A3-4-Z/BX do bazowania osi

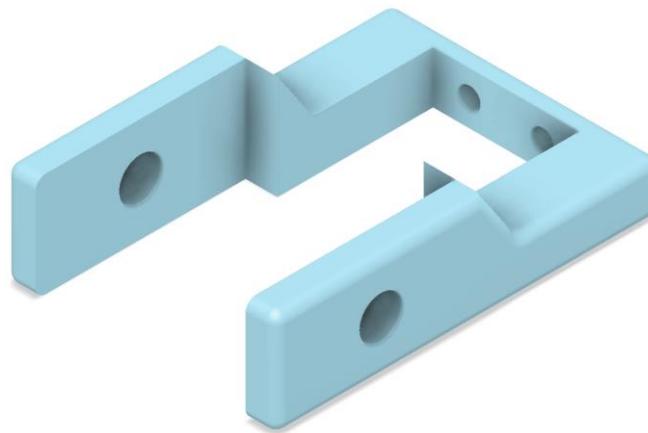
Czujniki indukcyjne typowo kalibrowane są do wykrywania stali, a ich czułość względna jest mniejsza dla innych rodzajów metalu. W celu zwiększenia dystansu wykrywania do aluminiowego wózka przyklejono stalową blaszkę.

Na końcach dozwolonego zakresu ruchu każdej z osi zamocowano mechaniczne wyłączniki krańcowe, aktywowane w momencie najazdu wózka. Sygnał z wyłączników jest przekazywany do sterownika silnika właściwego dla danej osi, który zatrzymuje silnik.



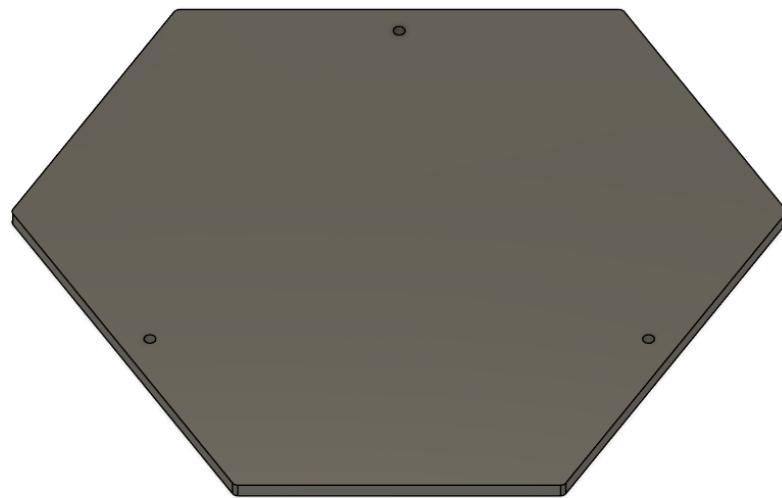
Rysunek 22. Wyłączniki krańcowe osi maszyny numerycznej

Uchwyt pod wyłącznik krańcowy zaprojektowano w programie SOLIDWORKS, a następnie model wydrukowano na drukarce 3D.



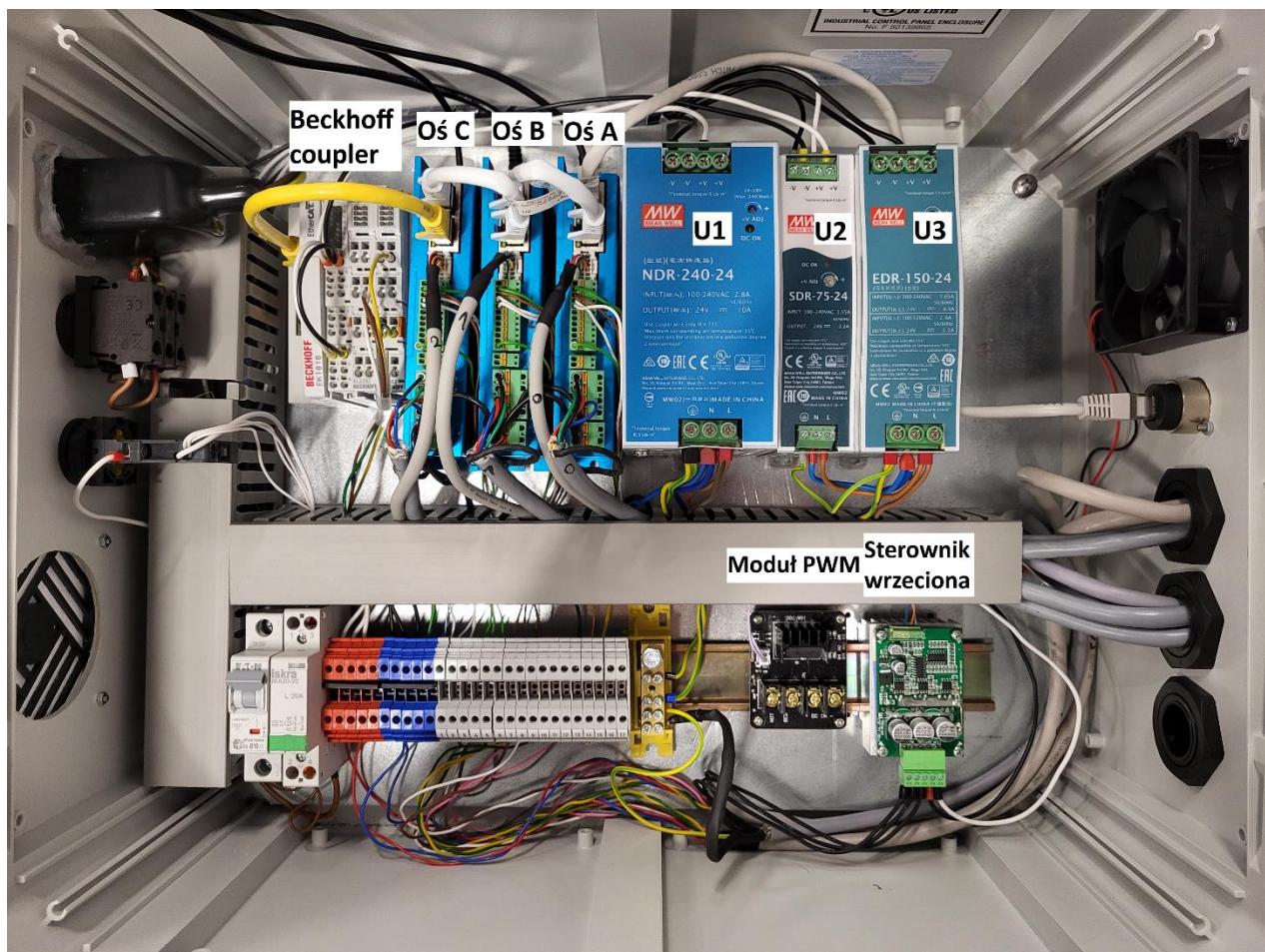
Rysunek 23. Model uchwytu pod wyłącznik krańcowy wykonany w programie SOLIDWORKS

Stół maszyny zaprojektowano w programie Autodesk Fusion 360, wygenerowano program z G-kodami za pomocą preprocesora LinuxCNC (EMC2). Następnie wyfrezowano stół z prefabrykatu (pleksi) na frezarce sterowanej oprogramowaniem LinuxCNC.



Rysunek 24. Model stołu maszyny numerycznej wykonany w programie Autodesk Fusion 360

Podzespoły sterownika umieszczone w szafie sterującej na szynach DIN. Zasilanie załączane i wyłączane jest za pomocą przycisków chwilowych START-STOP z samo-podtrzymaniem na styczniaku.



Rysunek 25. Szafa sterująca

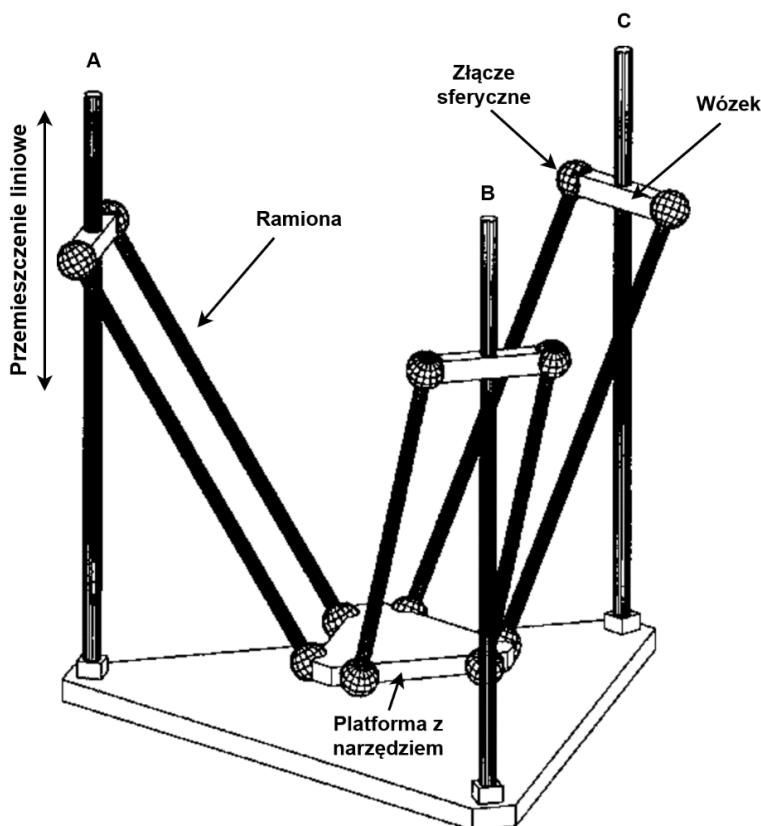
Sekcja zasilająca złożona jest z trzech zasilaczy 24V:

- U1 - MEAN WELL NDR-240-24 o maks. prądzie 10A i mocy 240W do zasilania serwonapędów osi
- U2 - MEAN WELL SDR-75-24 o maks. prądzie 3.2A i mocy 75W do zasilania logiki modułów wejść/wyjść
- U3 - MEAN WELL EDR-150-24 o maks. prądzie 6.5A i mocy 150W do zasilania napędu wrzeciona i dodatkowych peryferii

Jako układ wejścia-wyjścia zastosowano EtherCAT Coupler EK1818 firmy Beckhoff. Jest to urządzenie pełniące funkcję mostu pomiędzy fizycznymi sygnałami a siecią EtherCAT. Układ ten posiada 8 wejść cyfrowych i 8 wyjść cyfrowych. Dodatkowo zastosowano moduł rozszerzeń Beckhoff EL2502 posiadający 2 wyjścia PWM. Jedno z nich wykorzystano jako sygnał sterujący prędkością wrzeciona.

3.2. Kinematyka maszyny typu Delta liniowa

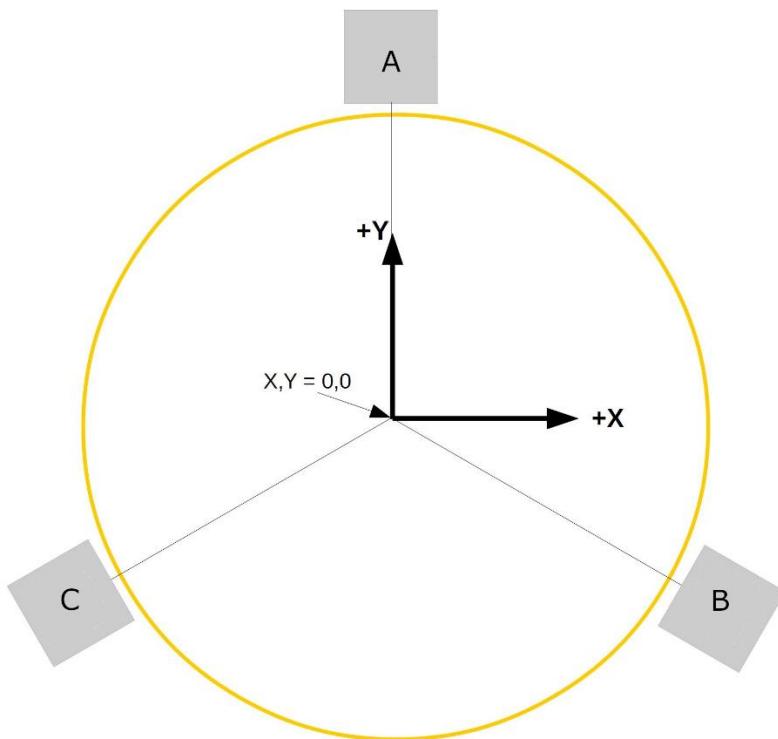
Maszyna numeryczna w układzie typu delta liniowa składa się z trzech równoległych kolumn A, B, C w konfiguracji trójkąta równobocznego gdzie każda kolumna jest prostopadła do płaszczyzny podstawy maszyny i przecina się z nią w wierzchołku trójkąta.



Rysunek 26. Struktura manipulatora typu delta liniowa. Źródło: [18]

Każda kolumna ma złącze przesuwne w postaci wózka poruszającego się po prowadnicy. Przesunięcie liniowe wózka realizowane jest poprzez ruch obrotowy silnika krokkowego sprzążonego za pomocą paska na kole zębatym o średnicy 10mm. Każdy wózek połączony jest z platformą narzędziową za pomocą pary ramion o identycznej długości. Na końcach ramion znajdują się złącza sferyczne. Narzędzie umieszczone w środku platformy posiada 3 stopnie swobody. [18] [19]

Jako środek układu współrzędnych przyjęto środek platformy z narzędziem. Do wyznaczenia pozycji platformy konieczne jest przekształcenie kinematyczne. Kolumny A, B, C są umieszczone w odległości R od środka platformy. Oś Y skierowana jest w kierunku kolumny A (Rysunek 27). Wobec tego kolumna A posiada współrzędne oznaczone jako $(0, R)$. [20]



Rysunek 27. Układ współrzędnych maszyny. Źródło: [20]

Współrzędne kolumn B i C są przesunięte względem kolumny A kolejno o 120° .

$$B_x = -\cos(30) * R \quad (1)$$

$$B_y = -\sin(30) * R$$

$$C_x = \cos(30) * R \quad (2)$$

$$C_y = \sin(30) * R$$

Przyjęto, że punkt, w którym wartość współrzędnej wysokości Z wynosi zero znajduje się na płaszczyźnie podstawy (stołu). Wysokość każdego wózka nad stołem można opisać jako [21]:

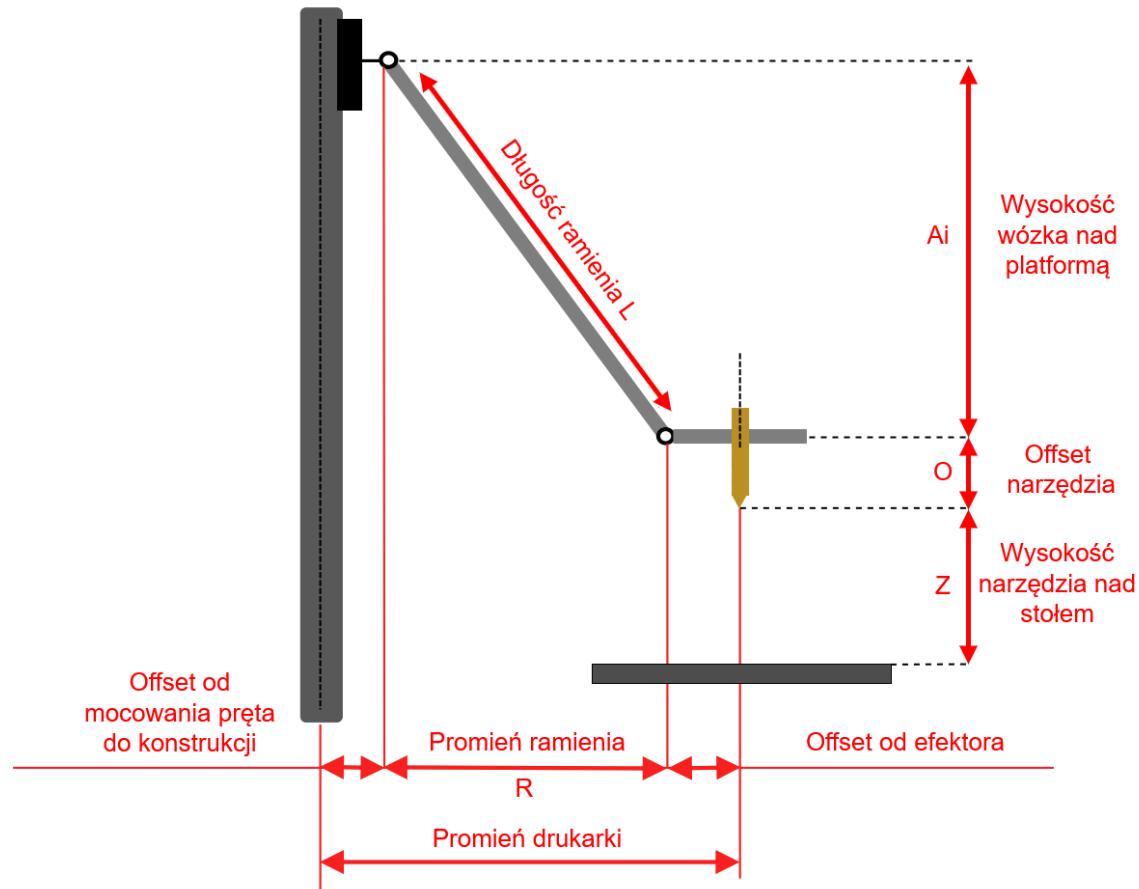
$$\begin{aligned} A &= Z + A_z + O \\ B &= Z + B_z + O \\ C &= Z + C_z + O \end{aligned} \quad (3)$$

Gdzie:

- Z – wysokość narzędzia nad płaszczyzną podstawy
- A_i – wysokość i-tego wózka nad platformą z narzędziem
- O - offset narzędzia (długość na jaką narzędzie wystaje poniżej platformy)

Z równania (3) można wywnioskować, że zmiana pozycji Z narzędzia spowoduje jednakową zmianę wysokości wszystkich wózków. Można zaobserwować, że wysokości wózków nad platformą A_i są zależne tylko od osi X i Y.

Rzut boczny pojedynczego łańcucha kinematycznego przedstawiono na rysunku poniżej:



Rysunek 28. Rzut boczny łańcucha kinematycznego manipulatora o kinematyce typu delta liniowa. Źródło: [22]

Każda para ramion jest takiej samej długości \mathbf{L} . Rzut ramienia na płaszczyznę platformy (promień ramienia) gdy narzędzie znajduje się w punkcie (0,0) na płaszczyźnie XY maszyny wynosi \mathbf{R} .

