



SIMATIC S7-1500T Motion Control

Przedmowa

Niniejsza publikacja powstała z myślą o inżynierach automatyki przemysłowej. Każdy automatyk pracujący w przemyśle – jako integrator, serwisant czy w sekcji utrzymaniu ruchu – na co dzień mierzy się z zagadnieniami związanymi ze sterownikiem programowalnym oraz techniką napędową. Podręcznik S7-1500T – SIMATIC Motion Control stanowi kompendium informacji w zakresie połączenia tych dwóch poziomów układu automatyki – czyli współpracy sterownika PLC z urządzeniami generującymi ruch (Motion Control).

Poradnik wprowadza czytelnika w podstawowe zagadnienia techniki serwonapędowej oraz problematykę integracji osi technologicznych w programie sterownika. Począwszy od teorii związanej ze strukturą systemu oraz układem pomiarowym, przez opis techniczny sterowania pojedynczym napędem (zadawanie prędkości/pozycjonowanie), a kończąc na zaawansowanych mechanizmach synchronizacji oraz sprzężeń krzywkowych w wieloosiowych układach Motion Control – te wszystkie zagadnienia zostały przedstawione, zarówno od strony teoretycznej jak i praktycznej (opis konfiguracji w środowisku inżynierskim) w niniejszym podręczniku.

Każdy, kto posiada fundamentalną wiedzę na temat sterowników programowalnych PLC, po lekturze dokumentu – zrozumie zasady pracy serwonapędu, mechanizmy działania nadrzędnego układu sterowania osiami napędowymi oraz reguły jakimi należy się kierować podczas budowania funkcjonalnie kompletnego oraz bezpiecznego systemu zarządzania ruchem.

Radostaw Krzyżanowski

SIEMENS

SIMATIC

Motion Control

<u>Zagadnienia podstawowe</u>	<u>1</u>
<u>Układ pomiarowy</u>	<u>2</u>
<u>Sterowanie pojedynczą osią</u>	<u>3</u>
<u>Ruch skoordynowany</u>	<u>4</u>
<u>Sprężenie krzywkowe</u>	<u>5</u>
Wejścia pomiarowe wyjścia krzywkowe	<u>6</u>
Bezpieczeństwo w układach Motion Control	<u>7</u>

03/2018

Informacje prawne

System ostrzeżeń

Niniejszy podręcznik zawiera informacje, których przestrzeganie jest niezbędne aby zapewnić bezpieczeństwo ludzi i zapobiec przed zniszczeniem maszyn. Zalecenia związane z bezpieczeństwem ludzi są eksponowane w ramach ze znakiem ostrzegawczym. Oznaczenia te stopniują ważność ostrzeżeń.



Niebezpieczeństwo

Nieprzestrzeganie zaleceń spowoduje poważne obrażenia a nawet śmierć ludzi.



Ostrzeżenie

Nieprzestrzeganie zaleceń może spowodować poważne obrażenia a nawet śmierć ludzi.



Ostrzeżenie

Nieprzestrzeganie zaleceń może spowodować wystąpienie drobnych obrażeń.

Zalecenie

Nieprzestrzeganie zaleceń może doprowadzić do zniszczenia mienia.

Jeśli widoczny jest więcej niż jeden znak ostrzegający, oznacza to zwiększony poziom zagrożeń. Ostrzeżenia związane z zagrożeniami dla osób mogą być także związane ze zniszczeniem mienia.

Kwalifikowany personel

Produkty opisane w niniejszym podręczniku mogą być obsługiwane tylko przez wykwalifikowany personel, który zna specyfikę pracy urządzeń, zna dokumentację urządzeń oraz związane z nimi przepisy bezpieczeństwa. Kwalifikowany personel to ludzie, którzy przeszli odpowiednie Szkolenia, posiadają doświadczenie w obsłudze urządzeń, potrafią zorganizować bezpieczne stanowisko pracy.

Właściwe użycie urządzeń SIEMENS

Zwróć uwagę na:



Ostrzeżenie

Produkty SIEMENS mogą być użyte tylko w aplikacjach opisanych w katalogu lub innych dedykowanych dokumentacjach. Jeśli z urządzeniami SIEMENS mają być użyte komponenty innych firm, muszą one posiadać odpowiednie dopuszczenia od Siemens. Aby zapewnić bezawaryjność, jakość oraz bezpieczeństwo pracy urządzeń SIEMENS muszą one być odpowiednio transportowane, przechowywane, instalowane, uruchamiane, eksploatowane tj. wg zaleceń producenta zawartych w dokumentacjach technicznych.

Znaki handlowe

Wszystkie znaki handlowe z[®] są zarejestrowanymi znakami handlowymi Siemens AG. Pozostałe oznaczenia zawarte w podręczniku mogą być znakami handlowymi firm trzecich. Ich publikacja bez zgody właściciela może naruszać jego dobra osobiste.

Przeniesienie odpowiedzialności

Dołożyliśmy wszelkich starań aby zawartość niniejszego podręcznika była aktualna z dostępnym oprogramowaniem i wersją sprzętu. Nie można jednak wykluczyć rozbieżności związanych z ich bieżącą aktualizacją przez fabryki produkcyjne. Informacje zawarte w kolejnych podręcznikach będą weryfikowane pod tym kątem na bieżąco.

