

ROBOCHALLENGE

BECKHOFF

New Automation Technology

Koordynacja pracy robota Fanuc
z chwytakiem Schunk w Beckhoff PLC

Spis treści

1. Szczegóły konkurencji	3
2. Wstęp.....	4
3. Opis zadania	4
4. Etapy zadania.....	4
4.1. Realizacja zadania – etap 1.....	5
Dodanie pliku konfiguracyjnego	5
Skanowanie magistrali EtherCAT	5
Konfiguracja karty sieciowej do obsługi protokołu EtherCAT	5
Skanowanie magistrali.....	5
Dodanie modułu EL7221 i osi do TwinCAT MOTION	6
Zakończenie konfiguracji magistrali EtherCAT	7
Ustawienie trybu pracy osi chwytaka.....	8
Ustawienie skalowania osi chwytaka	8
4.2. Realizacja zadania – etap 2.....	9
Deklaracja bloków funkcyjnych do obsługi robota i chwytaka	9
Linkowanie zmiennych.....	10
Linkowanie zmiennych robota.....	10
Linkowanie zmiennych osi chwytaka	10
Parametryzacja osi w TwinCAT MOTION	11
Ustawienie offsetu pozycji.....	11
Włączenie programowych krańcówek.....	11
4.3. Realizacja zadania – etap 3.....	13
Opis przygotowanych bloków funkcyjnych.....	13
Blok do sterowania pracą robota.....	13
Blok do sterowania pracą chwytaka	14
Sekwencja pracy robota i chwytaka	15
5. System oceniania	16

1. Szczegóły konkurencji

Nazwa konkurencji: „Koordynacja pracy robota FANUC i chwytaka Schunk w Beckhoff PLC”.

Komponenty **BECKHOFF**:

1. **C6015** - Ultra-compact industrial PC
<https://www.beckhoff.com/pl-pl/products/ipc/pcs/c60xx-ultra-compact-industrial-pcs/c6015-0010.html?>
2. **EK1100**- EtherCAT Coupler
<https://www.beckhoff.com/pl-pl/products/i-o/ethercat-terminals/ek1xxx-bk1xx0-ethercat-coupler/ek1100.html>
3. **EL1008** - EtherCAT Terminal, 8-channel digital input
<https://www.beckhoff.com/pl-pl/products/i-o/ethercat-terminals/el1xxx-digital-input/el1008.html>
4. **EL2008** - EtherCAT Terminal, 8-channel digital output
<https://www.beckhoff.com/pl-pl/products/i-o/ethercat-terminals/el2xxx-digital-output/el2008.html>
5. **EL7221-9014** - EtherCAT Terminal, 1-channel motion interface, servomotor, 48 V DC, 4.5 A
<https://www.beckhoff.com/pl-pl/products/i-o/ethercat-terminals/el7xxx-compact-drive-technology/el7211.html>
6. **AM8122** - Low-voltage servomotor, voltage range < 50 V DC
<https://www.beckhoff.com/pl-pl/products/motion/rotary-servomotors/am8100-servomotors-for-compact-drive-technology/am8122-wfyz.html>

Komponenty **FANUC**:

1. Robot FANUC **M-20iD/25**
<https://www.fanuc.eu/pl/pl/roboty/robot-strona-filtrowania/m-20-series/m-20id-25>

Komponenty **SCHUNK**:

1. **LEG 760-1-15-2-10X3-B** - Chwytnik synchroniczny
<https://schunk.com/pl/pl/systemy-chwytnikowe/chwytnik-rownolegly/leg/leg-760-1-15-2-10x3-b/p/000000000000308060>

2. Wstęp

Aplikacje związane z paletyzowaniem stanowią ważną część wszystkich zadań jakie spotykamy na współczesnych liniach produkcyjnych. Zadania te najczęściej polegają na podnoszeniu, pozycjonowaniu i odkładaniu opakowań w dokładnie określony sposób na przygotowaną wcześniej paletę. W związku z tym, że zadania te są obecne w znacznej części procesów produkcyjnych, dlatego mamy do czynienia z ich dużą różnorodnością wynikającą z charakterystyki paletyzowanych produktów, która znacząco wpływa na proces ich obsługi.