Z twierdzenia Pitagorasa można wyznaczyć długość ramion następująco:

$$L = \sqrt{A_z^2 + A_d^2} = \sqrt{B_z^2 + B_d^2} = \sqrt{C_z^2 + C_d^2} \quad (4)$$

Gdzie A_d oznacza promień ramienia.

Zakładając, że platforma połączona jest pojedynczymi ramionami można opisać zakres ruchu na płaszczyźnie XY jako część wspólną okręgów o środkach w miejscu prowadnic osi [21]. Poniżej opisano równanie dla kolumny A:

$$(A_x - x)^2 - (A_y - y)^2 = A_d^2 = L^2 - A_z^2 \quad (5)$$

Przekształcając równanie (5) w celu wyznaczenia wysokości wózka nad platformą A_z otrzymuje się:

$$A_z = \sqrt{L^2 - (A_x - x)^2 - (A_y - y)^2} \quad (6)$$

Zestawiając otrzymane równanie z równaniem (3) otrzymuje się równania wyznaczające wysokości kolumn dla zadanego punktu narzędzia (x,y):

$$\begin{aligned} A &= Z + \sqrt{L^2 - (A_x - x)^2 - (A_y - y)^2} + O \\ B &= Z + \sqrt{L^2 - (B_x - x)^2 - (B_y - y)^2} + O \\ C &= Z + \sqrt{L^2 - (C_x - x)^2 - (C_y - y)^2} + O \end{aligned} \quad (7)$$

W szczególności podstawiając do pierwszego równania współrzędne kolumny $A_{xy} = (0, R)$ otrzymuje się:

$$A = Z + \sqrt{L^2 + x^2 - (R - y)^2} + O \quad (8)$$

Analogiczne równania dla kolumn B i C można wyznaczyć podstawiając współrzędne danej kolumny.

4. Struktura oprogramowania sterującego

Sterownikiem maszyny jest komputer stacjonarny Dell Optiflex 3070 z systemem operacyjnym Linux (Debian 11) z wgranym patch'em jądra systemu RT-Preempt umożliwiającym sterowanie numeryczne w czasie rzeczywistym. Komunikacja z napędami i innymi urządzeniami odbywa się za pomocą sieci EtherCAT zapewniającej wymianę danych w czasie rzeczywistym. W tym celu komputer wyposażono w kartę sieciową Intel I350T2M współpracującą ze standardowym sterownikiem Intel e1000e. Zainstalowano otwartoźródłową implementację stosu mastera IgH EtherCAT [23]. W celu zapewnienia determinizmu czasowego należy ustawić parametr *isolcpus* jądra systemu służący do wyizolowania rdzeni procesora wyłącznie dla jądra systemowego. Do konfiguracji programu rozruchowego GRUB (ang. Grand Unified Bootloader) dodano parametr isolcpus. Procesor Intel Core i5-9500 posiada 6 rdzeni. Pierwsze trzy rdzenie procesora wyizolowano na potrzeby jądra systemu.

4.1. Oprogramowanie LinuxCNC

LinuxCNC to zbiór swobodnie konfigurowalnych programów służących do sterowania numerycznego maszynami. Oprogramowanie obsługuje wiele rodzajów konstrukcji takich jak frezarki, obrabiarki, drukarki 3D, roboty, wycinarki laserowe. [4]

Głównymi komponentami z których składa się LinuxCNC są:

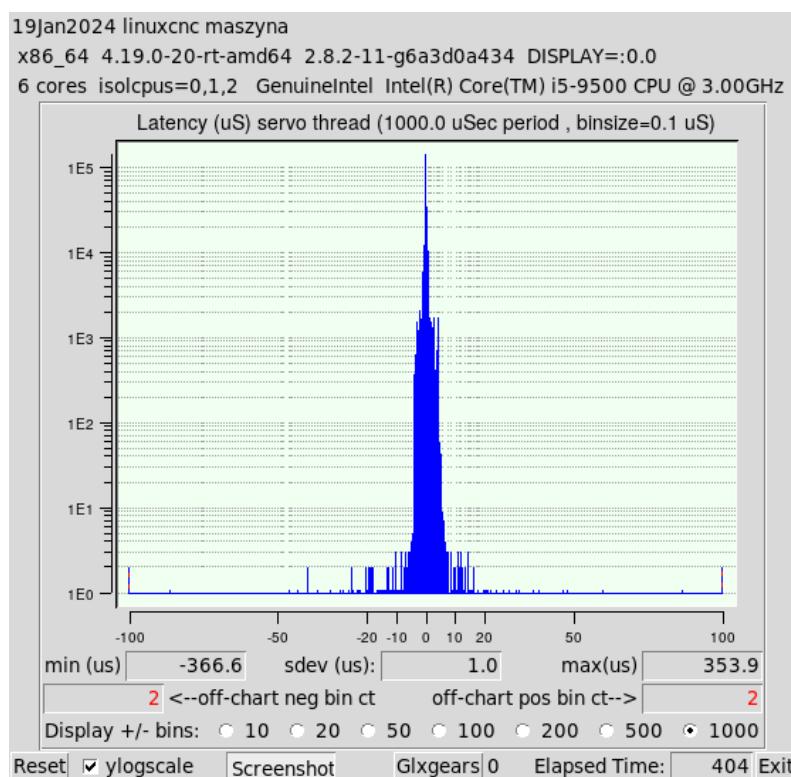
- interfejs graficzny (GUI) pełniący rolę interfejsu człowieka z maszyną (HMI)
- oprogramowanie HAL
- zbiór komponentów implementujących sterowanie numeryczne: interpreter, generator trajektorii ruchu, interpolator, obsługa I/O [4]

HAL (ang. Hardware Abstraction Layer) to ważna część oprogramowania LinuxCNC pełniąca rolę interfejsu pomiędzy fizycznym sprzętem a oprogramowaniem. Oprogramowanie to składa się z komponentów, których kod może być wykonywany na poziomie użytkownika lub w jądrze systemu w celu zapewnienia determinizmu czasu rzeczywistego. Struktura oprogramowania HAL została zaprojektowana na wzór fizycznych układów elektronicznych. Z tego powodu nazewnictwo odzwierciedla takie układy. Komponenty HAL definiują własne „piny” czyli zmienne wejściowe i wyjściowe modułu. Komponenty można łączyć ze sobą tworząc przepływ danych. Połączenie to nazywane jest sygnałem. Ponadto każdy komponent może definiować parametry konfiguracyjne. Dzięki oprogramowaniu HAL istnieje możliwość swobodnej konfiguracji maszyny, ponieważ oprogramowanie LinuxCNC zależne jest od

abstrakcyjnego interfejsu warstwy HAL. W ten sposób LinuxCNC nie odczytuje sygnałów bezpośrednio z fizycznych układów wejść/wyjść, lecz odwołuje się do sygnałów zdefiniowanych w warstwie HAL. Od użytkownika zależy jakie będzie źródło danych wejściowych i dokąd będą trafiać dane wyjściowe komponentów.

Oprogramowanie HAL zawiera gotową bibliotekę komponentów. Oprócz tego istnieje możliwość tworzenia własnych komponentów. Do oprogramowania HAL zainstalowano otwarto-źródłowy komponent linuxcnc-ethercat¹ (moduł o nazwie lcec) umożliwiający wymianę danych z magistralą EtherCAT. Zainstalowano także komponent hal-cia402² do obsługi napędów Nanotec C5-E zgodnych ze standardem CiA 402.

Na komputerze zainstalowano oprogramowanie LinuxCNC w wersji 2.8.2-11. W pierwszej kolejności wykonano test rozrzutów czasowych cyklu (jitter) wątku servo programu LinuxCNC (Rysunek 29). Wątek ten odpowiada za cykliczną realizację zadań czasu rzeczywistego, w szczególności jądra numerycznego oraz komunikacji EtherCAT.



Rysunek 29. Histogram opóźnień wątku serwo oprogramowania LinuxCNC

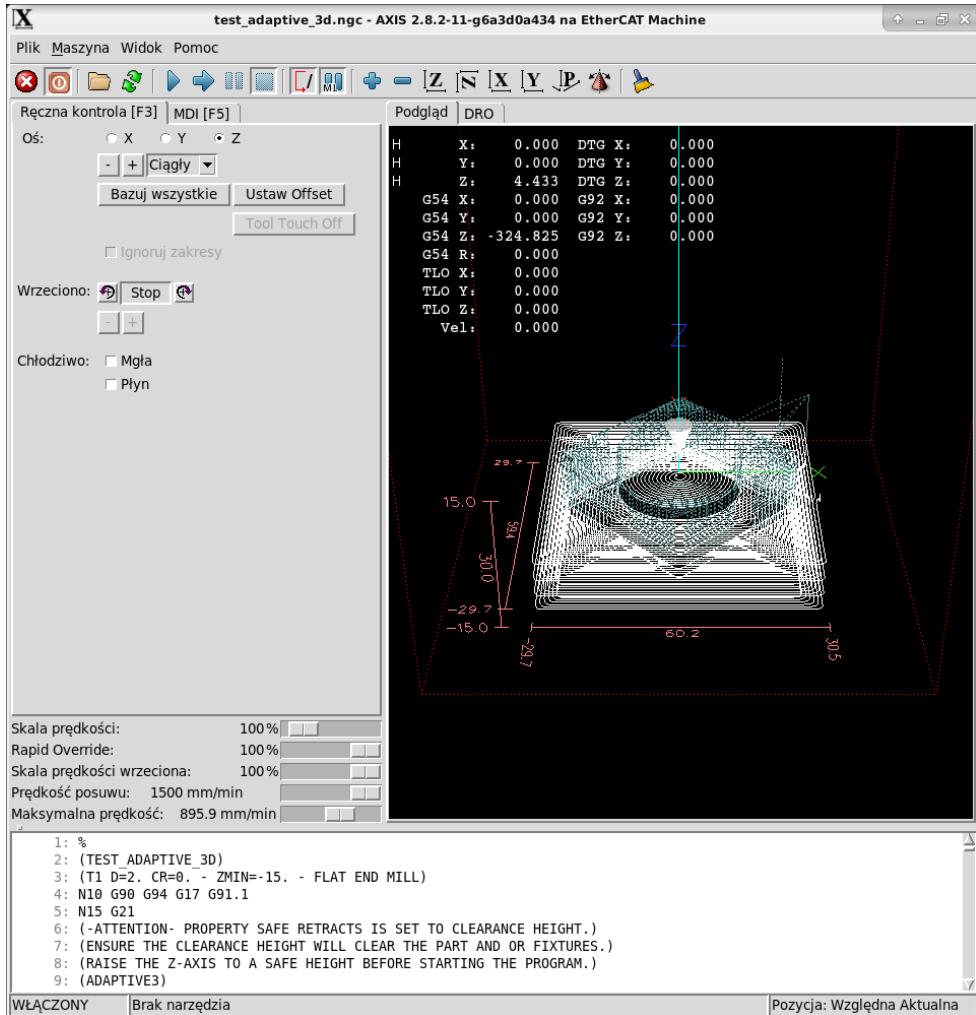
Test należy wykonywać obciążając procesor dodatkowymi zadaniami. W tym celu w trakcie testu uruchomiono kilka instancji programu demonstracyjnego grafiki 3D o nazwie Glxgears.

¹ <https://github.com/sittner/linuxcnc-ethercat> [24.01.2024]

² <https://github.com/dbraun1981/hal-cia402> [24.01.2024]

Większość próbek zawierała się w przedziale +/- 20μs. Maksymalne fluktuacje czasu cyklu zawierały się poniżej wartości +/- 400μs. Największe odchyłki miały miejsce przy inicjalizacji wątku. Jest to akceptowalny wynik przy czasie cyklu komunikacyjnego wynoszącym 1ms. W trakcie działania układu sterowania nie stwierdzono błędów napędów wynikających z występowania nadmiernej odchyłki od zadanego czasu cyklu magistrali EtherCAT.

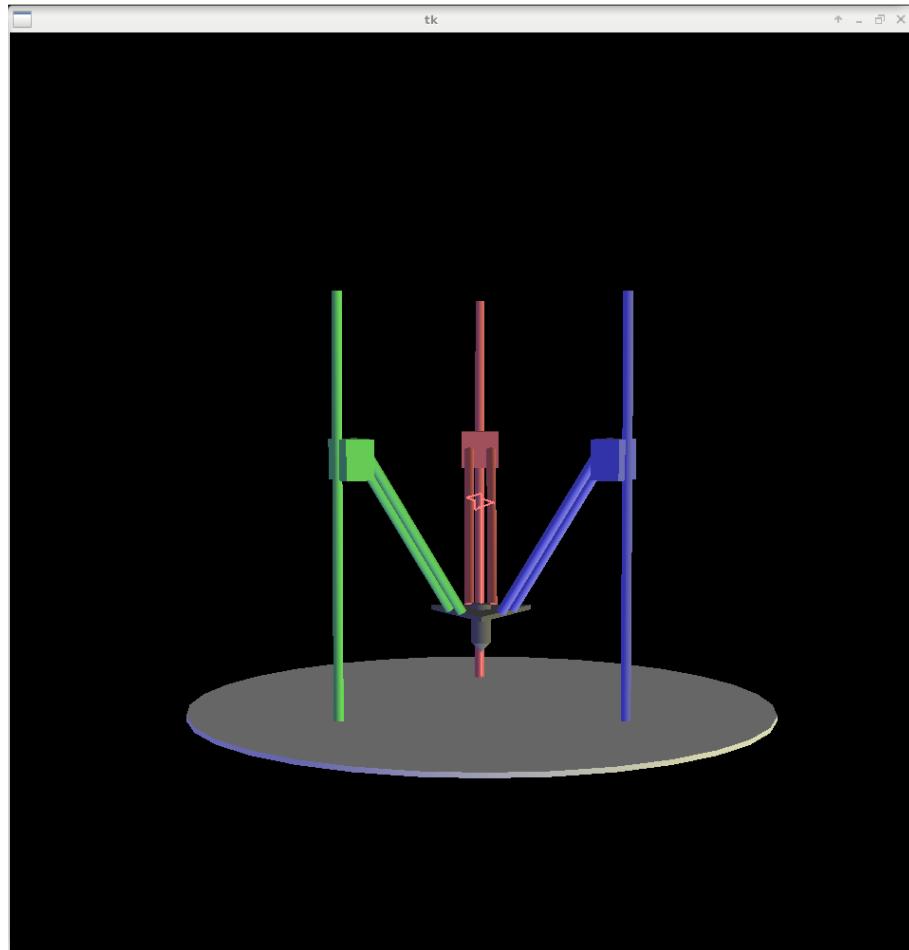
Jako interfejs graficzny wykorzystano domyślny interfejs AXIS przeznaczony do sterowania z klawiatury:



Rysunek 30. Interfejs graficzny AXIS oprogramowania LinuxCNC

U góry znajduje się panel z przyciskami: wyłącznik awaryjny, start, pauza oraz zatrzymanie programu technologicznego i inne. Z lewej strony okna znajduje się zakładka ręcznej kontroli umożliwiająca sterowanie osiami maszyny i wrzecionem za pomocą klawiatury bądź przycisków. Z prawej strony jest podgląd 3D aktualnie wykonywanej ścieżki programu technologicznego. Z dołu wyświetlany jest kod programu technologicznego z oznaczeniem obecnie wykonywanego wiersza.

Oprócz tego zastosowano moduł ***lineardelta*** pełniący rolę interfejsu GUI maszyny w układzie delta liniowa:

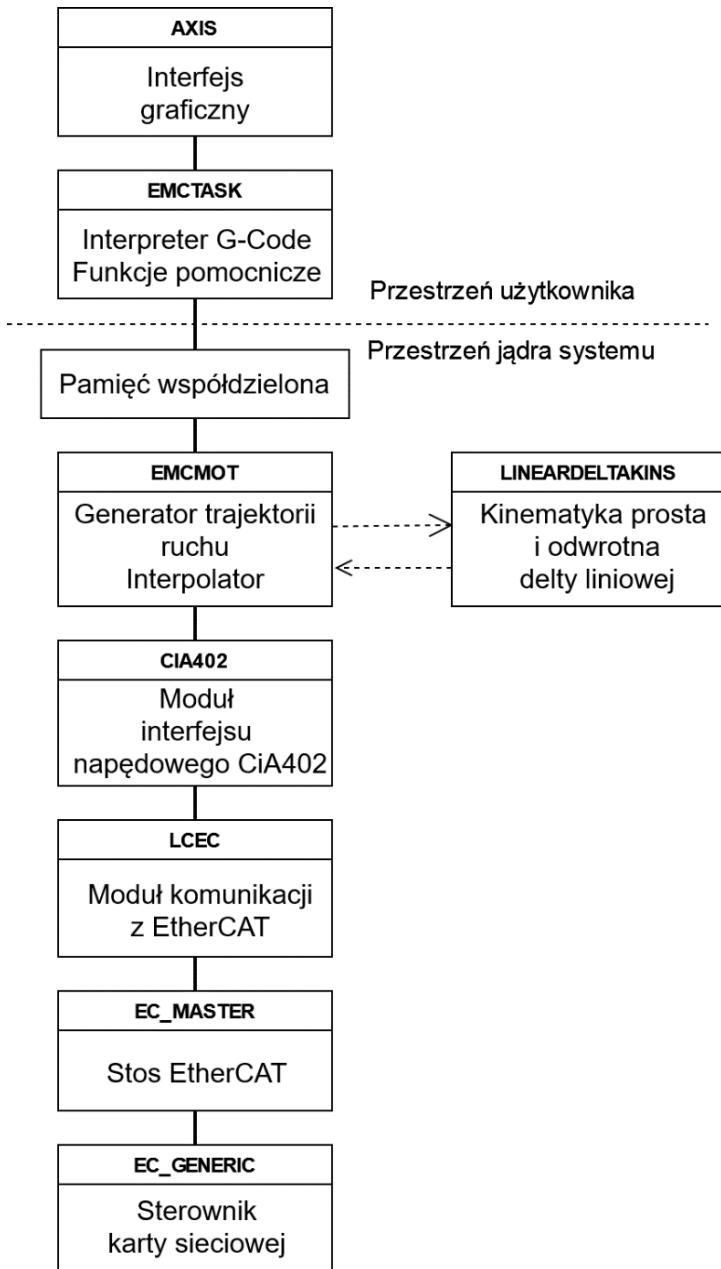


Rysunek 31. Interfejs graficzny modulu lineardelta

Interfejs ten wyświetla symulację aktualnego położenia złącz i efektora maszyny.