Informacje związane z bezpieczeństwem

Siemens dostarcza produkty i rozwiązania dla przemysłu, które zapewniają bezpieczną pracę maszyn, urządzeń, sieci teleinformatycznych i całych zakładów. Stanowią one ważne ogniwo w globalnej koncepcji bezpieczeństwa przemysłowego. W myśl tej idei Siemens ciągle rozwija i doskonali swoje produkty. Siemens zaleca aby jego produkty były regularnie sprawdzane pod kątem nowych wersji oraz uaktualnień.

Bezpieczeństwo eksploatacji systemów produkcyjnych wymusza stosowanie najnowszych koncepcji ochrony (np. grupowanie urządzeń w lokalne podstrefy bezpieczeństwa) i integracji do systemu wszystkich komponentów stosując się do globalnej koncepcji bezpieczeństwa przemysłowego. Więcej informacji na ten temat można znaleźć w Internecie na stronie: <http://www.siemens.com/industrialsecurity>.

Aby być na bieżąco z publikowanymi informacjami, można zapisać się na listę odpowiedniego newslettera. Więcej informacji na ten temat można znaleźć na: <http://support.automation.siemens.com>.

Spis treści

Przedmowa	5
Spis treści	8
I. Zagadnienia podstawowe	
1. Sterowanie napędem	12
1.1 Sterowanie lokalne (napęd)	12
1.2 Struktura scentralizowana (PLC)	13
2. Komunikacja napęd–sterownik	15
2.1 Protokoły komunikacyjne	15
2.2 Telegramy PROFIDrive	16
2.3 Tryb izochroniczny czasu rzeczywistego (IRT)	16
3. Typowa konfiguracja systemu Motion Control ze sterowaniem nadrzędnym	21
4. S7 Motion Control	22
4.1 Zasada działania	22
4.2 TIA Portal	23
5. Pozycjonowanie	25
5.1 Struktura systemu	25
5.2 Interpolator	26
5.3 Regulator pozycji	26
5.4 Regulator prędkości	26
II. Układ pomiarowy	
6. Enkodery inkrementalne	28
6.1 TTL	31
6.2 HTL	32
6.3 SIN/COS	32
6.4 Połączenie z systemami SIMATIC/SINAMICS	33
7. Enkodery absolutne	35
7.1 Siemens DRIVE-CLiQ	38
7.2 SSI	39
7.3 EnDat	39
7.4 Hiperface	39
7.5 BiSS	40
7.6 SIN/COS ze ścieżką C/D	40
7.7 Resolver	41
7.8 Sieć przemysłowa	41
7.9 Enkoder liniowy	42
7.10 Połączenie z systemami SIMATIC/SINAMICS	42
III. Sterowanie pojedynczą osią	
8. Konfiguracja sprzętowa	46
8.1 Dodanie sterownika do projektu	46
8.2 Wstawienie do projektu napędu oraz jego konfiguracja	46
8.3 Uruchomienie napędu	48
8.4 Topologia sieci PROFINET IO IRT	50
9. Zadawanie prędkości	51

9.1	Konfiguracja obiektu technologicznego	51
9.2	Podstawowe funkcje interfejsu użytkownika	55
9.3	Ruch z zadaną prędkością	56
10.	Pozycjonowanie względne	58
10.1	Konfiguracja obiektu technologicznego	58
10.2	Podstawowe funkcje interfejsu użytkownika	59
10.3	Pozycjonowanie	62
10.4	Trace	63
11.	Położenia krańcowe osi	65
11.1	Ograniczenie programowe	65
11.2	Ograniczenie sprzętowe	66
12.	Bazowanie systemu	67
12.1	Funkcje interfejsu użytkownika	67
12.2	Status wybazowania systemu	68
12.3	Metody bazowania systemu	69
13.	Pozycjonowanie absolutne	76
13.1	Warunki dopuszczające	76
13.2	Funkcje użytkownika	76
IV.	Ruch skoordynowany	79
14.	Opis systemu	79
15.	Sprzężenie relatywne osi	80
15.1	Obiekt technologiczny TO_Cam	80
15.2	Konfiguracja w TIA Portal	81
16.	Sprzężenie absolutne osi	86
16.1	Synchronizacja absolutna	86
16.2	Konfiguracja w TIA Portal	87
17.	Przesunięcie w ruchu synchronicznym	93
17.1	Przesunięcie fazowe	93
17.2	Przesunięcie relatywne nałożone	95
18.	Sprzężenie do wartości aktualnej oraz zadanej	96
V.	Sprzężenie krzywkowe	103
19.	Mechanizm krzywkowy	103
20.	Krzywki w SIMATIC S7-1500T	105
20.1	Zastosowanie krzywki elektronicznej	105
20.2	Obiekt technologiczny TO_Cam	106
20.3	Edytor krzywek	108
20.4	Interfejs użytkownika	118
20.5	Edycja krzywek w programie PLC	123
VI.	Wejścia pomiarowe wyjścia krzywkowe	129
21.	Moduły technologiczne	129
22.	Wejścia/wyjścia typu time-based	132
22.1	Detekcja sygnałów	136
22.2	Wysterowanie wyjść	138
22.3	Przykładowe aplikacje	140