Aby sprostać wyzwaniom związanym z paletyzacją produktów, stosuje się zrobotyzowane systemy wyposażone w roboty przemysłowe oraz zaawansowane konstrukcyjnie chwytaki. Połączenie tych systemów daje elastyczne rozwiązanie pozwalające na precyzyjne dopasowanie ich do konkretnej aplikacji, często wymagającej obsługi produktów o różnych gabarytach lub właściwościach, a co za tym idzie, również szybkiego i łatwego parametryzowania algorytmu pracy.

Dobrze dobrany robot przemysłowy i odpowiedni system chwytania zapewnia znaczny wzrost produktywności, oszczędność pieniędzy i skrócenie czasu produkcji.

3. Opis zadania

Zadanie polega na uruchomieniu komunikacji z Robotem FANUC, przygotowaniu obsługi chwytaka SHUNK w oprogramowaniu TwinCAT, oraz zaprogramowaniu algorytmu koordynującego przebieg zadania paletyzacji.

W zadaniu zostanie wykorzystana komunikacja po protokole EtherCAT. Kompaktowy komputer przemysłowy **Beckhoff C6015** będzie w tej komunikacji pełnił rolę mastera EtherCAT'a, natomiast robot **FANUC M-20iD/25** będzie miał zaimplementowaną funkcję EtherCAT Slave. Do obsługi ruchu chwytaka zostanie wykorzystany moduł **EL7221** wraz z oprogramowaniem **TwinCAT MOTION**.

Kolejnym krokiem, po nawiązaniu komunikacji będzie zadeklarowanie zmiennych w module **TwinCAT PLC** i wykonanie linkowania tych zmiennych z fizycznymi IO. Ostatnią częścią zadania będzie oprogramowanie sekwencji przekładania dwóch produktów, o różnych gabarytach, pomiędzy dwiema lokalizacjami.

4. Etapy zadania

Etap 1.

Uruchomienie komunikacji z robotem po protokole EtherCAT oraz konfiguracja osi w Drive Manager:

- 1) Skanowanie i konfiguracja urządzeń na magistrali EtherCAT.
- 2) Ustawienie skalowania osi chwytaka.
- 3) Aktywacja konfiguracji.

Etap 2.

Przygotowanie projektu w module TwinCAT PLC i parametryzacja osi w TwinCAT MOTION:

- 4) Zadeklarowanie w module TwinCAT PLC instancji bloku funkcyjnego do obsługi robota **FANUC**.
- 5) Zadeklarowanie w module TwinCAT PLC instancji bloku funkcyjnego do obsługi chwytaka **SCHUNK** napędzanego przez serwośilnik **AM8122**.
- 6) Poprawna kompilacja projektu PLC.
- 7) Linkowanie zmiennych służących do realizacji komunikacji z robotem.

ROBOCHALLENGE

- 8) Linkowanie osi chwytaka z modułem EL7211 i strukturą AXIS_REF w TwinCAT PLC.
- 9) Parametryzacja osi chwytaka w module TwinCAT MOTION.
- 10) Aktywacja konfiguracji

Etap 3.

Przygotowanie programu synchronizującego pracę robota i chwytaka.

4.1. Realizacja zadania – etap 1

Na tym etapie należy skonfigurować magistralę EtherCAT oraz oś w module TwinCAT MOTION.

Dodanie pliku konfiguracyjnego

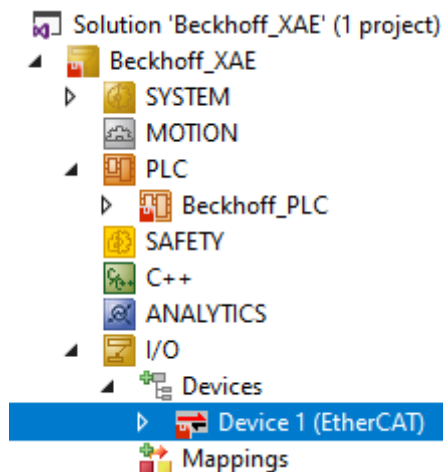
Dodawanie pliku:

- a) Na pulpicie w katalogu o nazwie „Pliki_do_zadania” znajduje się plik konfiguracyjny robota – „*fanuc_rs_esi0002.xml*”
- b) Plik konfiguracyjny należy przekopiować do katalogu „*C:\TwinCAT\3.1\Config\Io\EtherCAT*”.
- c) Po umieszczeniu pliku we wskazanym katalogu należy załadować pliki konfiguracyjne do systemu TwinCAT 3 XAE.
Aby to wykonać wracamy do środowiska TwinCAT, następnie z górnego paska narzędzi, wybieramy:
„*TwinCAT*” -> „*EtherCAT Devices*” -> „*Reload Device Descriptions*”.