Oprogramowanie LinuxCNC składa się z części wykonywanej na poziomie użytkownika: interfejsu graficznego, interpretera G-Code, funkcji pomocniczych; a także z części wykonywanej na poziomie jądra systemu: modułu generatora trajektorii i interpolatora korzystającego z funkcji modułu kinematyki delty liniowej, modułu implementującego interfejs napędowy CiA402 oraz modułu ***lcec*** służącego do komunikacji z oprogramowaniem IgH EtherCAT Master, na które składają się dwa moduły jądra systemu operacyjnego: stos mastera EtherCAT (moduł ***ec_master***) oraz sterownik karty sieciowej (moduł ***ec_generic***). Sterownik „generic” w rzeczywistości wykorzystuje standardowy sterownik systemu Linux (w tym przypadku Intel e1000e). Dzięki temu nie ma konieczności dostosowywania dedykowanego sterownika dla jądra czasu rzeczywistego dla konkretnej karty sieciowej i konkretnej wersji jądra.

Na diagramie poniżej przedstawiono opisywaną strukturę oprogramowania sterującego:



Rysunek 32. Struktura komponentów oprogramowania sterującego

Pojedyncza linia oznacza komunikację za pomocą pin'ów HAL'a. Przerywane strzałki oznaczają komunikację poprzez wywołania funkcji. Dane z interpretera G-Code pracującego w przestrzeni użytkownika są wymieniane poprzez pamięć współdzieloną z komponentem działającym w jądrze systemu **emcmot** służącego do generacji trajektorii ruchu i interpolacji. Po wykonaniu interpolacji pozycje we współrzędnych kartezjańskich przeliczane są na pozycje złącz za pomocą modułu kinematyki **lineardeltakins**. Dalej dane poprzez moduł **cia402** w formie kompatybilnej z interfejsem CiA402 trafiają do modułu **lcec**. Następnie poprzez stos mastera EtherCAT'a dane trafiają do sterownika karty sieciowej skąd są wysyłane do napędów.

4.2. Konfiguracja oprogramowania sterującego

Oprogramowanie LinuxCNC konfiguruje się za pomocą plików tekstowych. Katalog konfiguracyjny projektu składa się z:

- ethercat-conf.xml – plik mapujący dane procesowe magistrali EtherCAT z pinami interfejsu HAL
- umk_delta.ini – główny plik definiujący parametry maszyny
- umk_delta.hal – plik z konfiguracją i połączeniami komponentów HAL
- umk_delta.tbl – definicje narzędzi

oraz kilku plików pomocniczych.

Poniżej w tabeli wymieniono najważniejsze parametry głównego pliku projektowego:

Tabela 1. Skonfigurowane parametry w głównym pliku projektu INI

Sekcja	Parametr	Opis	Wartość
DISPLAY	DEFAULT_LINEAR_VELOCITY	Domyślna prędkość jog	15.0 [mm/s]
	MAX_LINEAR_VELOCITY	Maksymalna prędkość jog	25.0 [mm/s]
	POSITION_OFFSET	Domyślny układ współrzędnych w GUI	RELATIVE
	POSITION_FEEDBACK	Domyślne wartości współrzędnych	ACTUAL
EMCMOT	SERVO_PERIOD	Okres wątku „serwo”	1000000 ns
	TRAJ_PERIOD	Okres wątku generatora trajektorii ruchu	1000000 ns
TASK	CYCLE_TIME	Okres wątku „milltask” interpretującego G-Code	0.01 s
TRAJ	COORDINATES	Nazwy współrzędnych maszyny	XYZ
	AXES	Ilość osi	3
	JOINTS	Ilość złącz	3
	DEFAULT_LINEAR_VELOCITY	Domyślna prędkość osi	15 [mm/s]
	MAX_LINEAR_VELOCITY	Maksymalna prędkość osi	25 [mm/s]
KINS	KINEMATICS	Wybrany komponent kinematyki	lineardeltakins
AXIS_X	MIN_LIMIT	Programowy limit dolny osi X	-50 [mm]
	MAX_LIMIT	Programowy limit górny osi X	50 [mm]
	MAX_VELOCITY	Maksymalna prędkość osi X	15 [mm/s]
	MAX_ACCELERATION	Maksymalne przyspieszenie osi X	100 [mm/s ²]

AXIS_Y	MIN_LIMIT	Programowy limit dolny osi Y	-50 [mm]
	MAX_LIMIT	Programowy limit górny osi Y	50 [mm]
	MAX_VELOCITY	Maksymalna prędkość osi Y	15 [mm/s]
	MAX_ACCELERATION	Maksymalne przyspieszenie osi Y	100 [mm/s ²]
AXIS_Z	MIN_LIMIT	Programowy limit dolny osi Z	-345 [mm]
	MAX_LIMIT	Programowy limit górny osi Z	-125 [mm]
	MAX_VELOCITY	Maksymalna prędkość osi Z	15 [mm/s]
	MAX_ACCELERATION	Maksymalne przyspieszenie osi Z	100 [mm/s ²]
JOINT_0 JOINT_1 JOINT_2	MIN_LIMIT	Programowy limit dolny złącz A,B,C	-280 [mm]
	MAX_LIMIT	Programowy limit górny złącz A,B,C	25 [mm]
	MAX_VELOCITY	Maksymalna prędkość złącz A,B,C	15 [mm/s]
	MAX_ACCELERATION	Maksymalne przyspieszenie złącz A,B,C	100 [mm/s ²]
	HOME_SEQUENCE	Numer sekwencji bazowania złącz A,B,C	0

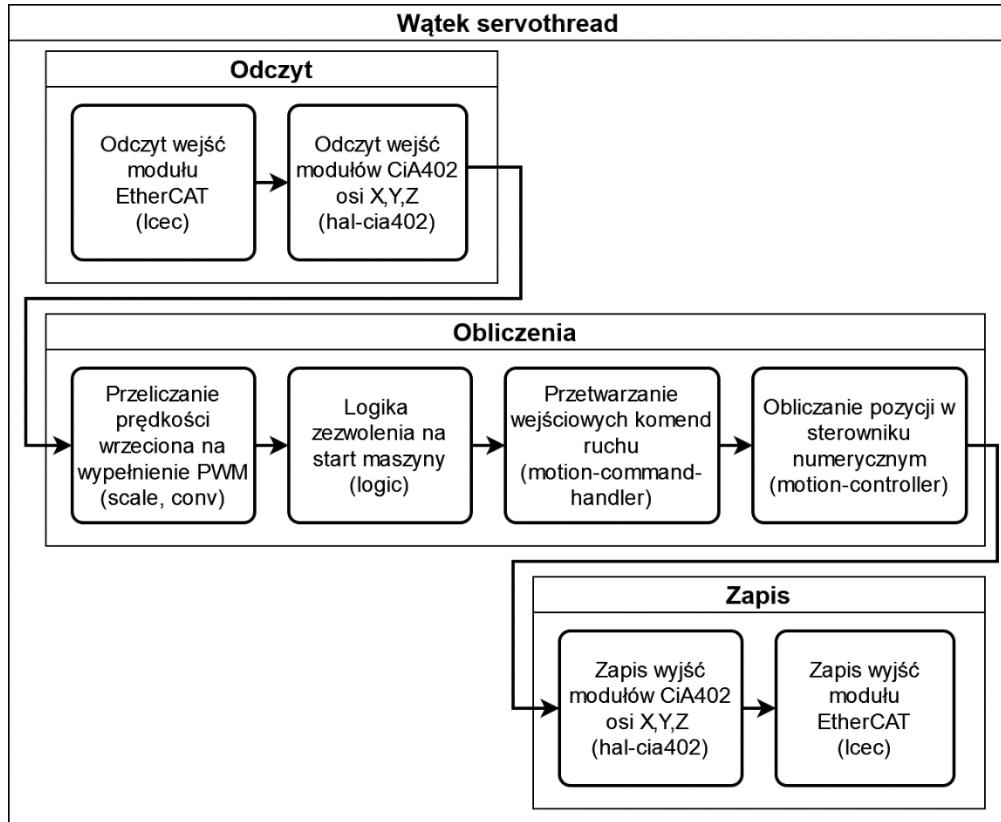
W programie zdefiniowano 3 osie (ang. axis) maszyny kartezjańskiego układu współrzędnych X, Y, Z do których odwołuje się program technologiczny. Złącza (ang. joint) są fizycznymi stopniami swobody maszyny, w tym przypadku są to złącza liniowe na prowadnicach kolumn nazwane A, B, C. Pozycja osi jest obliczana z pozycji złącz za pomocą komponentu kinematyki.

Najprostszym rodzajem kinematyki jest kinematyka kartezjańska w której każda oś posiada jedno złącze jako oś kartezjańskiego układu współrzędnych. Zastosowana kinematyka typu delta liniowa oblicza jaka będzie pozycja wynikowa efektora w układzie kartezjańskim przy danym ustawieniu złączy liniowych kolumn. Kinematyka odwrotna oblicza konieczne położenia złącz, aby osiągnąć zadaną pozycję efektora.

Na plik konfiguracyjny HAL składa się:

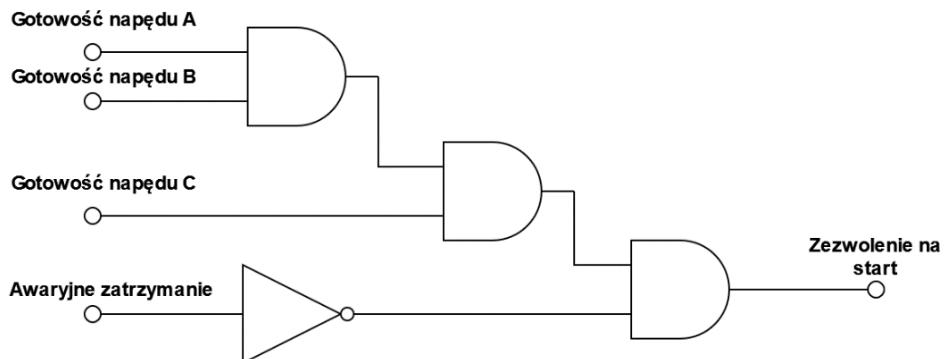
- ładowanie komponentów za pomocą polecenia **loadrt** w przypadku gdy komponent ma się wykonywać w wątku jądra systemowego lub polecenia **loadusr** dla komponentów w przestrzeni użytkownika (ang. userspace)
- linkowanie funkcji do wątków, których kolejność wykonywania zdeterminowana jest przez ich pozycję w pliku
- definicje połączeń sygnałów pomiędzy pinami komponentów

W konfiguracji HAL wykorzystano tylko jeden wątek „servothread” który wykonuje funkcje w kolejności przedstawionej poniżej:



Rysunek 33. Kolejność wykonywania funkcji w wątku servothread

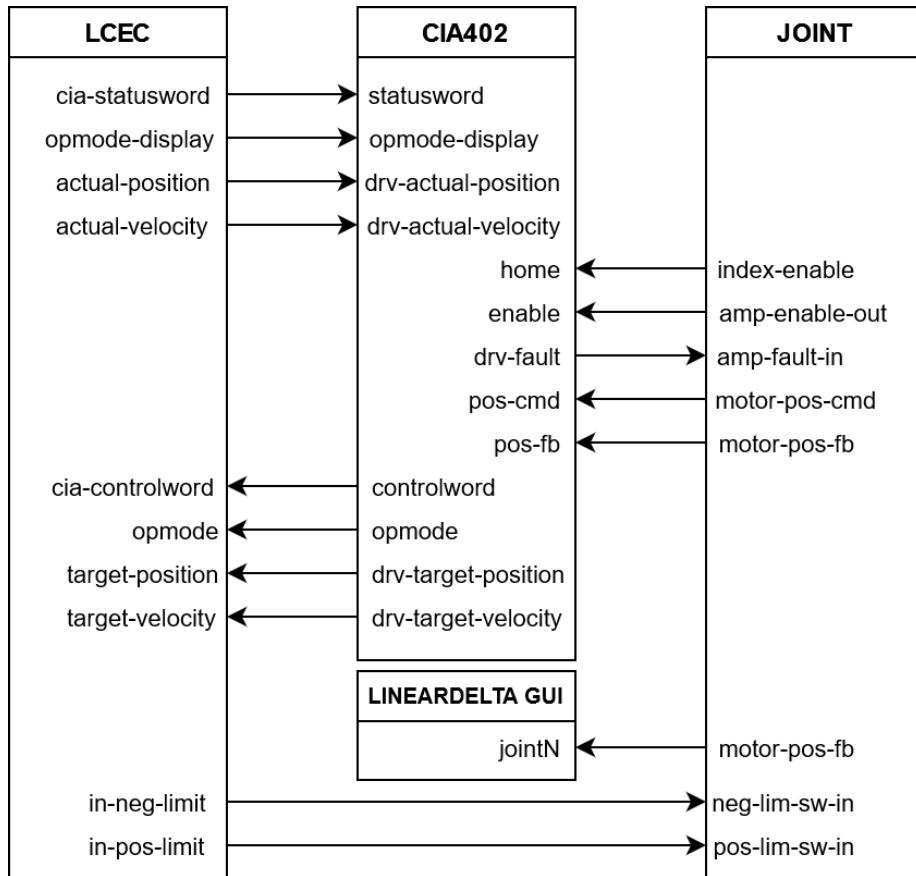
W pliku konfiguracji komponentów HAL napisano warunki konieczne do załączenia maszyny. Poniżej przedstawiono schemat ideowy działania:



Rysunek 34. Logika zezwolenia na start maszyny

Wykorzystano komponent HAL o nazwie LOGIC aby stworzyć programową bramkę AND. Do uruchomienia maszyny konieczna jest gotowość napędów (stan gotowości maszyny stanu CiA402), wyłączony programowy przycisk awaryjnego zatrzymania oraz załączony główny włącznik programowy maszyny.

Poniżej przedstawiono szczegółowy diagram połączeń pomiędzy pinami modułów *lcec*, *cia402*, *joint* (motion) oraz modułu *lineardelta*:



Rysunek 35. Diagram sygnałów HAL pomiędzy *lcec*, *cia402* oraz *joint* (motion)

Powyższe połączenia realizują następującą funkcjonalność:

- obsługa interfejsu CiA402 napędów w trybie CSP (Cyclic Synchronous Position)
- bazowanie osi w trybie Homing za pomocą interfejsu CiA402
- obsługa wyłączników krańcowych osi
- przekazywanie aktualnej pozycji złącza do modułu GUI *lineardelta*

4.3. Konfiguracja oprogramowania serwonapędów

Napędy serwo-krokowe Nanotec C5E-1-11 można skonfigurować za pomocą programu Plug & Drive Studio. Komunikacja programu z napędami odbywa się poprzez interfejs USB.

Z uwagi na symetryczność konstrukcji każdy napęd osi skonfigurowano jednakowo. W poniższej tabeli przedstawiono parametry, których wartość zmodyfikowano w stosunku do nastaw fabrycznych. Pominięto parametry regulatorów, których wartości ustalono podczas badań:

Tabela 2. Parametry serwonapędów Nanotec C5-E

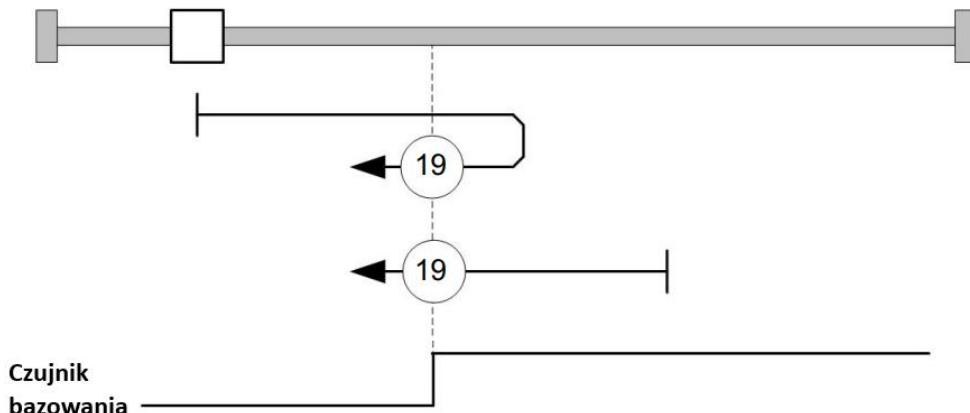
Adres parametru	Parametr	Opis	Wartość
2030h:00h	Pole Pair Count	Ilość par biegunów silnika	50
2031h:00h	Max Motor Current	Maksymalny prąd silnika	3000 mA
203Bh:01h	Motor Rated Current	Znamionowy prąd silnika	3000 mA
3202h:00h	Motor Drive Submode Select	Tryb pracy sterownika. Ustawiono pracę w pętli zamkniętej	1
3203h:03h	Feedback Selection – 3rd Feedback Interface	Enkoder wybrany jako sprzężenie zwrotne regulatora momentu, prędkości i położenia	0111b
3240h:01h	Digital Inputs Control – Special Function Enable	Załączanie wejść cyfrowych specjalnego przeznaczenia. Bit 0 – krańcówka dolna, bit 1 – krańcówka górna, bit 2 – czujnik bazowania, bit 3 – e-stop	1111b
3240h:06h	Digital Inputs Control – Input Range Select	Wybór napięcia wejść cyfrowych. Bit 0 – 1 wejście, bit 1 – 2 wejście itd. Wartość „0” – 5V; wartość „1” – 24V	1111b
3240h:08h	Digital Inputs Control – Routing Enable	Załączanie routing'u wejść cyfrowych	1
3242h:01h	Digital Input Routing – Input Source For Bit #1	Nr wejścia, którego wartość znajdzie się w 1. bicie obiektu 60FDh	1
3242h:02h	Digital Input Routing – Input Source For Bit #2	Nr wejścia, którego wartość znajdzie się w 2. bicie obiektu 60FDh	2
3242h:03h	Digital Input Routing – Input Source For Bit #3	Nr wejścia, którego wartość znajdzie się w 3. bicie obiektu 60FDh	3
3242h:04h	Digital Input Routing – Input Source For Bit #4	Nr wejścia, którego wartość znajdzie się w 4. bicie obiektu 60FDh	4
605Ah:00h	Quick Stop Option Code	Wybór reakcji na stan Quick Stop	2
6080h:00h	Max Speed	Maksymalna prędkość przemieszczenia	20 mm/s
6083h:00h	Profile Acceleration	Maksymalne przyspieszenie	500 mm/s ²
6084h:00h	Profile Deceleration	Maksymalne zwalnianie	500 mm/s ²
608Fh:01h	Position Encoder Resolution	Rozdzielcość enkodera dla regulatora położenia	16000 ink/obr
6090h:01h	Velocity Encoder Resolution	Rozdzielcość enkodera dla regulatora prędkości	16000 ink/obr
6092h:01h	Feed Constant	Stała posuwu	31.416 mm/obr
6098h:01h	Homing Method	Metoda bazowania	19
6099h:01h	Homing Speed During Search For Switch	Prędkość szukania czujnika bazowania	20 mm/s
6099h:02h	Homing Speed During Search For Zero	Prędkość zjazdu z czujnika bazowania	2 mm/s
60A8h:00h	SI Unit Position	Jednostka położenia. Bity 16-23 – jednostka; bity 24-31 – eksponent potęgi 10	FA010000h
60A9h:00h	SI Unit Velocity	Jednostka prędkości. Bity 8-15 – jednostka czasu; bity 16-23 – jednostka położenia; bity 24-31 – eksponent potęgi 10	FD010300h

Pracę zrealizowano dla firmware'u w wersji FIR-v1926. W najnowszym, w chwili składania pracy firmware FIR-v2213, zmianie uległy adresy parametrów, co zostało opisane w najnowszej wersji instrukcji, lecz stare adresy są nadal kompatybilne.

Wpisano parametry silnika (prąd, ilość biegunów) oraz enkodera (rozdzielcość) do ustawień serwonapędu. Wybrano enkoder jako sprzężenie zwrotne dla wszystkich regulatorów w kaskadzie regulacji (parametr 3203:03). Zakres wejść cyfrowych ustawiono na 24V oraz włączono wejścia specjalnego przeznaczenia: wyłączniki krańcowe, e-stop, czujniki bazowania. Wartości z wejść przekierowano do obiektu 60FD za pomocą wbudowanej możliwości przekierowywania (routing'u). Jako reakcję na stan awaryjnego zatrzymania (Quick stop) wybrano hamowanie z szybką rampą i następnie zmianę stanu na Switch on disabled. Jako jednostkę położenia ustawiono mikrometr, zaś jako jednostkę prędkości ustawiono milimetry na sekundę. Wpisano ograniczenia maksymalnej prędkości i przyspieszenia silnika. Stałą posuwu przeliczającą położenie kątowe wału silnika na przesunięcie liniowe paska na kole zębatym o średnicy 10mm obliczono ze wzoru na obwód koła:

$$sp = 2\pi r = \pi d = \pi * 10\text{mm} \approx 31.416\text{ mm}$$

Serwonapęd Nanotec C5-E posiada wiele metod bazowania. Wybrano metodę polegającą na najazd na czujnik bazowania przy prędkości szukania (20mm/s), a następnie powolny zjazd z czujnika przy prędkości zjazdowej (2mm/s).

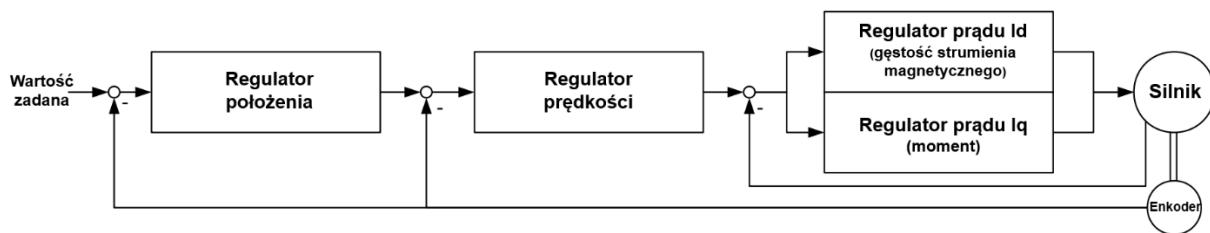


Rysunek 36. Metoda bazowania serwonapędu osi Nanotec C5-E. Źródło: [24]

5. Badania

5.1. Strojenie regulatorów serwonapędów osi

W tej części zestrojono regulatory serwonapędów Nanotec C5-E i zbadano jakość regulacji. Napędy zostały ustawione w tryb pracy w pętli zamkniętej regulacji. Zgodnie z instrukcją napędu strojenie wykonywano na docelowym obciążeniu silnika. Regulacja położenia odbywa się poprzez kaskadową strukturę regulacji położenia kątowego. W napędzie znajdują się regulatory PI prądu, prędkości oraz położenia. Regulator prądu składa się z dwóch członów: regulatora momentu i regulatora strumienia.



Rysunek 37. Struktura regulacji w serwonapędach Nanotec C5-E. Źródło: [24]

Skorzystano z wbudowanej funkcjonalności tuning'u w oprogramowaniu Plug & Drive Studio. Rozpoczęto od strojenia regulatora prądu. Uruchomiono oscyloskop w programie Plug & Drive Studio. Domyślne nastawy wartości regulatorów dla silników krokowych z serii NEMA 17 przedstawiono w tabeli poniżej:

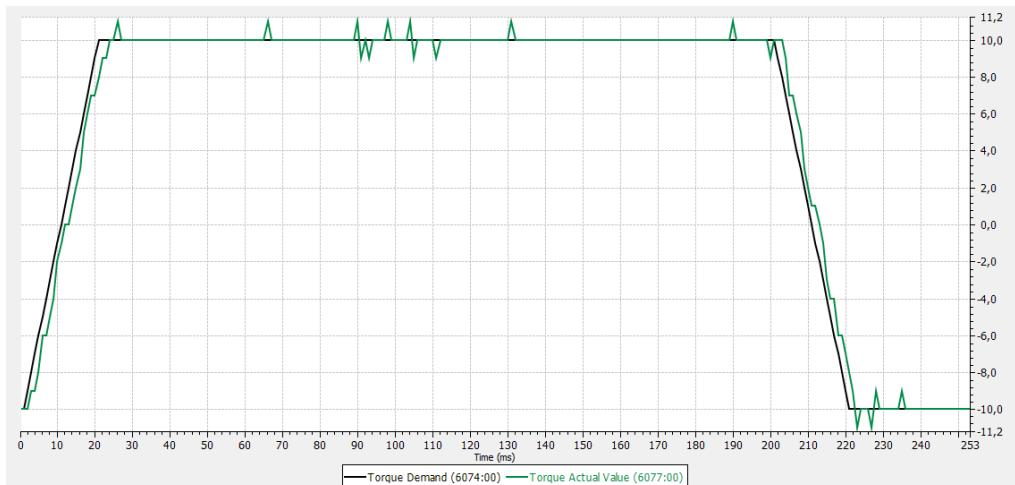
Tabela 3. Domyślne nastawy regulatorów dla silników krokowych serii NEMA 17

Adres parametru	Parametr	Opis	Wartość
3210h:01h	Position Loop, Proportional Gain	Wzmocnienie proporcjonalne pętli położenia	6K
3210h:02h	Position Loop, Integral Gain	Wzmocnienie całkowania pętli położenia	0
3210h:03h	Velocity Loop, Proportional Gain	Wzmocnienie proporcjonalne pętli prędkości	10K
3210h:04h	Velocity Loop, Integral Gain	Wzmocnienie całkowania pętli prędkości	50
3210h:05h	Flux Current Loop, Proportional Gain	Wzmocnienie proporcjonalne strumienia pętli prądu	2M
3210h:06h	Flux Current Loop, Integral Gain	Wzmocnienie całkowania strumienia pętli prądu	10K
3210h:07h	Torque Current Loop, Proportional Gain	Wzmocnienie proporcjonalne momentu pętli prądu	2M
3210h:08h	Torque Current Loop, Integral Gain	Wzmocnienie całkowania momentu pętli prądu	10K
3210h:0Bh	Velocity Feed Forward Factor	Wartość sprzężenia w przód od prędkości wyrażona w promilach	1000%
3210h:0Ch	Acceleration Feed Forward Factor	Wartość sprzężenia w przód od przyspieszenia wyrażona w promilach	0%

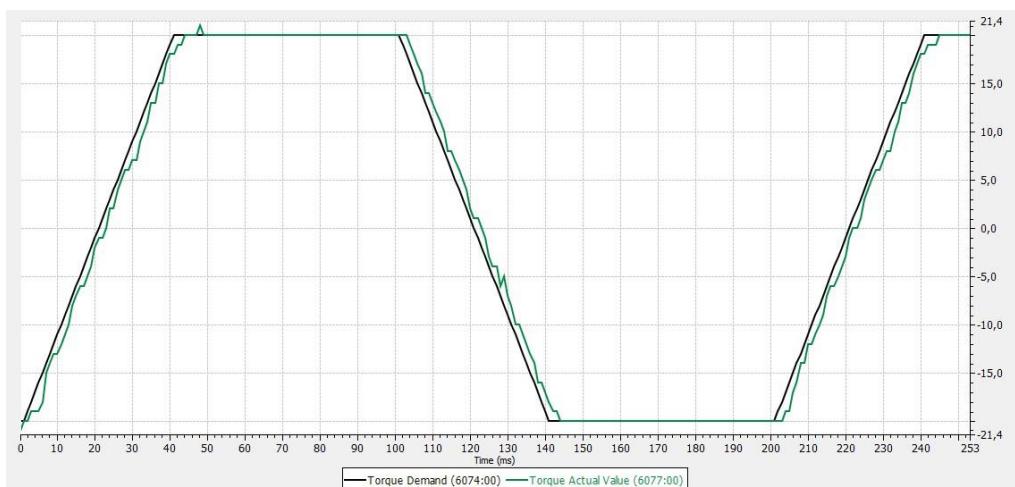
Podczas strojenia regulatorów prądu napęd jest w trybie sterowania momentem (Profile Torque). Ustalono maksymalny moment (max torque) równy 100% wartości nominalnej oraz nachylenie momentu (torque slope) równe maksymalnej zmianie momentu o 100% na sekundę. Strojenie rozpoczęto od zerowych wartości obu regulatorów prądu - momentu i strumienia. Wartości wzmacnianie zmieniano dla obu regulatorów o taką samą wartość. Wstępnie wzmacnianie proporcjonalne zwiększano aż do momentu w którym napęd zaczął wyraźnie hałasować ($P \approx 280000$). Następnie podzielono tą wartość przez dwa i zmniejszano aż do momentu uzyskania łagodnej pracy napędu. Następnie dodano człon całkujący w celu zniwelowania uchybu ustalonego. Zaproponowane parametry dla regulatorów prądu:

- wzmacnianie proporcjonalne momentu i strumienia pętli prądu $P = 120000$
- wzmacnianie całkowania momentu i strumienia pętli prądu $I = 5000$

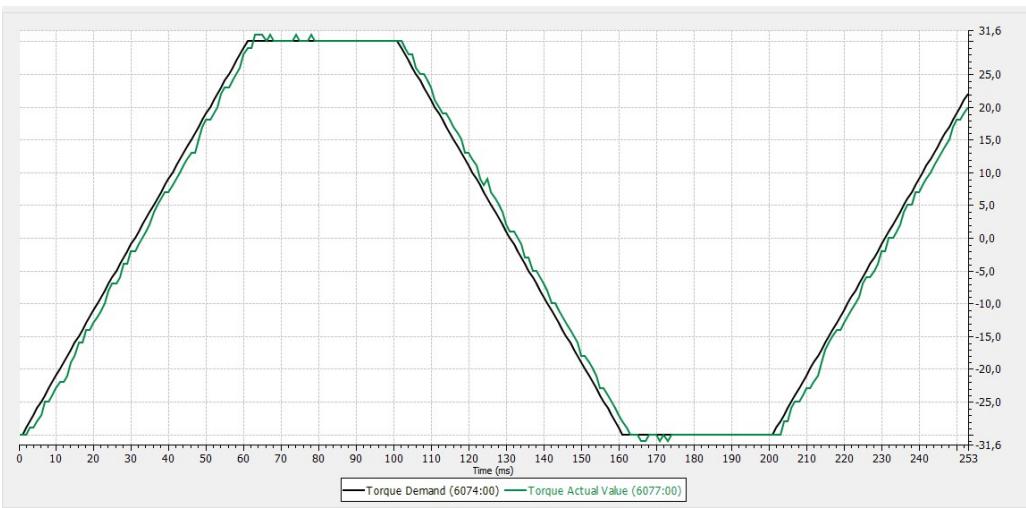
Odpowiedzi dla trapezowego profilu momentu zadanej od ± 10 do $\pm 30\%$ wartości nominalnej:



Rysunek 38. Charakterystyka odpowiedzi trapezowej regulatora prądu dla $T = \pm 10\%$.



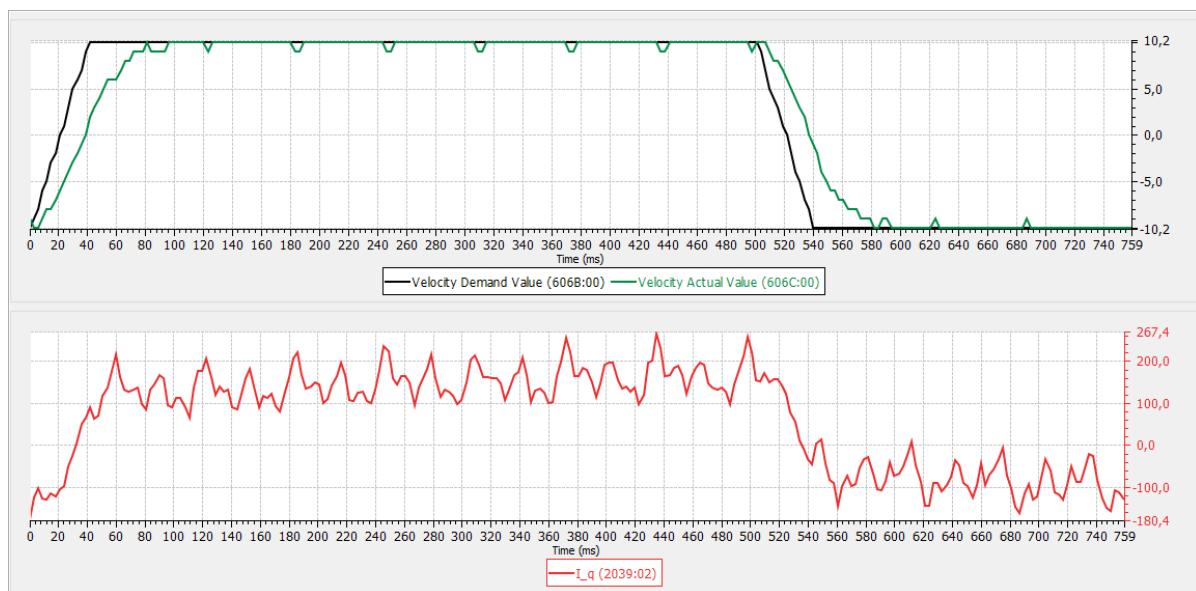
Rysunek 39. Charakterystyka odpowiedzi trapezowej regulatora prądu dla $T = \pm 20\%$



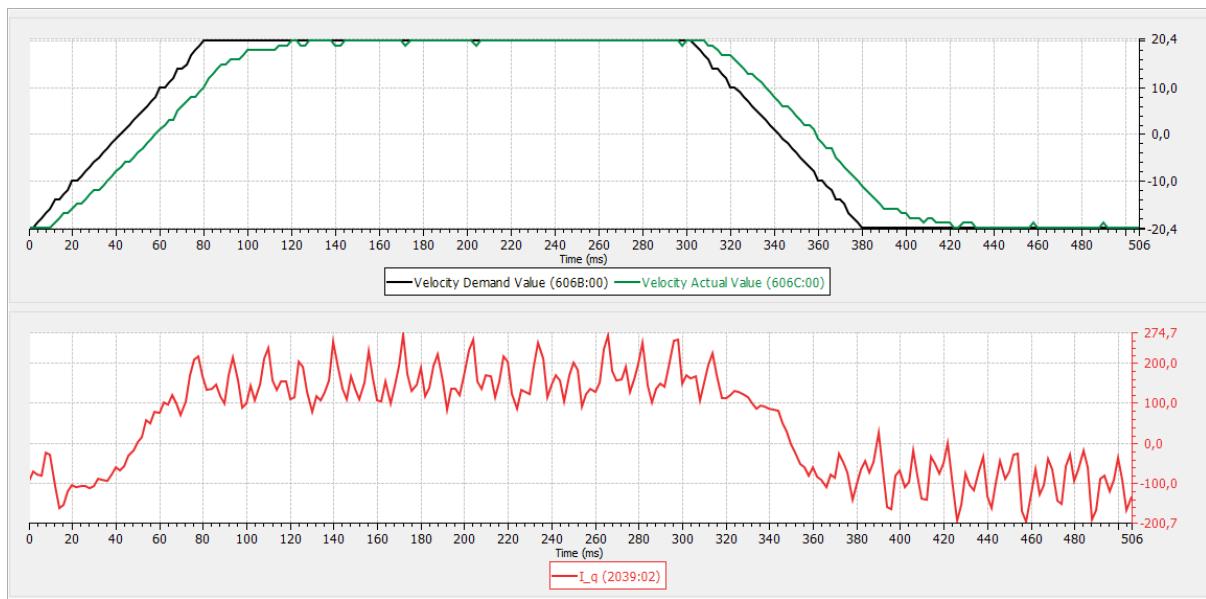
Rysunek 40. Charakterystyka odpowiedzi trapezowej regulatora prądu dla $T = \pm 30\%$

Występowały minimalne tętnienia momentu, których powodem mogły być szумy pomiarowe. Moment osiągał docelową wartość zadaną. Podczas narastania występował niewielki błąd nadążania. Jakość regulacji była zadowalająca.

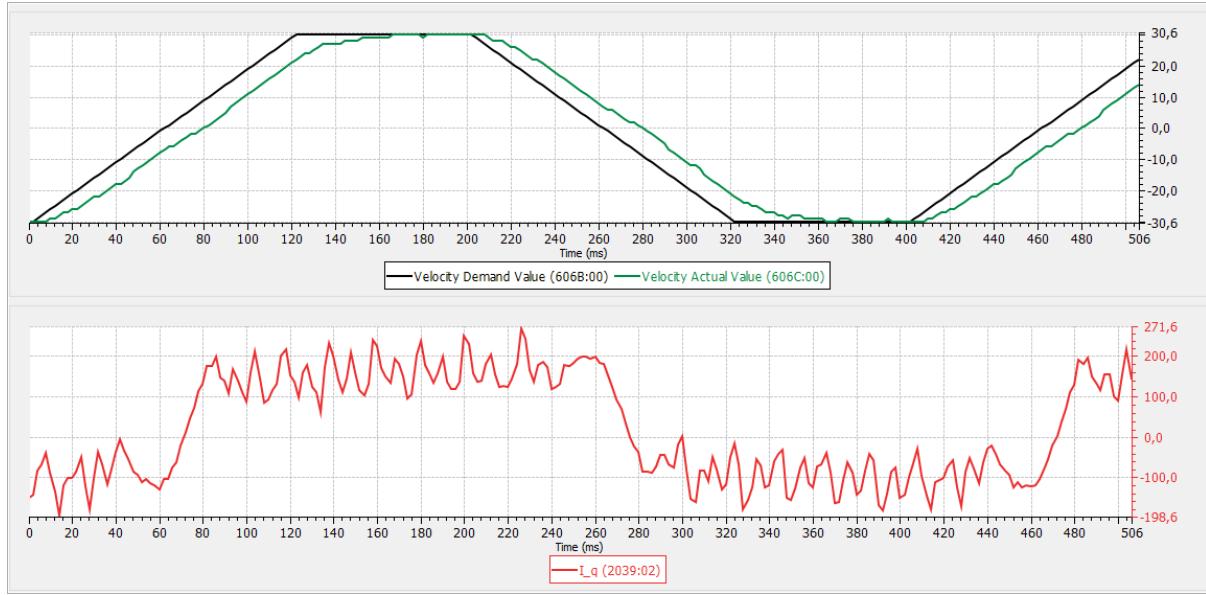
Następnie strojono regulator prędkości przy zachowaniu wcześniej wyznaczonych parametrów regulatora prądu. Przyspieszenie i spowalnianie prędkości pozostawiono na wartości 500 mm/s^2 . Ponownie zaczęto od zerowych nastaw wzmacnienia P oraz I. Napędy strojono dla prędkości zbliżonych do prędkości docelowych dla tej maszyny – od 10 do 30 mm/s. Zbyt duże wartości wzmacnienia proporcjonalnego skutkowały wzbudzaniem się napędu ($P > 28000$). Zbyt małe wartości powodowały drgania mechaniczne ($P < 12000$). Ostatecznie ustalono wzmacnienie proporcjonalne na poziomie $P = 22000$. Dodano człon całkujący o wartości $I = 1600$ w celu zniwelowania uchybu ustalonego.