Skanowanie magistrali EtherCAT

Konfiguracja karty sieciowej do obsługi protokołu EtherCAT

Rozwijamy drzewo projektu w następujący sposób:



Klikamy dwukrotnie na „*Device 1 (EtherCAT)*” następnie, w prawej części, wybieramy zakładkę „*Adapter*”.

Klikamy na przycisk „*Search...*” i w nowo otwartym oknie „*TwinCAT Ethernet Compatible Devices*” wybieramy kartę sieciową o nazwie „*EtherCAT*” i klikamy „*OK*”.

Skanowanie magistrali

Po zakończeniu konfiguracji karty sieciowej, klikamy prawym przyciskiem myszy na urządzenie „*Device 1 (EtherCAT)*” i z menu kontekstowego wybieramy opcję „*Scan*”.

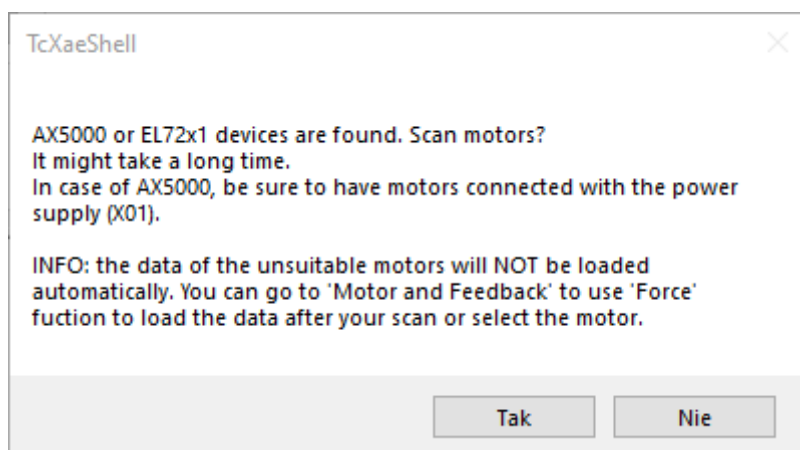
ROBOCHALLENGE

Dodanie modułu EL7221 i osi do TwinCAT MOTION

Następuje wyszukanie wszystkich urządzeń podłączonych do magistrali EtherCAT.

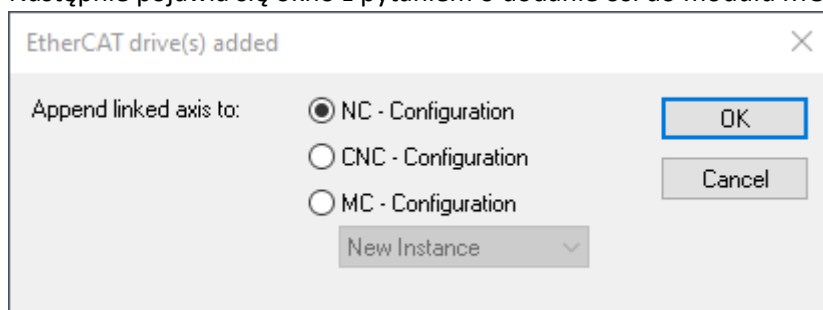
W trakcie skanowania zostanie wykryty moduł **EL7221** co spowoduje, że zostaną wyświetlone pytania o to czy chcemy wyskanować podłączony do niego silnik oraz czy dodać oś do modułu MOTION, która zostanie automatycznie zlinkowana z modułem **EL7221**.

Okno z pytaniem o skanowanie w poszukiwaniu silników wygląda następująco:



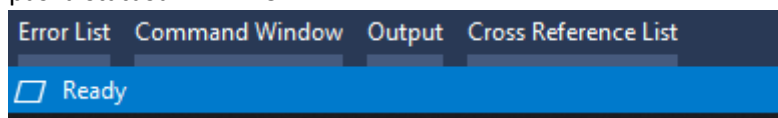
Na pytanie o skanowanie w poszukiwaniu silników klikamy „Tak”.

Następnie pojawia się okno z pytaniem o dodanie osi do modułu MOTION:



W powyższym oknie zaznaczamy „NC – Configuration” i klikamy „OK”.

Czekamy na zakończenie skanowania magistrali EtherCAT, poznamy to po napisie „Ready” na dolnym pasku statusu w TwinCAT:



Zakończenie konfiguracji magistrali EtherCAT

Po zakończeniu skanowania, na drzewku projektu w sekcji „I/O” pojawią się znalezione na magistrali EtherCAT urządzenia.

Konfiguracja powinna składać się z modułu EK1100, modułu EL7211 oraz robota FANUC o nazwie „Box 1 (FANUC R-30iB Plus)”.

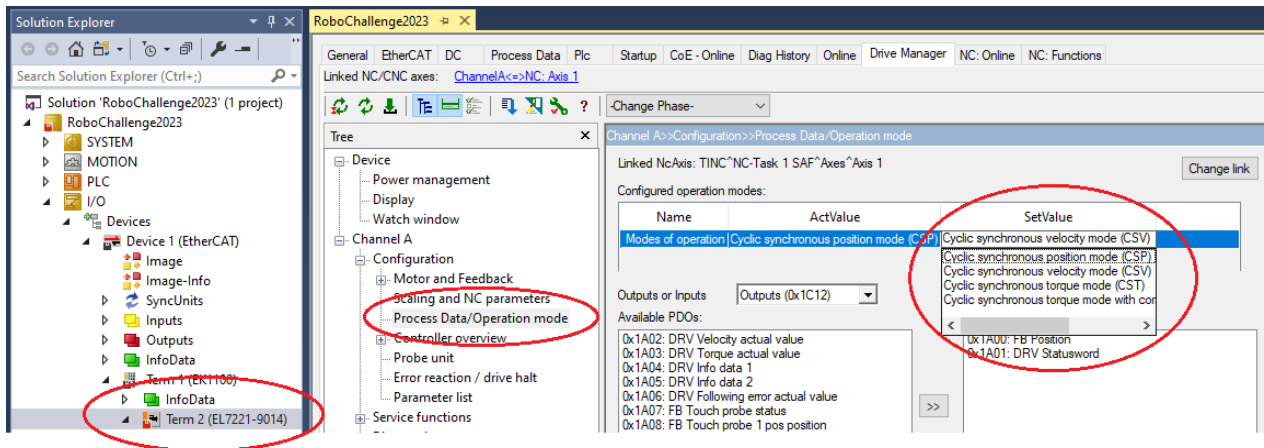
Sprawdzamy poprawność i przechodzimy do następnych czynności.

Ustawienie trybu pracy osi chwytaka

W przypadku modułu EL7221 domyślny tryb pracy to tryb prędkościowy.

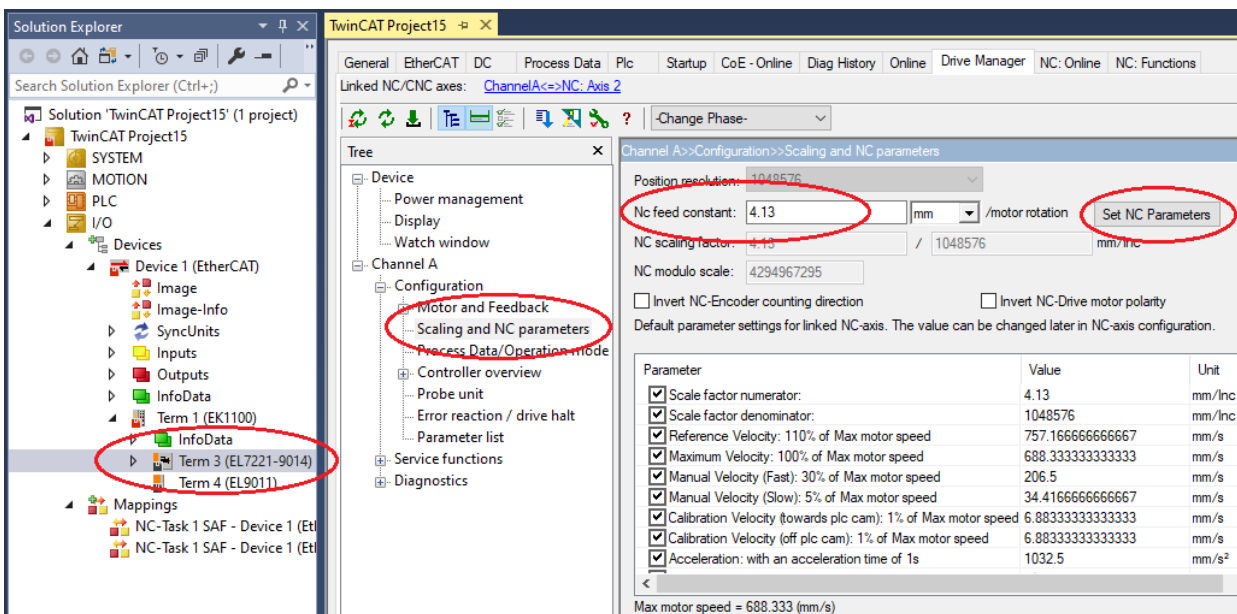
Aby uzyskać precyzyjne sterowanie chwytakiem należy skonfigurować moduł do pracy w trybie pozycyjnym.