Rysunek 41. Charakterystyka odpowiedzi trapezowej prędkości dla docelowej $V = 10 \text{ mm/s}$, okres $T = 500\text{ms}$



Rysunek 42. Charakterystyka odpowiedzi trapezowej prędkości dla docelowej $V = 20$ mm/s, okres $T = 300$ ms



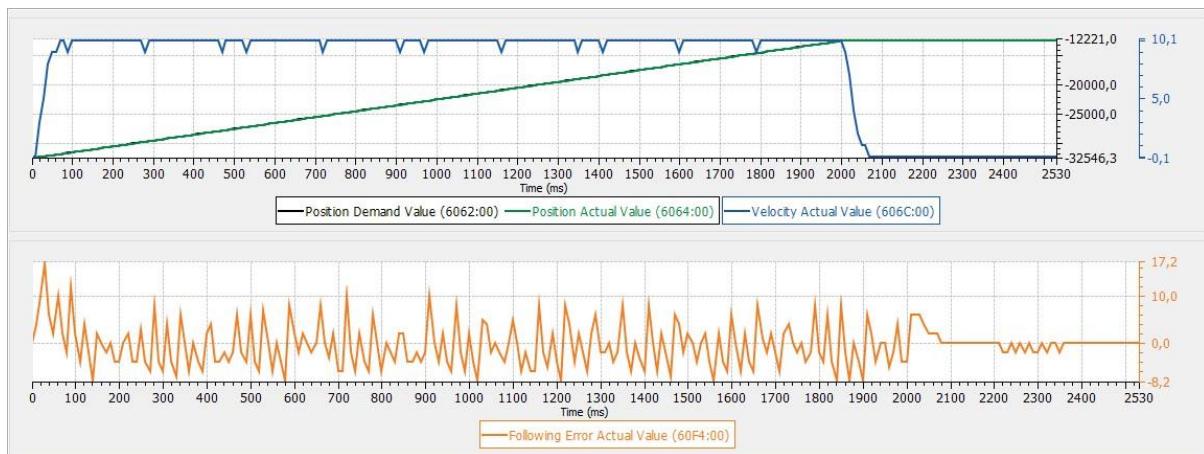
Rysunek 43. Charakterystyka odpowiedzi trapezowej prędkości dla docelowej $V = 30$ mm/s, okres $T = 200$ ms

Widoczne są niewielkie oscylacje prędkości o okresie ok. 60ms wynikające z niedoskonałości konstrukcji mechanicznej. Napęd osiągał zadane prędkości przy zachowaniu trapezowego kształtu profilu bez przeregulowania. Występował uchyb ustalony podczas narastania i opadania wprowadzający opóźnienie ok. 20ms względem trajektorii zadanej. Jest to oczekiwane zachowanie ze względu na fizyczne ograniczenia napędu (maksymalny moment, przyspieszenie i prędkość), a także opóźnienia w torze regulacji i pomiaru.

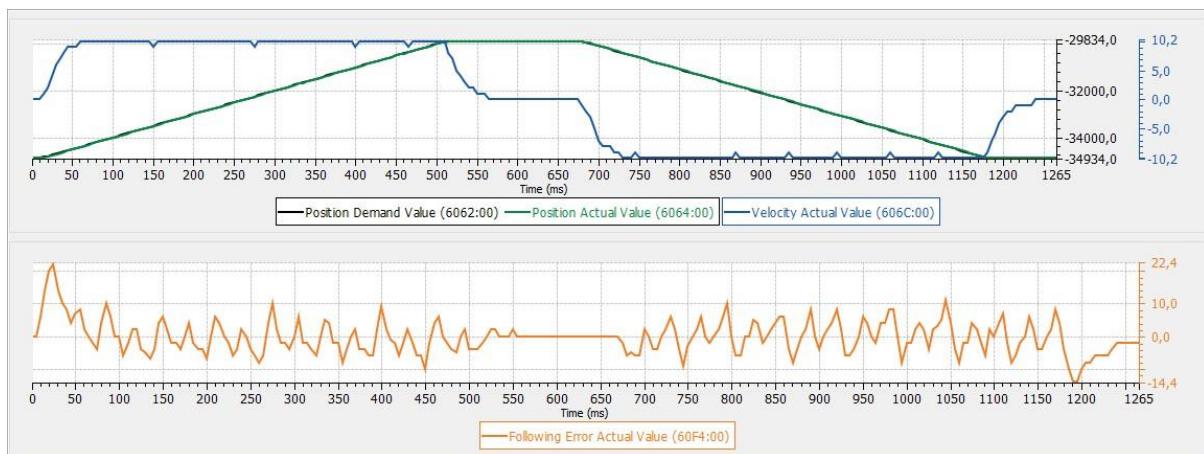
Ostatnim etapem strojenia był dobór nastaw regulatora położenia. Zastosowano sam człon proporcjonalny regulatora PI. W serwonapędach domyślnie aktywne jest sprzężenie w przód od prędkości (ang. velocity feedforward). Domyślna wartość 1000 oznacza, że 100%

wartości prędkości podawane jest jako sprzężenie w przód. Skutkuje to eliminacją błędu położenia w stanie ustalonym i poprawą dynamiki regulatora w stanach przejściowych. Jest to standardowa struktura regulacji dla serwonapędów w maszynach sterowanych numerycznie.

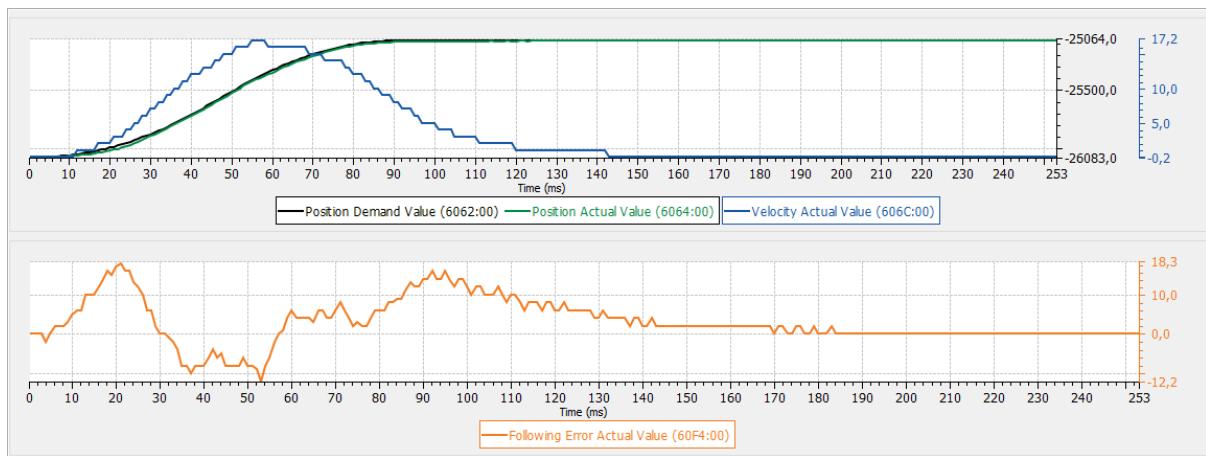
Człon proporcjonalny regulatora położenia zwiększano aż do czasu braku widocznej poprawy w odpowiedzi. Ustalono wartość wzmacnienia proporcjonalnego P na poziomie 1500. Ponownie zebrano odpowiedzi dla trapezowego profilu ruchu przy wartościach przyspieszenia i spowalniania równym 500 mm/s^2 . Zadawano przemieszczenia z zakresu od 1 mm do 20mm przy okresie trwania ruchu od 100 do 2000 ms. Na wykresach poniżej wartości położenia i błędu nadążania podane są w mikrometrach.



Rysunek 44. Charakterystyka trapezowego profilu ruchu dla przemieszczenia +20mm przy $V = 10\text{mm/s}$ i $T = 2000\text{ms}$



Rysunek 45. Charakterystyka trapezowego profilu ruchu dla przemieszczenia $\pm 2.5\text{mm}$ przy $V = 10\text{mm/s}$ i $T = 500\text{ms}$



Rysunek 46. Charakterystyka trapezowego profilu ruchu dla przemieszczenia +1mm przy V = 20mm/s i T = 100ms
 Napęd zmieniał położenie po zadanym profilu. Trapezowy profil ruchu i prędkość po krzywej S przebiegały prawidłowo. Podczas tuning'u zarejestrowano maksymalny błąd nadążania (following error) na poziomie 23 μ m. Ruch narzędziowy był gładki bez widocznego zrywu.

Ostateczne parametry regulatorów podano w tabeli poniżej:

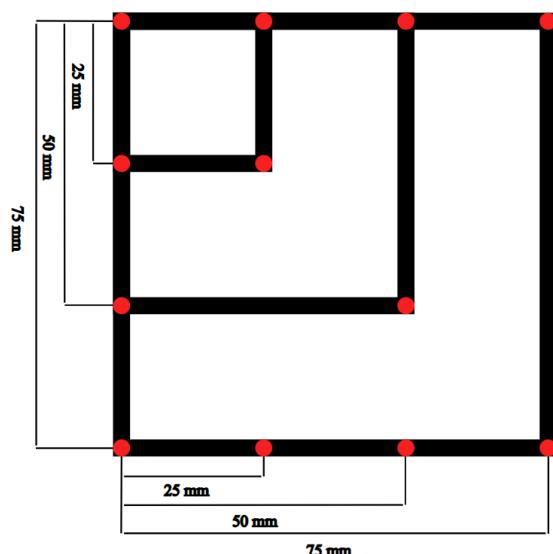
Tabela 4. Zestawienie parametrów układu regulacji w serwonapędach osi po wykonaniu procedury strojenia.

Adres parametru	Parametr	Opis	Wartość
3210h:01h	Position Loop, Proportional Gain	Wzmocnienie proporcjonalne pętli położenia	1500
3210h:02h	Position Loop, Integral Gain	Wzmocnienie całkowania pętli położenia	0
3210h:03h	Velocity Loop, Proportional Gain	Wzmocnienie proporcjonalne pętli prędkości	22000
3210h:04h	Velocity Loop, Integral Gain	Wzmocnienie całkowania pętli prędkości	1600
3210h:05h	Flux Current Loop, Proportional Gain	Wzmocnienie proporcjonalne strumienia pętli prądu	120000
3210h:06h	Flux Current Loop, Integral Gain	Wzmocnienie całkowania strumienia pętli prądu	5000
3210h:07h	Torque Current Loop, Proportional Gain	Wzmocnienie proporcjonalne momentu pętli prądu	120000
3210h:08h	Torque Current Loop, Integral Gain	Wzmocnienie całkowania momentu pętli prądu	5000
3210h:0Bh	Velocity Feed Forward Factor	Wartość sprzężenia w przód od prędkości wyrażona w promilach	1000%
3210h:0Ch	Acceleration Feed Forward Factor	Wartość sprzężenia w przód od przyspieszenia wyrażona w promilach	0%

5.2. Testy jakości obróbki

Maszyna w układzie typu delta liniowa wymaga kalibracji poprzez dobór parametrów kinematyki L (długości ramienia) i R (promienia ramienia) wpływających na przemieszczenia rzeczywiste wrzeciona. Parametr L wpływa na skalowanie odległości pokonywanych w płaszczyźnie XY. Im większe L tym mniejsza rzeczywista odległość. Im mniejsze L – tym większa odległość. Parametr R wpływa na nachylenie płaszczyzny stołu. Im R jest większe tym wrzeciono znajduje się niżej w OXY niż przy kolumnach. Im R mniejsze tym wrzeciono znajduje się wyżej w OXY niż przy kolumnach. Wstępnie wartość parametru L przyjęto na podstawie nominalnej długości ramion podanej przez producenta jako 180mm. Na podstawie zgrubnego pomiaru ustalono że parametr R wynosi 95mm.

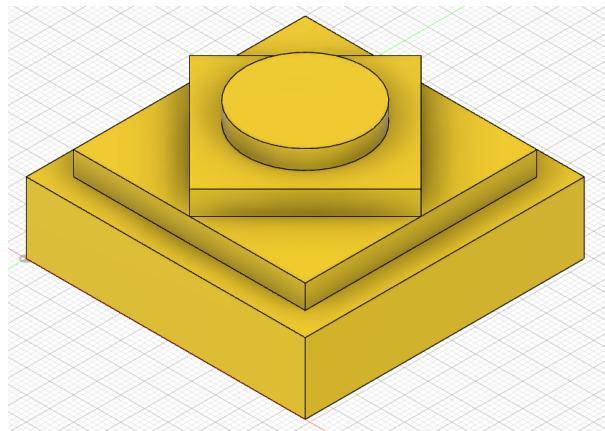
Następnie zaprojektowano w programie GIMP rysunek zawierający kwadraty o znanych odległościach:



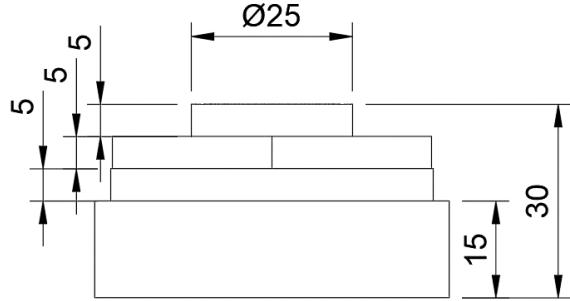
Rysunek 47.Rysunek z kwadratami do kalibracji maszyny

Wydrukowany rysunek przymocowano do stołu i przemieszczano się wrzecionem po czerwonych punktach sprawdzając czy położenia zadane zgadzają się z rzeczywistymi przesunięciami oraz dobierano R tak, aby końcówka narzędzia znajdowała się na stałej wysokości nad płaszczyzną stołu w każdym punkcie dozwolonego zakresu ruchu na płaszczyźnie. Pozwoliło to na wstępna korektę parametru L jako 176 mm i parametru R jako 100 mm.

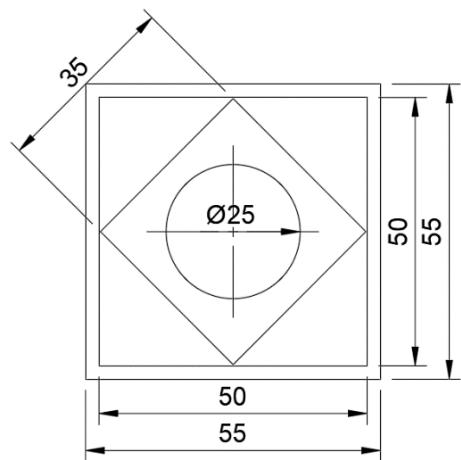
W dalszym etapie należało sprawdzić jakość obróbki materiału. W tym celu zaprojektowano w programie Autodesk Fusion 360 testowy model – kostkę:



Rysunek 48. Testowy model kostki z różnymi kształtami



Rysunek 49. Rzut boczny testowej kostki

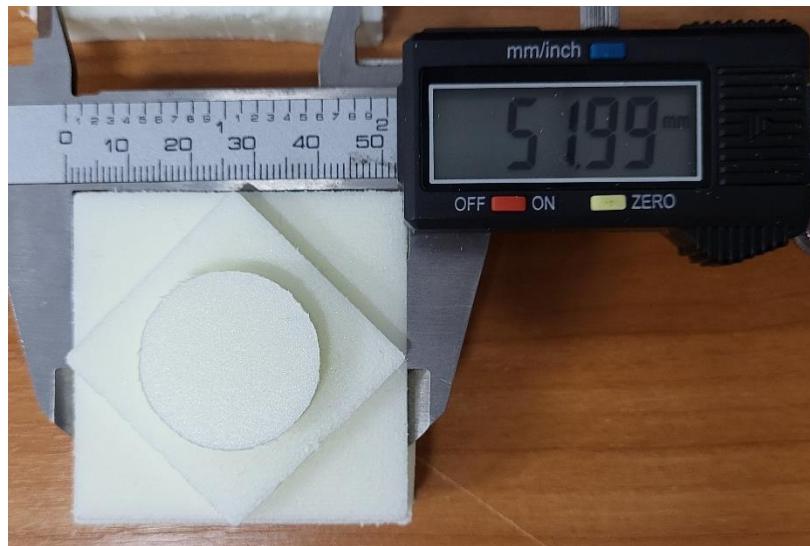


Rysunek 50. Rzut z góry testowej kostki

Wymiary kostki to 55x55x30 mm. Wycięto z prefabrykatu kostkę o takich wymiarach. Zamocowano w uchwycie narzędzia frez płaski 2mm. Czas frezowania wynosił ok. 22 minuty.



Rysunek 51. Wyfrezowana kostka kalibracyjna



Rysunek 52. Wymiar zewnętrzny wyfrezowanej kostki kalibracyjnej

Wszystkie wymiary kostki były przeskalowane o stały współczynnik względem teoretycznych wymiarów. Oznaczało to, że parametr R był dobrany poprawnie. Aby skorygować różnicę wymiarów należało przeskalać wartość L o współczynnik będący stosunkiem zmierzonej długości (52mm) do długości rzeczywistej (50mm):

$$\frac{x_z}{x_r} = \frac{51.99}{50} \approx 1.04 \quad (9)$$

Po przeskalowaniu nowa wartość L wyniosła w przybliżeniu 183mm. Wartość zapisano w konfiguracji i ponownie wyfrezowano ten sam model:



Rysunek 53. Wymiar zewnętrzny drugiej kostki kalibracyjnej



Rysunek 54. Wymiar rombu (35mm) drugiej kostki kalibracyjnej



Rysunek 55. Wymiar koła (25mm) drugiej kostki kalibracyjnej

Wyniki pomiarów przedstawiono w tabeli poniżej:

Tabela 5. Porównanie teoretycznych i rzeczywistych wymiarów figur

Figura	Wymiar teoretyczny	Wymiar zmierzony
Kwadrat	50.00 mm	(50.18 ± 0.02) mm
Romb	35.00 mm	(34.92 ± 0.02) mm
Koło	25.00 mm	(24.96 ± 0.02) mm

Wymiary różniły się od teoretycznych w zakresie $\pm 0.20\text{mm}$. Wykonano kolejne iteracje kalibracji, jednak z uwagi na miękkość materiału oraz niską powtarzalność wynikającą z niedokładności konstrukcji maszyny nie udało się osiągnąć lepszych rezultatów.

Ostatecznie otrzymano następujące wartości parametrów kinematyki:

- Długość ramienia $L = 183\text{mm}$
- Promień ramienia $R = 100\text{mm}$

6. Podsumowanie

Celem pracy było sterowanie maszyną numeryczną w układzie typu Delta liniowa za pomocą komputera stacjonarnego z oprogramowaniem LinuxCNC. Wyposażono maszynę w niezbędne komponenty, wykonano szafę sterowniczą i okablowanie oraz skonfigurowano napędy serwokrokowe osi. W warstwie programowej połączono komponenty HAL do obsługi podzespołów maszyny, w tym zrealizowano komunikację z napędami za pomocą standardu CiA402. Pomyślnie zrealizowano sterowanie numeryczne w oprogramowaniu LinuxCNC z interfejsem komunikacyjnym w formie magistrali EtherCAT.

Wykonana procedura strojenia regulatorów napędów serwokrokowych pozwoliła na osiągnięcie błędu nadążania na poziomie w przybliżeniu $\sim 20\mu\text{m}$ przy docelowych prędkościach pracy maszyny. Zapoznano się z kinematyką maszyny w układzie delta i skalibrowano maszynę poprzez dobór parametrów kinematyki długości ramienia L (183mm) i promienia ramienia R (100mm). Osiągnięto zadowalającą jakość obróbki materiału z dokładnością na poziomie $\pm 0.2\text{mm}$.

Praca pozwoliła na zrozumienie kluczowych w automatyce aspektów inżynierskich, m.in. strojenia regulatorów, sterowania numerycznego oraz przekształceń kinematycznych. Maszyna z powodzeniem może służyć jako stanowisko edukacyjne dla studentów.

Bibliografia

- [1] S.-H. Suh, S. K. Kang, D.-H. Chung i I. Stroud, Theory and Design of CNC Systems, Springer London, 2008.
- [2] P. Radhakrishnan, Computer Numerical Control Machines and Computer Aided Manufacture, New Academic Science, 2015.
- [3] „cadcamlessons,” [Online]. Available: <https://cadcamlessons.com/wp-content/uploads/2019/02/g-code-basics-alphacam-01.png>. [Data uzyskania dostępu: 15 grudzień 2023].
- [4] „LinuxCNC Documentation,” [Online]. Available: <http://linuxcnc.org/docs/2.8/pdf/>. [Data uzyskania dostępu: 15 grudzień 2023].
- [5] Automatykab2b, „Motion control - czyli jak tworzyć systemy serwonapędowe - część 1,” [Online]. Available: <https://automatykab2b.pl/temat-miesiąca/43092-motion-control-czyli-jak-twórzyc-systemy-serwonapedowe-czesc-1>. [Data uzyskania dostępu: 17 grudzień 2023].
- [6] N. K. Mandal, Computer Numerical Controlled Machines: : Constructional Features and Programming, Alpha Science International, 2016.
- [7] FORBOT, „Jak działają silniki krokowe?,” [Online]. Available: <https://forbot.pl/blog/jak-dzialaja-silniki-krokowe-czym-sa-mikrokroki-id7057>. [Data uzyskania dostępu: 17 grudzień 2023].
- [8] K. Erwiński i M. Paprocki, „Synchronization of Electrical Drives via EtherCAT Fieldbus Communication Modules,” *Energies*, nr 2, 2022.
- [9] „Visutech system - EtherCAT in industry,” [Online]. Available: <https://visutechsystem.by/en/ethercat-in-industry/>. [Data uzyskania dostępu: 26 grudzień 2023].
- [10] A. Wawrzak, K. Erwiński, K. Kazimierz i M. Paprocki, „PC based CNC control system with EtherCAT fieldbus,” *Pomiary Automatyka Robotyka*, pp. 29-34, Czerwiec 2016.
- [11] „SINUMERIK 808 - Systemy automatyzacji CNC,” [Online]. Available: <https://www.siemens.com/pl/pl/produkty/automatyka/systemy/cnc-sinumerik/automatyzacja-cnc/sinumerik-808.html>. [Data uzyskania dostępu: 20 grudzień 2023].

- [12] M. Paprocki, K. Erwiński, A. Wawrzak i M. Kłosowiak, „Elastyczny układ sterowania CNC maszyn bazujący na PC,” *Mechanik*, nr 91, pp. 299-303, Kwiecień 2018.
- [13] „ST4118L3004-B,” [Online]. Available: <https://en.nanotec.com/products/376-st4118l3004-b>. [Data uzyskania dostępu: 14 styczeń 2024].
- [14] „NOE2-05-B14,” [Online]. Available: <https://en.nanotec.com/products/784-noe2-05-b14-5mm>. [Data uzyskania dostępu: 14 styczeń 2024].
- [15] „C5-E - Controller for BLDC and Stepper motors,” [Online]. Available: <https://en.nanotec.com/products/1764-c5-e-controller-for-stepper-motors-bldc>. [Data uzyskania dostępu: 14 styczeń 2024].
- [16] „FEICHAO C4250 560KV,” [Online]. Available: <https://www.aliexpress.com/item/1005001694867136.html>. [Data uzyskania dostępu: 14 styczeń 2024].
- [17] „ZUHEGELA 500 W 15 A DC12-36V,” [Online]. Available: <https://www.amazon.pl/ZUHEGELA-bezszerwotkowy-sterownika-przel%C4%85cznika-kontrolerem/dp/B0CLL49ZRG>. [Data uzyskania dostępu: 14 styczeń 2024].
- [18] J. W. K.-k. O. J. K. Xin-Jun Liu, „A New Approach to the Design of a DELTA Robot,” *Journal of Intelligent and Robotic Systems*, pp. 209-225, 2004.
- [19] S. Graves, „Johann C. Rocholl (Rostock) Style Delta Robot Kinematics,” 2015.
- [20] Reprap, „Delta geometry,” [Online]. Available: https://reprap.org/wiki/Delta_geometry. [Data uzyskania dostępu: 23 styczeń 2024].
- [21] G. C. Karasek, „Sterowanie maszyną typu Delta z wykorzystaniem systemu Linux i magistrali EtherCAT,” Toruń, 2021.
- [22] M. Bogłowski, „Modernizacja układu sterowania drukarki 3D o kinematyce równoległej typu delta liniowa,” Toruń, 2023.
- [23] „IgH EtherCAT Master,” [Online]. Available: <https://gitlab.com/etherlab.org/ethercat>. [Data uzyskania dostępu: 15 styczeń 2024].
- [24] „C5-E EtherCAT/USB Technical Manual,” [Online]. Available: https://en.nanotec.com/fileadmin/files/Handbuecher/Motorsteuerungen/C5-E/fir-v2213/C5E_EtherCAT_USB_Technical-Manual_V1.6.0.pdf. [Data uzyskania dostępu: 15 grudzień 2023].

- [25] G. E. Thyer, Computer Numerical Control of Machine Tools, Newnes, 1991.
- [26] M. Fitzpatrick, Machining and CNC Technology, McGraw-Hill Education, 2018.
- [27] K. Evans, Programming of CNC Machines, Fourth Edition, Industrial Press, Inc., 2016.
- [28] D. Kandray, Programmable Automation Technologies: An Introduction to CNC, Robotics and PLCs, Industrial Press, 2010.
- [29] „MathWorks,” [Online]. Available: https://www.mathworks.com/help/examples/robotics/win64/DesignATrajectoryWithVelocityLimitsUsingTrapVelTrajExample_03.png. [Data uzyskania dostępu: 23 grudzień 2023].
- [30] „Machine Design,” [Online]. Available: https://img.machinedesign.com/files/base/ebm/machinedesign/image/2019/10/machine_design_com_sites_machinedesign.com_files_Pulse_Control_Graphic2.png. [Data uzyskania dostępu: 23 grudzień 2023].

Załącznik nr 1. Listingi plików konfiguracyjnych oprogramowania LinuxCNC

```
##### Loading #####
loadrt [KINS]KINEMATICS
loadrt [EMCMOT]EMCMOT servo_period_nsec=[EMCMOT]SERVO_PERIOD num_joints=[KINS]JOINTS
loadusr -W ./lcec_conf ethercat-conf.xml
loadrt lcec
loadrt cia402 count=3
loadusr -W lineardelta MIN_JOINT=-420
loadrt scale count=2
loadrt conv_u32_float
loadrt conv_float_u32
loadrt sum2 count=1
loadrt logic count=1 personality=0x104

##### Linking functions #####
# read
addf lcec.read-all servo-thread
addf cia402.0.read-all servo-thread
addf cia402.1.read-all servo-thread
addf cia402.2.read-all servo-thread
# calculate
addf sampler.0 servo-thread
addf scale.0 servo-thread
addf conv-float-u32.0 servo-thread
addf logic.0 servo-thread
addf motion-command-handler servo-thread
addf motion-controller servo-thread
# write
addf cia402.0.write-all servo-thread
addf cia402.1.write-all servo-thread
addf cia402.2.write-all servo-thread
addf lcec.write-all servo-thread

##### Nets #####
net L lineardeltakins.L => lineardelta.L
net R lineardeltakins.R => lineardelta.R
sets L 187.5
sets R 102.75

# estop loopback
net estop-in-0 logic.0.in-00 => lcec.0.0.slave-state-op
net estop-in-1 logic.0.in-01 => lcec.0.1.slave-state-op
net estop-in-2 logic.0.in-02 => lcec.0.2.slave-state-op
net estop-in-3 logic.0.in-03 => iocontrol.0.user-enable-out
net emc-enable logic.0.and => iocontrol.0.emc-enable-in
```

```

# speed

net max-speed => lcec.0.0.max-speed
net max-speed => lcec.0.1.max-speed
net max-speed => lcec.0.2.max-speed
sets max-speed 15 # mm/s

## Joint 0 ##
setup cia402.0.csp-mode 1
setup cia402.0.pos-scale 1000
# from servo(ethercat) to cia402
net 0-statusword      lcec.0.0.cia-statusword => cia402.0.statusword
net 0-opmode-display  lcec.0.0.opmode-display => cia402.0.opmode-display
net 0-drv-act-pos     lcec.0.0.actual-position => cia402.0.drv-actual-position
net 0-drv-act-velo    lcec.0.0.actual-velocity => cia402.0.drv-actual-velocity
# from motion to cia
net 0-neg-lim-sw-in  lcec.0.0.in-neg-limit => joint.0.neg-lim-sw-in
net 0-pos-lim-sw-in  lcec.0.0.in-pos-limit => joint.0.pos-lim-sw-in
net 0-home-index <= joint.0.index-enable => cia402.0.home
net 0-enable      <= joint.0.amp-enable-out => cia402.0.enable
net 0-amp-fault   => joint.0.amp-fault-in  <= cia402.0.drv-fault
net 0-pos-cmd     <= joint.0.motor-pos-cmd => cia402.0.pos-cmd
net 0-pos-fb      => joint.0.motor-pos-fb  <= cia402.0.pos-fb
net 0-pos-fb => linedelta.joint0
# from cia402 to servo(ethercat)
net 0-controlword    cia402.0.controlword => lcec.0.0.cia-controlword
net 0-modes-of-operation cia402.0.opmode => lcec.0.0.opmode
net 0-drv-target-pos   cia402.0.drv-target-position => lcec.0.0.target-position
net 0-drv-target-velo   cia402.0.drv-target-velocity => lcec.0.0.target-velocity

## Joint 1 ##
setup cia402.1.csp-mode 1
setup cia402.1.pos-scale 1000
# from servo(ethercat) to cia402
net 1-statusword      lcec.0.1.cia-statusword => cia402.1.statusword
net 1-opmode-display  lcec.0.1.opmode-display => cia402.1.opmode-display
net 1-drv-act-pos     lcec.0.1.actual-position => cia402.1.drv-actual-position
net 1-drv-act-velo    lcec.0.1.actual-velocity => cia402.1.drv-actual-velocity
# from motion to cia
net 1-neg-lim-sw-in  lcec.0.1.in-neg-limit => joint.1.neg-lim-sw-in
net 1-pos-lim-sw-in  lcec.0.1.in-pos-limit => joint.1.pos-lim-sw-in
net 1-home-index <= joint.1.index-enable => cia402.1.home
net 1-enable      <= joint.1.amp-enable-out => cia402.1.enable
net 1-amp-fault   => joint.1.amp-fault-in  <= cia402.1.drv-fault
net 1-pos-cmd     <= joint.1.motor-pos-cmd => cia402.1.pos-cmd
net 1-pos-fb      => joint.1.motor-pos-fb  <= cia402.1.pos-fb
net 1-pos-fb => linedelta.joint1
# from cia402 to servo(ethercat)

```

```

net 1-controlword      cia402.1.controlword => lcec.0.1.cia-controlword
net 1-modes-of-operation cia402.1.opmode => lcec.0.1.opmode
net 1-drv-target-pos    cia402.1.drv-target-position => lcec.0.1.target-position
net 1-drv-target-velo   cia402.1.drv-target-velocity => lcec.0.1.target-velocity

## Joint 2 ##
setp cia402.2.csp-mode 1
setp cia402.2.pos-scale 1000
# from servo(ethercat) to cia402
net 2-statusword      lcec.0.2.cia-statusword => cia402.2.statusword
net 2-opmode-display   lcec.0.2.opmode-display => cia402.2.opmode-display
net 2-drv-act-pos      lcec.0.2.actual-position => cia402.2.drv-actual-position
net 2-drv-act-velo     lcec.0.2.actual-velocity => cia402.2.drv-actual-velocity
# from motion to cia
net 2-neg-lim-sw-in   lcec.0.2.in-neg-limit => joint.2.neg-lim-sw-in
net 2-pos-lim-sw-in   lcec.0.2.in-pos-limit => joint.2.pos-lim-sw-in
net 2-home-index <= joint.2.index-enable => cia402.2.home
net 2-enable      <= joint.2.amp-enable-out => cia402.2.enable
net 2-amp-fault    => joint.2.amp-fault-in <= cia402.2.drv-fault
net 2-pos-cmd      <= joint.2.motor-pos-cmd => cia402.2.pos-cmd
net 2-pos-fb       => joint.2.motor-pos-fb <= cia402.2.pos-fb
net 2-pos-fb => linedelta.joint2
# from cia402 to servo(ethercat)
net 2-controlword     cia402.2.controlword => lcec.0.2.cia-controlword
net 2-modes-of-operation cia402.2.opmode => lcec.0.2.opmode
net 2-drv-target-pos    cia402.2.drv-target-position => lcec.0.2.target-position
net 2-drv-target-velo   cia402.2.drv-target-velocity => lcec.0.2.target-velocity
## Spindle ## 80..65535 DAC PWM
setp scale.0.gain 4.87612
net spindle-speed-scale spindle.0.speed-out => scale.0.in
net spindle-conv-in conv-float-u32.0.in <= scale.0.out
net spindle-speed-DAC conv-float-u32.0.out => lcec.0.EL2502.pwm-ch2-out

```

Listing 1. Kod źródłowy pliku konfiguracyjnego HAL'a oprogramowania LinuxCNC

```

[EMC]
VERSION = 1.1
MACHINE = EtherCAT Machine
DEBUG = 1

[DISPLAY]
DISPLAY = axis
EDITOR = gedit
TOOL_EDITOR = tooledit
PYVCP_POSITION = RIGHT
# Cycle time, in seconds, that display will sleep between polls
CYCLE_TIME = 0.100
# Initial display setting for position, RELATIVE or MACHINE
POSITION_OFFSET = RELATIVE

```

```

POSITION_FEEDBACK = ACTUAL
MAX_FEED_OVERRIDE = 10.0
PROGRAM_PREFIX = /home/demo/linuxcnc/nc_files
INTRO_GRAPHIC = linuxcnc.gif
INTRO_TIME = 0
INCREMENTS = 100mm 10mm 5mm 1mm .5mm .1mm .05mm .01mm .005mm
DEFAULT_LINEAR_VELOCITY = 25.0
MAX_LINEAR_VELOCITY = 25.0
MAX_SPINDLE_OVERRIDE = 1.0

[FILTER]
PROGRAM_EXTENSION = .png,.gif,.jpg Greyscale Depth Image
PROGRAM_EXTENSION = .py Python Script
png = image-to-gcode
gif = image-to-gcode
jpg = image-to-gcode
py = python
nc = /usr/bin/axis

[RS274NGC]
PARAMETER_FILE = linuxcnc.var

[EMCMOT]
EMCMOT = motmod
COMM_TIMEOUT = 1.0
BASE_PERIOD = 1000000
SERVO_PERIOD = 1000000
TRAJ_PERIOD = 1000000

[TASK]
TASK = milltask
CYCLE_TIME = 0.001

[HAL]
HALFILE = umk_delta.hal
POSTGUI_HALFILE = postgui.hal
HALUI = halui

[EMCIO]
EMCIO = io
CYCLE_TIME = 0.100
TOOL_TABLE = cia402.tbl
TOOL_CHANGE_POSITION = 0 0 50.8

[TRAJ]
HOME = 0 0 0
COORDINATES = X Y Z
AXES = 3

```

```

JOINTS = 3
LINEAR_UNITS = mm
ANGULAR_UNITS = deg
DEFAULT_LINEAR_VELOCITY = 15.0
MAX_LINEAR_VELOCITY = 15.0
POSITION_FILE = position.txt
DEFAULT_VELOCITY = 10.0
DEFAULT_ACCELERATION = 10.0

[KINS]
KINEMATICS = lineardeltakins
JOINTS = 3

[AXIS_X]
TYPE = LINEAR
MIN_LIMIT = -50
MAX_LIMIT = 50
MAX_VELOCITY = 15.0
MAX_ACCELERATION = 100.0

[AXIS_Y]
TYPE = LINEAR
MIN_LIMIT = -50
MAX_LIMIT = 50
MAX_VELOCITY = 15.0
MAX_ACCELERATION = 100.0

[AXIS_Z]
TYPE = LINEAR
MIN_LIMIT = -345
MAX_LIMIT = -125
MAX_VELOCITY = 15.0
MAX_ACCELERATION = 100.0

[JOINT_0]
TYPE = LINEAR
MAX_VELOCITY = 15.0
MAX_ACCELERATION = 100.0
MIN_LIMIT = -280
MAX_LIMIT = 25
HOME_OFFSET = 0
HOME_SEARCH_VEL = 0.0
HOME_LATCH_VEL = 2.0
HOME_IGNORE_LIMITS = NO
HOME_USE_INDEX = YES
HOME_SEQUENCE = 0
FERROR = 1000.0
MIN_FERROR = 1000.0