W zakładce „Process Data/Operation mode” w zakładce „Drive Manager” przestawimy wartość w parametrze „Configured operation modes / SetValue” na „Cyclic synchronous position mode (CSP)”. Jeśli pojawi się okno z pytaniem czy wykonać relinkowanie osi chwytaka zgadzamy się na wykonanie tej operacji.



Ustawienie skalowania osi chwytaka

Na zakładce „Drive Manager”, z drzewka po lewej stronie wybieramy „Scaling and NC parameters”. Następnie w pole „Nc feed constant” wprowadzamy dystans jaki przejdzie oś, jeśli silnik wykona jeden obrót. W naszym przypadku jest to **4.13 mm**. Następnie wciskamy przycisk „Set NC Parameters”. W oknie, które się pojawi klikamy „OK” i po jego zamknięciu aktywujemy konfigurację.



4.2. Realizacja zadania – etap 2

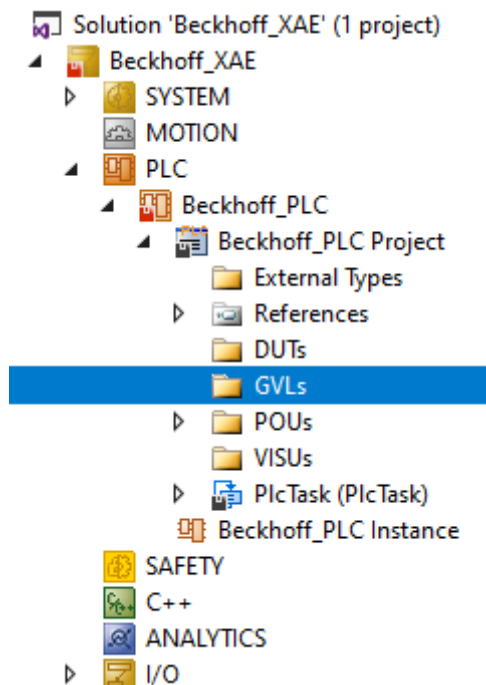
Na tym etapie należy uzupełnić projekt w module TwinCAT PLC o odpowiednie zmienne i bloki funkcyjne oraz Sparametryzować oś chwytaka w module TwinCAT MOTION.

Deklaracja bloków funkcyjnych do obsługi robota i chwytaka

Bloki funkcyjne do obsługi robota i chwytaka („FB_FanucRobot” i „FB_SchunkGripper”) zostały wcześniej przygotowane i umieszczone w katalogu „FBs” w projekcie PLC.

Wymienione wyżej bloki funkcyjne zawierają potrzebne zmienne wejściowe i wyjściowe, które mogą być zlinkowane z danymi udostępnianymi przez robota i oś chwytaka.

Wchodzimy w projekt PLC i w katalogu „GVLs” tworzymy plik (w instrukcji zastosowano nazwę „gvlRobotAndGripper”) do przechowania zmiennych globalnych (PPM -> „Add” -> „Global Variable List”):



W stworzonym pliku deklarujemy bloki funkcyjne do obsługi robota i chwytaka.

Przykładowa deklaracja bloku funkcyjnego:

fbRobot: FB_FanucRobot;

Po zadeklarowaniu bloków funkcyjnych należy skompilować projekt.

Można to zrobić wybierając z menu następującą opcję:

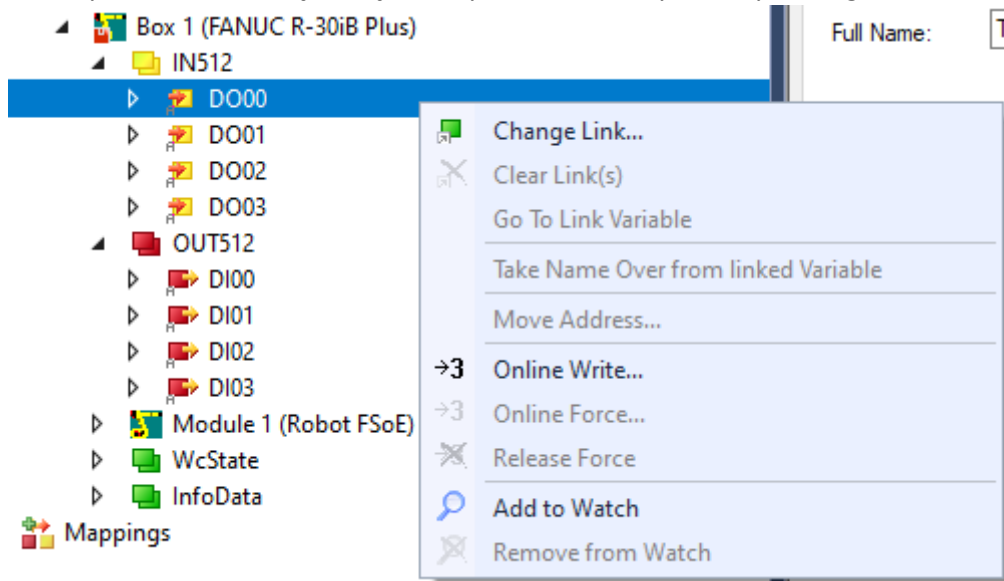
„Build” -> „Build Solution” lub skorzystać z kombinacji klawiszy „Ctrl + Shift + B”.

Po pomyślnym zakończeniu kompilacji można przystąpić do linkowania zmiennych.

Linkowanie zmiennych

Linkowanie zmiennych robota

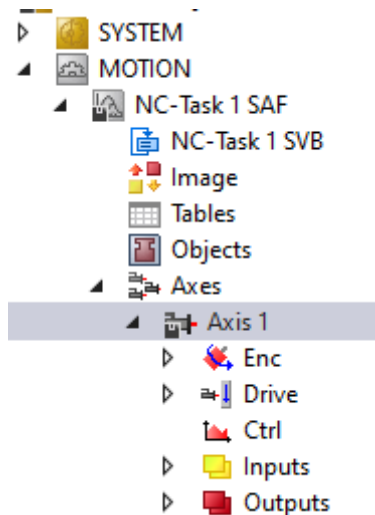
Klikamy PPM na zmienną, którą chcemy zalinkować i wybieramy „Change Link...”.



Otworzy się okno, w którym można wybrać zmienne (zadeklarowane wcześniej w projekcie PLC) do podlinkowania we wskazane miejsce.

Linkowanie zmiennych osi chwytaka

Wchodzimy w moduł TwinCAT MOTION i odnajdujemy wcześniej dodaną (automatycznie) oś chwytaka.



Klikamy podwójnie na nazwę osi i po prawej stronie wchodzimy w zakładkę „Settings”.

W tym miejscu mamy możliwość zlinkowania osi z modulem **EL7221** (to linkowanie zostało wykonane automatycznie w 1. etapie zadania) oraz ze strukturą pozwalającą na sterowanie osią z poziomu programu PLC.

Aby zalinkować oś ze strukturą w naszym programie PLC należy klikać na przycisk „Link to PLC...”, w nowo otwartym oknie wybrać odpowiednią strukturę i kliknąć „OK”.

Parametryzacja osi w TwinCAT MOTION

Ustawienie offsetu pozycji.

Aby skorygować pozycję przesyłaną odczytaną z enkodera absolutnego umieszczonego w silniku napędzającym chwytak należy odnaleźć parametr „Position Bias” i wpisać do niego wartość **108.25 mm**.