```

```

[JOINT_1]
TYPE = LINEAR
MAX_VELOCITY = 15.0
MAX_ACCELERATION = 100.0
MIN_LIMIT = -280
MAX_LIMIT = 25
HOME_OFFSET = 0
HOME_SEARCH_VEL = 0.0
HOME_LATCH_VEL = 2.0
HOME_IGNORE_LIMITS = NO
HOME_USE_INDEX = YES
HOME_SEQUENCE = 0
FERROR = 1000.0
MIN_FERROR = 1000.0

[JOINT_2]
TYPE = LINEAR
MAX_VELOCITY = 15.0
MAX_ACCELERATION = 100.0
MIN_LIMIT = -280
MAX_LIMIT = 25
HOME_OFFSET = 0
HOME_SEARCH_VEL = 0.0
HOME_LATCH_VEL = 2.0
HOME_IGNORE_LIMITS = NO
HOME_USE_INDEX = YES
HOME_SEQUENCE = 0
FERROR = 1000.0
MIN_FERROR = 1000.0

```

Listing 2. Kod źródłowy pliku konfiguracyjnego INI oprogramowania LinuxCNC

```

<masters>
  <master idx="0" appTimePeriod="1000000" refClockSyncCycles="10">
    <!-- Joint 0 -->
    <slave idx="0" type="generic" vid="0x0000026c" pid="0x0000004e" configPdos="true">
      <dcConf assignActivate="300" sync0Cycle="*1" sync0Shift="0"/>
      <syncManager idx="2" dir="out">
        <pdo idx="1600">
          <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="cia-controlword" halType="u32"/>
          <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>
          <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
          <pdoEntry idx="60FF" subIdx="00" bitLen="32" halPin="target-velocity" halType="s32"/>
          <pdoEntry idx="6080" subIdx="00" bitLen="32" halPin="max-speed" halType="u32"/>
        </pdo>
      </syncManager>
      <syncManager idx="3" dir="in">
        <pdo idx="1a00">
          <pdoEntry idx="6041" subIdx="00" bitLen="16" halPin="cia-statusword" halType="u32"/>

```

```

<pdoEntry idx="6061" subIdx="00" bitLen="8" halPin="opmode-display" halType="s32"/>
<pdoEntry idx="6064" subIdx="00" bitLen="32" halPin="actual-position" halType="s32"/>
<pdoEntry idx="606C" subIdx="00" bitLen="32" halPin="actual-velocity" halType="s32"/>
<pdoEntry idx="60F4" subIdx="00" bitLen="32" halPin="following-error" halType="s32"/>
<!-- Digital_inputs (cia402 compatible) -->
<pdoEntry idx="60FD" subIdx="0" bitLen="32" halType="complex">
    <complexEntry bitLen="1" halPin="in-neg-limit" halType="bit"/>
    <complexEntry bitLen="1" halPin="in-pos-limit" halType="bit"/>
    <complexEntry bitLen="1" halPin="in-home" halType="bit"/>
    <complexEntry bitLen="13"/>
    <complexEntry bitLen="1" halPin="in-1" halType="bit"/>
    <complexEntry bitLen="1" halPin="in-2" halType="bit"/>
    <complexEntry bitLen="1" halPin="in-3" halType="bit"/>
    <complexEntry bitLen="1" halPin="in-4" halType="bit"/>
    <complexEntry bitLen="1" halPin="in-5" halType="bit"/>
    <complexEntry bitLen="1" halPin="in-6" halType="bit"/>
    <complexEntry bitLen="10"/>
</pdoEntry>
</pdo>
</syncManager>
</slave>
<!-- Joint 1 -->
<slave idx="1" type="generic" vid="0x0000026c" pid="0x0000004e" configPdos="true">
    <dcConf assignActivate="300" sync0Cycle="*1" sync0Shift="0"/>
    <syncManager idx="2" dir="out">
        <pdo idx="1600">
            <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="cia-controlword" halType="u32"/>
            <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>
            <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
            <pdoEntry idx="60FF" subIdx="00" bitLen="32" halPin="target-velocity" halType="s32"/>
            <pdoEntry idx="6080" subIdx="00" bitLen="32" halPin="max-speed" halType="u32"/>
        </pdo>
    </syncManager>
    <syncManager idx="3" dir="in">
        <pdo idx="1a00">
            <pdoEntry idx="6041" subIdx="00" bitLen="16" halPin="cia-statusword" halType="u32"/>
            <pdoEntry idx="6061" subIdx="00" bitLen="8" halPin="opmode-display" halType="s32"/>
            <pdoEntry idx="6064" subIdx="00" bitLen="32" halPin="actual-position" halType="s32"/>
            <pdoEntry idx="606C" subIdx="00" bitLen="32" halPin="actual-velocity" halType="s32"/>
            <pdoEntry idx="6077" subIdx="00" bitLen="32" halPin="actual-torque" halType="s32"/>
            <!-- Digital_inputs (cia402 compatible) -->
            <pdoEntry idx="60FD" subIdx="0" bitLen="32" halType="complex">
                <complexEntry bitLen="1" halPin="in-neg-limit" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-pos-limit" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-home" halType="bit"/>
                <complexEntry bitLen="13"/>
                <complexEntry bitLen="1" halPin="in-1" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-2" halType="bit"/>

```

```

        <complexEntry bitLen="1" halPin="in-3" halType="bit"/>
        <complexEntry bitLen="1" halPin="in-4" halType="bit"/>
        <complexEntry bitLen="1" halPin="in-5" halType="bit"/>
        <complexEntry bitLen="1" halPin="in-6" halType="bit"/>
        <complexEntry bitLen="10"/>
    </pdoEntry>
</pdo>
</syncManager>
</slave>
<!-- Joint 2 -->
<slave idx="2" type="generic" vid="0x0000026c" pid="0x0000004e" configPdos="true">
    <dcConf assignActivate="300" sync0Cycle="*1" sync0Shift="0"/>
    <syncManager idx="2" dir="out">
        <pdo idx="1600">
            <pdoEntry idx="6040" subIdx="00" bitLen="16" halPin="cia-controlword" halType="u32"/>
            <pdoEntry idx="6060" subIdx="00" bitLen="8" halPin="opmode" halType="s32"/>
            <pdoEntry idx="607A" subIdx="00" bitLen="32" halPin="target-position" halType="s32"/>
            <pdoEntry idx="60FF" subIdx="00" bitLen="32" halPin="target-velocity" halType="s32"/>
            <pdoEntry idx="6080" subIdx="00" bitLen="32" halPin="max-speed" halType="u32"/>
        </pdo>
    </syncManager>
    <syncManager idx="3" dir="in">
        <pdo idx="1a00">
            <pdoEntry idx="6041" subIdx="00" bitLen="16" halPin="cia-statusword" halType="u32"/>
            <pdoEntry idx="6061" subIdx="00" bitLen="8" halPin="opmode-display" halType="s32"/>
            <pdoEntry idx="6064" subIdx="00" bitLen="32" halPin="actual-position" halType="s32"/>
            <pdoEntry idx="606C" subIdx="00" bitLen="32" halPin="actual-velocity" halType="s32"/>
            <pdoEntry idx="6077" subIdx="00" bitLen="32" halPin="actual-torque" halType="s32"/>
            <!-- Digital_inputs (cia402 compatible) -->
            <pdoEntry idx="60FD" subIdx="0" bitLen="32" halType="complex">
                <complexEntry bitLen="1" halPin="in-neg-limit" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-pos-limit" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-home" halType="bit"/>
                <complexEntry bitLen="13"/>
                <complexEntry bitLen="1" halPin="in-1" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-2" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-3" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-4" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-5" halType="bit"/>
                <complexEntry bitLen="1" halPin="in-6" halType="bit"/>
                <complexEntry bitLen="10"/>
            </pdoEntry>
        </pdo>
    </syncManager>
</slave>
<slave idx="3" name="EK1818" type="generic" vid="0x00000002" pid="0x071a2c52" configPdos="true">
    <syncManager idx="2" dir="out">
        <pdo idx="0x1600">

```

```

        <pdoEntry idx="0x7000" subIdx="1" bitLen="1" halPin="out1" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1601">
        <pdoEntry idx="0x7010" subIdx="1" bitLen="1" halPin="out2" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1602">
        <pdoEntry idx="0x7020" subIdx="1" bitLen="1" halPin="out3" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1603">
        <pdoEntry idx="0x7030" subIdx="1" bitLen="1" halPin="out4" halType="bit">
        </pdoEntry>
    </pdo>
</syncManager>
<syncManager idx="1" dir="in">
    <pdo idx="0x1a00">
        <pdoEntry idx="0x6000" subIdx="1" bitLen="1" halPin="in1" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1a01">
        <pdoEntry idx="0x6010" subIdx="1" bitLen="1" halPin="in2" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1a02">
        <pdoEntry idx="0x6020" subIdx="1" bitLen="1" halPin="in3" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1a03">
        <pdoEntry idx="0x6030" subIdx="1" bitLen="1" halPin="in4" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1a04">
        <pdoEntry idx="0x6040" subIdx="1" bitLen="1" halPin="in5" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1a05">
        <pdoEntry idx="0x6050" subIdx="1" bitLen="1" halPin="in6" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1a06">
        <pdoEntry idx="0x6060" subIdx="1" bitLen="1" halPin="in7" halType="bit">
        </pdoEntry>
    </pdo>
    <pdo idx="0x1a07">
        <pdoEntry idx="0x6070" subIdx="1" bitLen="1" halPin="in8" halType="bit">
        </pdoEntry>
    </pdo>

```

```

        </pdo>
    </syncManager>
</slave>
<slave idx="4" type="generic" name="EL3202" vid="0x00000002" pid="0x0c823052" configPdos="true">
    <syncManager idx="3" dir="in">
        <pdo idx="0x1a00">
            <pdoEntry idx="0x6000" subIdx="1" bitLen="1" halPin="ch1-underrange" halType="bit">
            </pdoEntry>
            <pdoEntry idx="0x6000" subIdx="2" bitLen="1" halPin="ch1-overrange" halType="bit">
            </pdoEntry>
            <pdoEntry idx="0x6000" subIdx="17" bitLen="16" halPin="ch1-value" halType="s32">
            </pdoEntry>
        </pdo>
        <pdo idx="0x1a01">
            <pdoEntry idx="0x6010" subIdx="1" bitLen="1" halPin="ch2-underrange" halType="bit">
            </pdoEntry>
            <pdoEntry idx="0x6010" subIdx="2" bitLen="1" halPin="ch2-overrange" halType="bit">
            </pdoEntry>
            <pdoEntry idx="0x6010" subIdx="17" bitLen="16" halPin="ch2-value" halType="s32">
            </pdoEntry>
        </pdo>
    </syncManager>
</slave>
<slave idx="5" type="generic" name="EL2502" vid="0x00000002" pid="0x09c63052" configPdos="true">
    <!-- Channel 1 -->
    <sdoConfig idx="8000" subIdx="2">
        <sdoDataRaw data="01"/>
    </sdoConfig>
    <sdoConfig idx="8000" subIdx="7">
        <sdoDataRaw data="00"/>
    </sdoConfig>
    <sdoConfig idx="8000" subIdx="15">
        <sdoDataRaw data="E8 03"/>
    </sdoConfig>
    <!-- Channel 2 -->
    <sdoConfig idx="8010" subIdx="2">
        <sdoDataRaw data="01"/>
    </sdoConfig>
    <sdoConfig idx="8010" subIdx="7">
        <sdoDataRaw data="00"/>
    </sdoConfig>
    <sdoConfig idx="8010" subIdx="15">
        <sdoDataRaw data="F2 03"/>
    </sdoConfig>
    <syncManager idx="2" dir="out">
        <pdo idx="0x1600">
            <pdoEntry idx="0x7000" subIdx="17" bitLen="16" halPin="pwm-ch1-out" halType="u32">
            </pdoEntry>

```

```
</pdo>
<pdo idx="0x1601">
    <pdoEntry idx="0x7010" subIdx="17" bitLen="16" halPin="pwm-ch2-out" halType="u32">
        </pdoEntry>
    </pdo>
</syncManager>
</slave>
</master>
</masters>
```

Listing 3. Kod źródłowy pliku ethercat-conf.xml mapującego zmienne procesowe EtherCAT'a do pinów HAL'a

Załącznik nr 2. Schemat elektryczny układu sterowania numerycznego



Tel.

Uniwersytet Mikołaja Kopernika w Toruniu

Firma / klient
Opis projektu
Numer projektu
Komisja

Projekt maszyny numerycznej typu Delta
 IEC_bas001
 UMK

Producent (firma)

Uniwersytet Mikołaja Kopernika w Toruniu

Ścieżka

Nazwa projektu

Fabrykat / wyrob

Typ

Miejsce instalacji

Osoba odpowiedzialna

Elementy specjalne

Schemat elektryczny sterowania numerycznego
 Delta Numerical Machine

Utworzono dnia 12.09.2022
 Edytowano dnia 05.02.2024

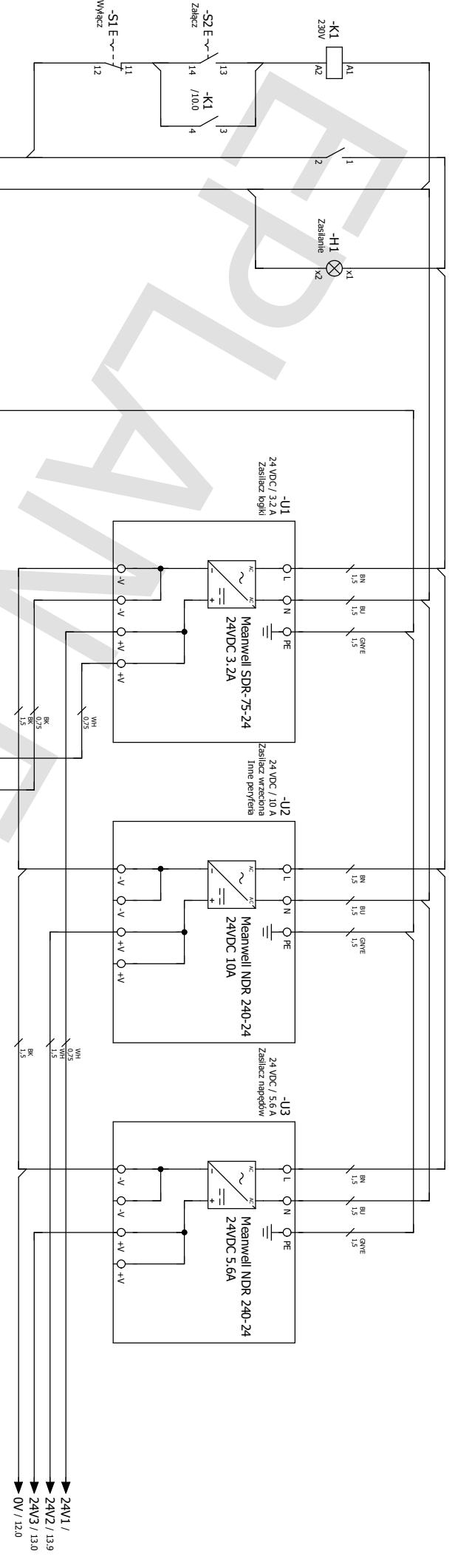
przez (Skrót) Huawei

Ilość stron 12

Zmiana	Data	Nazwa	Org.	Data	Etykieta	UMK
				29.09.2022	Huawei	Projekt maszyny numerycznej typu Delta

Uniw. M. Kopernika w Toruniu	Strona tytułowa / Okładka	= Sterownik
		+
		Arkusz 1

+Szafa/10



Włącznik nadprądowy
-F1 2A

-WE1:1
-WE1:2 = -S2 na zbiornik PE

BECKHOFF
-EK1818
/18.3

1 -> 2 / 10.1
3 -> 4 / 10.0

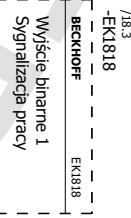
-XD1

Zasilanie

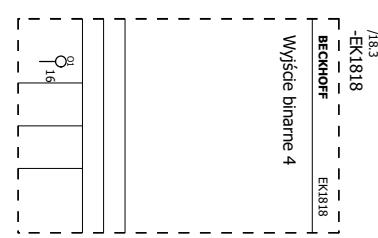
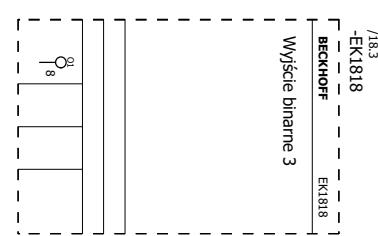
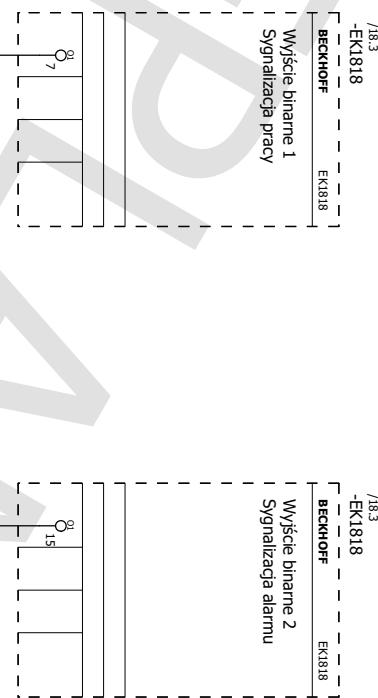
Zasilanie coupler'a EK1818
oraz szyny zasilającej moduły

		Data	05.02.2024	UMK	Uniwersytet Mikołaja Kopernika w Toruniu	Zasilanie 230 / 24 V	= Sterownik
		Eco	Huawei	Projekt maszyny numerycznej typu Delta			+ Szara
		Sprawdz.					Aktusz 10
Zmiana	Data	Nazwa	Oryg.		Rekomendata za	Zasłony przez	Strona 2 / 12