Parameter	Offline Value	Online Value
Encoder Evaluation:		
Invert Encoder Counting Direction	FALSE	FALSE
Scaling Factor Numerator	4.13	4.13
Scaling Factor Denominator (default: 1.0)	1048576.0	1048576.0
Position Bias	108.25	108.25
Modulo Factor (e.g. 360.0°)	360.0	360.0
Tolerance Window for Modulo Start	0.0	0.0
Encoder Mask (maximum encoder value)	0xFFFFFFFF	0xFFFFFFFF
Encoder Sub Mask (absolute range maximum value)	0x000FFFFF	0x000FFFFF
Reference System	'INCREMENTAL'	'INCREMENTAL'
Limit Switches:		
Soft Position Limit Minimum Monitoring	FALSE	FALSE
Minimum Position	0.0	0.0
Soft Position Limit Maximum Monitoring	FALSE	FALSE
Maximum Position	0.0	0.0

Włączenie programowych krańcówek

Aby uruchomić funkcjonalność ograniczania maksymalnej i minimalnej pozycji osi, bezpośrednio w oprogramowaniu, należy odnaleźć grupę parametrów o nazwie „Limit Switches” a następnie ustawić je w sposób pokazany poniżej.

Parameter	Offline Value	Online Value
Encoder Evaluation:		
Limit Switches:		
Soft Position Limit Minimum Monitoring	TRUE	FALSE
Minimum Position	0.0	0.01
Soft Position Limit Maximum Monitoring	TRUE	FALSE
Maximum Position	270.0	0.0
Filter:		
Homing:		
Other Settings:		

ROBOCHALLENGE

Po zalinkowaniu wszystkich zmiennych (wejściowych i wyjściowych) i skonfigurowaniu parametrów osi należy aktywować konfigurację poprzez wybranie na pasku menu ikony „Activate Configuration” i uruchomić system w trybie „Run” z włączonym autostartem programu PLC.

4.3. Realizacja zadania – etap 3

Realizacja etapu polega na zaprogramowaniu procedury przenoszenia produktów pomiędzy dwiema lokalizacjami w taki sposób, aby zsynchronizować pracę robota i chwytaka.

Do realizacji etapu należy wykorzystać przygotowane bloki funkcyjne do obsługi robota i chwytaka.

Za przeniesienie każdego produktu przyznawany jest jeden punkt.

Opis przygotowanych bloków funkcyjnych.

Blok do sterowania pracą robota.

Wejścia:

bEnable: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do robota wszystkich potrzebnych zezwoleń na pracę.

bGoToHomePosition: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do robota rozkazu przejazdu na pozycję startową.

bGoToSmallProductReceivingPosition: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do robota rozkazu przejazdu na pozycję pobrania małego detalu.

bGoToSmallProductPlacementPosition: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do robota rozkazu przejazdu na pozycję odłożenia małego detalu.

bGoToBigProductReceivingPosition: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do robota rozkazu przejazdu na pozycję pobrania dużego detalu.

bGoToBigProductPlacementPosition: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do robota rozkazu przejazdu na pozycję odłożenia dużego detalu.

Wyjścia:

bRobotReadyForTheNextTask: BOOL; Sygnał od robota, że jest gotowy na przyjęcie kolejnego rozkazu.

bRobotAtSmallProductReceivingPosition: BOOL; Sygnał od robota, że jest ustawiony na pozycji pobierania małego detalu.

bRobotAtSmallProductPlacementPosition: BOOL; Sygnał od robota, że jest ustawiony na pozycji odkładania małego detalu.

bRobotAtBigProductReceivingPosition: BOOL; Sygnał od robota, że jest ustawiony na pozycji pobierania dużego detalu.

bRobotAtBigProductPlacementPosition: BOOL; Sygnał od robota, że jest ustawiony na pozycji odkładania dużego detalu.

ROBOCHALLENGE

Blok do sterowania pracą chwytaka.

Wejścia:

bEnable: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do osi chwytaka polecenia załączenia regulatorów i pozwolenia na ruch.

bOpenRequest: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do chwytaka polecenia otwarcia uchwytu.

bCloseRequest: BOOL; Wejście reaguje na zbocze narastające. Podanie wartości TRUE powoduje wysłanie do chwytaka polecenia zamknięcia uchwytu.

iClosePosition: INT; Wybór pozycji zamkniętej: 1 -> pozycja zamknięcia odpowiednia dla małego detalu, 2 -> pozycja odpowiednia dla dużego detalu.

Wyjścia:

bGripperReadyForTheNextTask: BOOL; Chwytnak gotowy do przyjęcia kolejnego rozkazu.

bClosing: BOOL; Trwa zamykanie uchwytu.

bOpening: BOOL; Trwa otwieranie uchwytu.

bIsClosed: BOOL; Uchwyt jest zamknięty.

bIsOpened: BOOL; Uchwyt jest otwarty.

bAxisError: BOOL; Oś nie jest gotowa.

bMovingError: BOOL; Wystąpił błąd podczas ruchu uchwytu.

bParametersError: BOOL; Podane parametry są niepoprawne.

Sekwencja pracy robota i chwytaka

Należy pamiętać, aby w programie **umieścić wywołania bloków funkcyjnych** (*nazwa_instancji_bloku_funkcyjnego()*), jeśli ich zabraknie kod zawarty w blokach funkcyjnych nie będzie wykonywany przez sterownik.

Przykładowy algorytm:

- 1) Wysłanie do robota i chwytaka pozwolenia na pracę / rozkazu załączenia.
- 2) Sprawdzenie czy robot i chwytak zgłaszają gotowość do pracy.
- 3) Sprawdzenie czy chwytak jest otwarty, jeśli nie to otworzyć.
- 4) Jeśli chwytak otwarty to uruchomienie przejazdu robota na pozycję pobrania małego produktu.
- 5) Oczekiwanie na zakończenie przejazdu.
- 6) Rozkaz zamknięcia chwytaka.
- 7) Jeśli chwytak zamknięty to uruchomienie przejazdu na pozycję odłożenia małego produktu.
- 8) Oczekiwanie na zakończenie przejazdu.
- 9) Rozkaz otworzenia chwytaka.
- 10) Jeśli chwytak otwarty to uruchomienie przejazdu robota na pozycję pobrania dużego produktu.
- 11) Oczekiwanie na zakończenie przejazdu.
- 12) Rozkaz zamknięcia chwytaka.
- 13) Jeśli chwytak zamknięty to uruchomienie przejazdu na pozycję odłożenia dużego produktu.
- 14) Otworzenie chwytaka.
- 15) Jeśli chwytak otwarty to uruchomienie przejazdu robota na pozycję startową.

Po odpowiednim uzupełnieniu kodu w projekcie PLC należy zalogować się do sterownika w celu aktualizacji programu PLC. Robimy to wykorzystując zieloną ikonę o nazwie „Login”, która znajduje się na górnym pasku menu (logowanie jest możliwe, jeśli sterownik jest w trybie „Run”). Po kliknięciu tej ikony nastąpi kompilacja programu. Jeśli wykona się poprawnie program zostanie wgrany do sterownika PLC.

Składnia tej instrukcji jest następująca ({} - wyrażenie opcjonalne, <krok> - zmienna typu całkowitego, np. INT) :

```
CASE <krok> OF
  <krok_1>:
    <Instrukcje do wykonania w kroku pierwszym>
  <krok_2>:
    <Instrukcje do wykonania w kroku drugim>
  ...
  <krok_N>:
    <Instrukcje do wykonania w kroku N-tym>
{ELSE
  <<Instrukcje do wykonania jeśli numer kroku nie występuje na liście>>}
END_CASE;
```

5. System oceniania

1. Osiągnięcie celu zadania – 1 pkt.

Przeniesienie, za pomocą robota przemysłowego wyposażonego w chwytak, dwóch opakowań, o różnej wielkości, z pozycji pobierania na pozycję odkładania.

1. Zaliczenia etapów zadania:

a) Etap I – 1 pkt.

Celem etapu jest uruchomienie komunikacji z robotem po protokole EtherCAT oraz konfiguracja osi chwytaka w oprogramowaniu Drive Manager.

b) Etap II – 1 pkt.

Cel etapu to przygotowanie projektu w module TwinCAT PLC i parametryzacja osi chwytaka w TwinCAT MOTION.

c) Etap III – 2 pkt.

Cel etapu to przygotowanie programu PLC synchronizującego pracę robota i chwytaka w celu przeniesienia opakowań.

W tym etapie liczba punktów zależna jest od ilości przeniesionych produktów z pozycji pobrania na pozycję odłożenia. Za każdy przeniesiony produkt zostaje przyznany jeden punkt.

2. Czas wykonania po zapoznaniu się z instrukcją – maksymalnie 3 pkt.

a) Ukończenie zadania poniżej 30 min (3 pkt)

b) Ukończenie zadania poniżej 35 min (2 pkt)

c) Ukończenie zadania poniżej 40 min (1 pkt)

W trakcie wykonywania zadania, drużyna może poprosić o pomoc w wykonaniu danego punktu opiekuna stanowiska. Nie przysługuje jej wówczas punkt za wykonanie danego etapu.

Warunkiem udzielenia pomocy, jest stosowna ilość czasu na jej udzielenie.