Sygnalizacja pracy



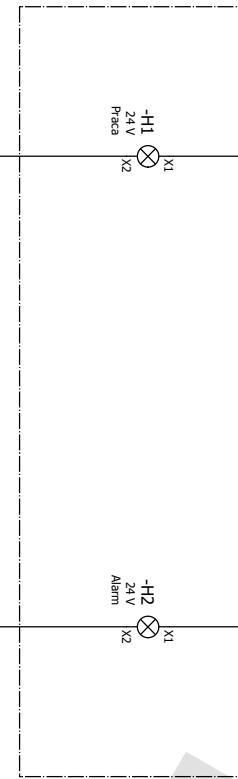
Signalizacja alarmu



10.9 / 0V

0V / 13.0

+Panel przedni



10

		Data	05.02.2024	UMK	
		Etykieta	Huawei	Projekt maszyny numerycznej typu Delta	
Zmiana	Data	Nazwa	Org.	Rekomendata za	Zastąpiony przez

Wydania

13

= Sterownik

+ Stan

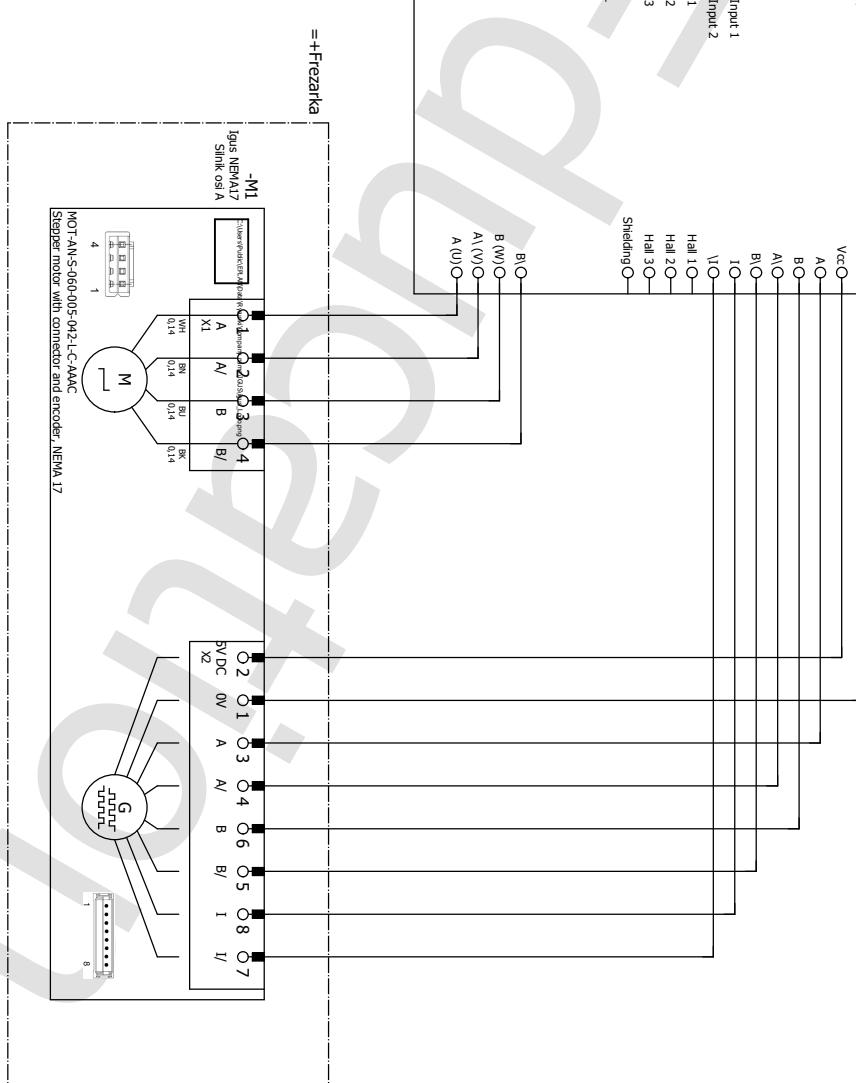
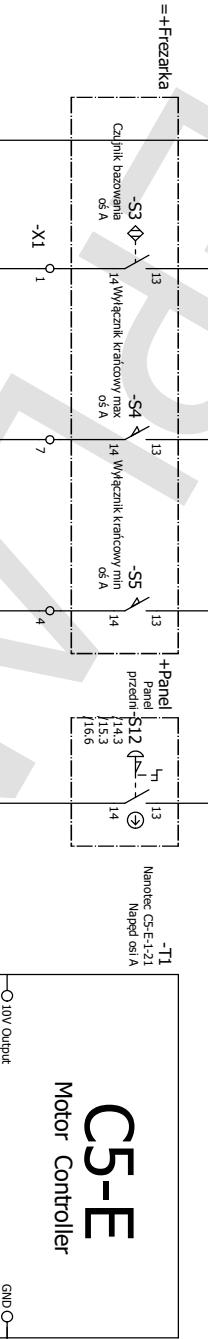
Aktusz

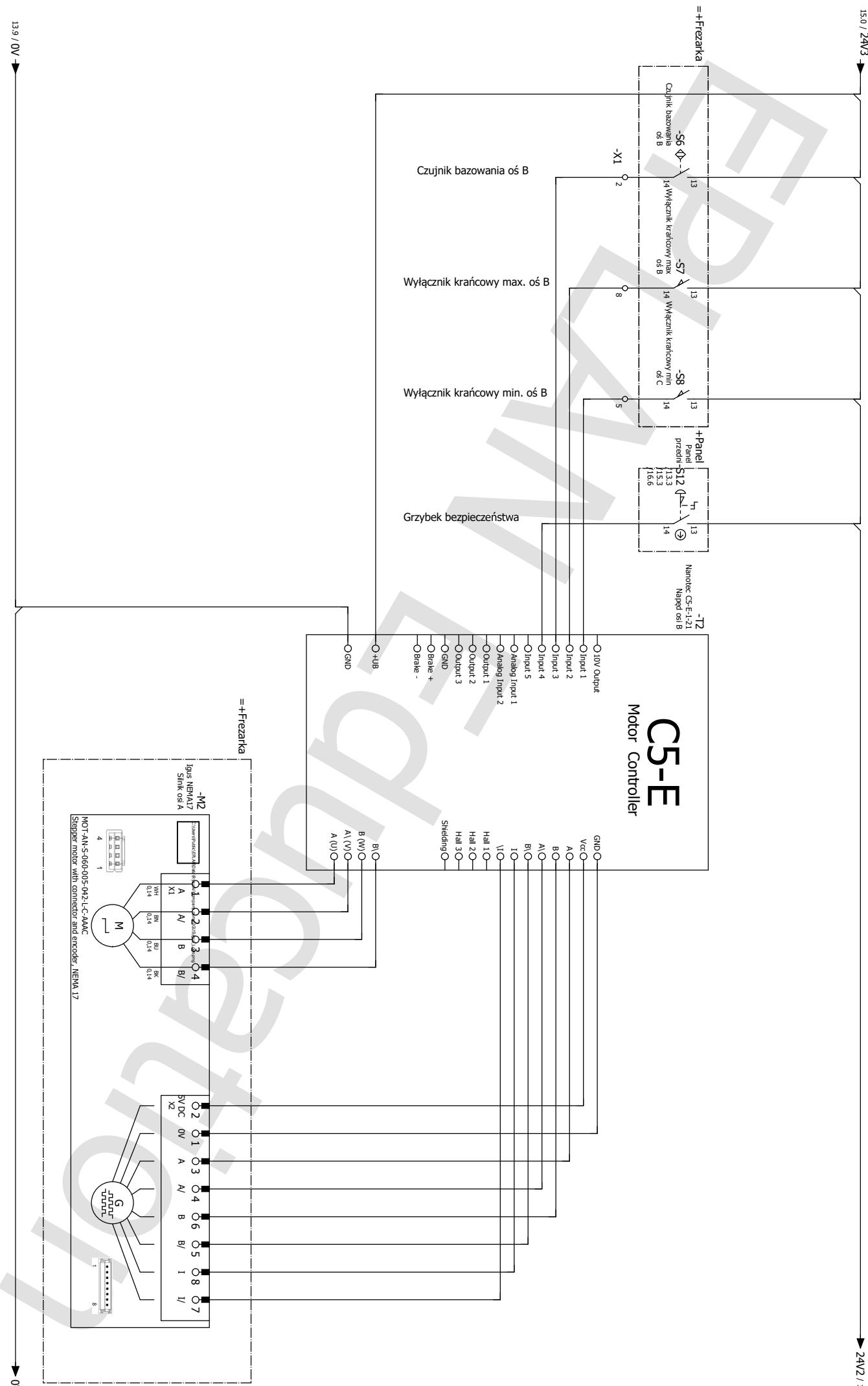
12

Strona

3 / 12

IEC_60501





13.9 / 1V →

1000

1

igus NEM
Slink O

AL17
SIA

1
A
X1
WH
0.14

14
K
3/

1

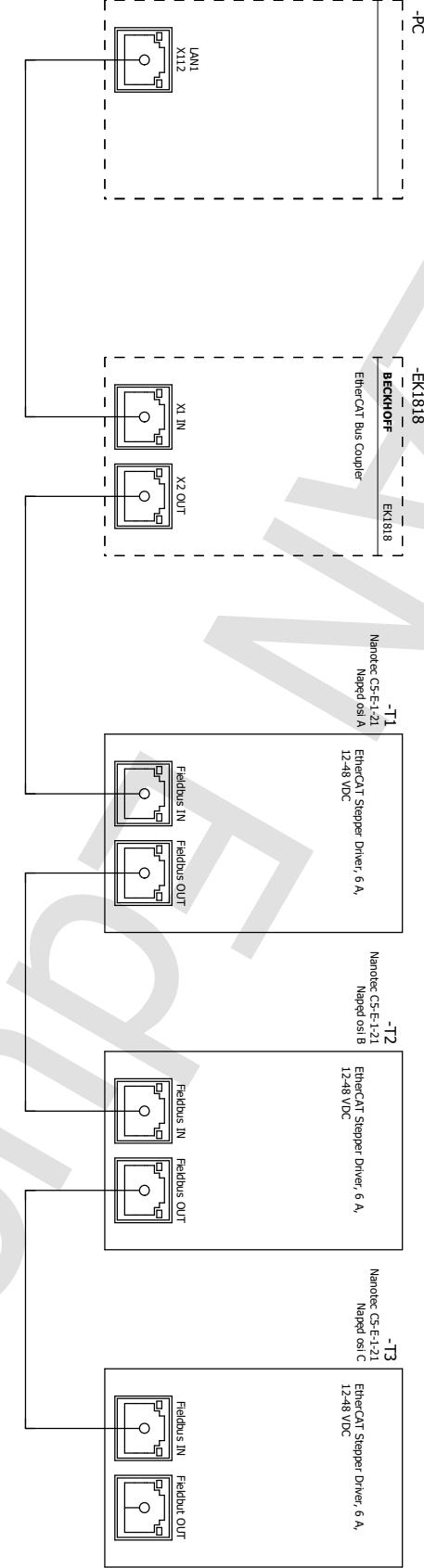
5V
DC
0V

A
A/
A/

B
B/
B/

The diagram shows a 10-bit binary counter with a 4-to-10 decoder. The counter has 10 flip-flops, each with a clock input (C), a data input (D), and a clock enable input (CE). The CE inputs are connected to the outputs of a 4-to-10 decoder. The decoder takes four inputs (A, B, C, D) and produces ten outputs (Y0 to Y9). The Y0 output is connected to the D input of the first flip-flop, and the Y1 output is connected to its CE input. This pattern continues for all ten flip-flops. The A, B, and C inputs of the decoder are connected to the outputs of a 3-to-8 decoder, which receives three inputs (X, Y, Z).

1



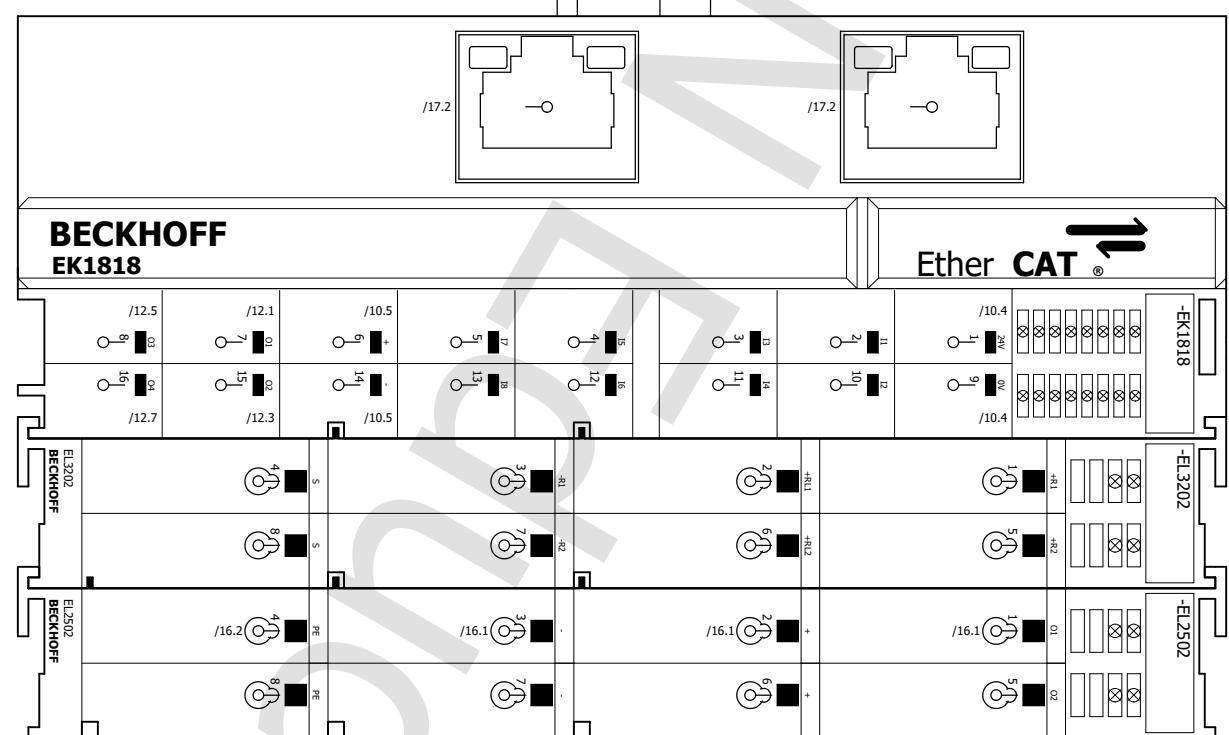
16
Data 05.02.2024 UMK
Etyka Huawei
Projekt maszyny numerycznej typu Delta
Rekomendata za Zastąpiony przez
Zmiana Nazwa Oryg.

Universytet Mikołaja Kopernika w Toruniu
Topologia EtherCAT

IEC_60501

18

= Sterownik
+ Szara
Aktusz 17
Strona 8 / 12



		Data	05.02.2024	UMK		
		Etykieta	Huawei			
		Spakowiz.				
Zmiana	Data	Nazwa	Oryg.	Rekomendata za	Zastąpiony przez	

Universytet Mikołaja Kopernika
w Toruniu
Przegląd modułów Beckhoff

= Sterownik
+ Szara

= Dokumenty/+31
= IEC_60501
Aktusz 18
Strona 9 / 12

Całościowa lista artykułów

F02_001

Numer katalogowy	Ilość	Opis Oznaczenie	Numer typu	Producent Dostawca	Cena detaliczna	Cena łączna	Poz.
MOT-AN-S-060-005-042-L-C-AAAC	3	Stepper motor with connector and encoder, NEMA 17	MOT-AN-S-060-005-042-L-C-AAAC IGUS, MOT-AN-S-060-005-042-L-C-AAAC	IGUS	0,00	0,00	
EK1818	0						
EL2502	1	EtherCAT Bus Coupler	EK1818 BEC, EK1818	BEC	0,00	0,00	
278523	1	2 Channel Pulse Width Output Terminal, 24 VDC	EL2502 BEC, EL2502	BEC	0,00	0,00	
C5-E-1-21	1	Analog Input Terminals Pt100 (RTD)	EL3202 BEC, EL3202	BEC	0,00	0,00	
SDR-75-24	1	Vyłącznik nadprądowy, 2 A, 1p, charakterystyka: B	FAZ-B2/1 ETN/FAZ-B2/1	ETN	0,00	0,00	
NDR-240-24	1	EtherCAT Stepper Driver, 6 A, 12-48 VDC	NANOTECC5-E-1-21 MEANWELL SDR-75-24	NANOTEC	0,00	0,00	
3581000	2	Zasilacz na szynę DIN 24 V / 10 A	SV-3581000 MEANWELL NIR-240-24	MEANWELL	0,00	0,00	
9340500	1	SV Szyna zbiornica E-Qu, SXW: 15x5 mm, dł.: 2400 mm	SV-9340500 RIT-9340500	RIT	0,00	0,00	
3024481	2	SV Wspornik szyn zbiorniczych, 1-bieg, do szyn o wym. SXW: 12x5-30x10 mm	ATP-ST 6 PXC-3024481	RIT	0,00	0,00	
3208142	3	Przegroda rozdzielająca sekcje	D-PT 1,5/S PXC-3208142	PXC	0,00	0,00	
2313150	1	Szukna szkoda zamkająca	SZ-2313150 RIT-2313150	RIT	0,00	0,00	
XB4FBV3	1	Szukna nośna według EN 60 715, wersja TS 35/15, dł.: 2000 mm	XB4FBV3 SE-XB4FBV3	SE	0,00	0,00	
XB4FBV4	1	Zielona lampa kontrolna kompletna R22 LED 24V	XB4FBV4 SE-XB4FBV4	SE	0,00	0,00	
		Czerwona lampa kontrolna kompletna R22 LED 24V		SE	0,00	0,00	

36

		Data 05.02.2024	UMK	Uniwersytet Mikołaja Kopernika w Toruniu	Całościowa lista artykułów : TGUS,MOT-AN-S-060-005-042-LC-AAAC -	= Dokument
		Etykieta Huawei	Projekt maszyny numerycznej typu Delta			+ IEC_6001
Zmiana	Data	Nazwa	Oryg.	Rekomendata za	Zastąpiony przez	Aktusz 35 Strona 11 / 12

Zestawienie listew zaciskowych

Tekst definiujący listwy zaciskowe

1

Strona graficzna planów zacisków

Listwa zaciskowa	Tekst definiujący listwy zaciskowe	Zaciski					Strona graficzna planów zacisków
		Pierwsza	Ostatnia	Suma PE	Suma N	Ogółem	
=Sterownik+Szafa-X0	Masa 0 V			0	0	0	
=Sterownik+Szafa-X1	Przyłącza wejść/wyjść			1	15	0	
=Sterownik+Szafa-X24	Zasilanie 24 V / 3.2 A			0	0	0	=Dokument/31
=Sterownik+Szafa-XD1	Zasilanie 230 V	1	3	1	1	3	=Dokument/34