23. Programowanie w TIA Portal	142
23.1 Konfiguracja manualna	142
23.2 Wejście pomiarowe – obiekt technologiczny	143
23.3 Wyjście krzywkowe – obiekt technologiczny	150
23.4 Tor krzywkowy – Cam Track	156
 VII. Bezpieczeństwo w układach Motion Control	
24. Programowanie w TIA Portal	159
25. Wskaźniki bezpieczeństwa	161
26. Realizacja funkcji bezpieczeństwa	163
26.1 Układ hard-wired	164
26.2 Przekaznik bezpieczeństwa	165
26.3 Zintegrowane funkcje bezpieczeństwa	168
26.4 Sterownik <i>fail-safe</i> (F-CPU)	170
27. Funkcje bezpieczeństwa Motion Control	172
27.1 STO – <i>Safe Torque Off</i>	172
27.2 SS1 – <i>Safe Stop 1</i>	173
27.3 SS2 – <i>Safe Stop 2</i>	174
27.4 SOS – <i>Safe Operating Stop</i>	174
27.5 SLS – <i>Safely-Limited Speed</i>	175
27.6 SSM – <i>Safe Speed Monitor</i>	176
27.7 SDI – <i>Safe Direction</i>	176
27.8 SBT – <i>Safe Brake Test</i>	177
27.9 SLP – <i>Safely-Limited Position</i>	177
27.10 SP – <i>Safe Position</i>	178
27.11 Funkcje zintegrowane napędów SINAMICS	178
28. Konfiguracja F-PLC w TIA Portal	179
28.1 Klasyczny układ elektryczny	179
28.1.1 Cechy projektu Safety	179
28.1.2 Ustawienia sprzętowe (<i>Device configuration</i>)	181
28.1.3 Program Safety	184
28.2 Komunikacja <i>PROFIsafe</i>	191
28.2.1 Konfiguracja w TIA Portal Startdrive	193
28.2.1 Konfiguracja przez Starter	197

I. Zagadnienia podstawowe

W pierwszej sekcji zajmiemy się omówieniem podstawowych zagadnień związanych ze sterowaniem ruchem czyli problematyką *Motion Control*. Pierwszym zagadnieniem, które zostanie krótko omówione jest struktura sterowania systemami napędowymi.

W zależności od wymagań aplikacji napędowej, sterowanie – zadawanie prędkości, pozycjonowanie, bazowanie systemu, etc. możemy realizować po stronie samego napędu lub wysyłać komendy przez nadrzędny system sterowania.

Na początku zastanówmy się jakie są cechy tych dwóch struktur systemu w odniesieniu do oczekiwań projektowych, możliwości konfiguracyjnych czy funkcjonalności samej aplikacji.

1 Sterowanie napędem

1.1 Sterowanie lokalne (napęd)

Zastosowanie odpowiedniej klasy napędu SINAMICS daje różne możliwości w zakresie sterowania lokalnego. Przy prostych napędach dla silników indukcyjnych możemy uruchomić napęd z poziomu panelu sterowniczego i sterować pojedynczą osią z odpowiednimi parametrami ruchu. Przy bardziej zaawansowanych napędach możemy zmieniać większą ilość parametrów oraz nadać ich pracy prostą logikę.

W przypadku napędów serwo pojawia się nieco więcej możliwości – już w prostych rozwiązaniach możemy zaprogramować bezpośrednio w napędzie chociażby funkcje sterowania prędkością, pozycjonowania czy bazowania systemu. Serwonapędy z wyższej półki pozwolą na stworzenie zaawansowanej logiki, wręcz programu sterowania przez dedykowane napędom narzędzia inżynierskie.

Wiemy zatem, że zależnie od wybranego rozwiązania napędowego istnieją mniej lub bardziej zaawansowane możliwości sterownia lokalnego. Pytanie zatem, jakie zalety bądź wady niesie za sobą takie podejście do realizacji funkcji sterowania napędami.

Przede wszystkim należy myśleć o aplikacji, którą realizujemy. Trudno będzie mówić o wadach sterowania lokalnego przy zastosowaniu np. napędu SINAMICS V20 w piekarni do wyrabiania ciasta. Proste rozwiązanie do prostej aplikacji – niski koszt napędu, prosta obsługa lokalna, brak konieczności tworzenia jakiegokolwiek logiki programu.

Przy bardziej zaawansowanych aplikacjach gdzie będziemy musieli zrealizować funkcje serwo – przykładowo synchronizm osi czy też sprzężenia krzywkowe – możemy rozważyć napęd SINAMICS S120, który pozwoli wykonać całą logikę wewnątrz napędu przez język programowania DCC (Drive Control Chart). Aplikacje typu piła latająca, nóż obrotowy czy obsługa prasy są do zrealizowania korzystając jedynie z tego napędu.

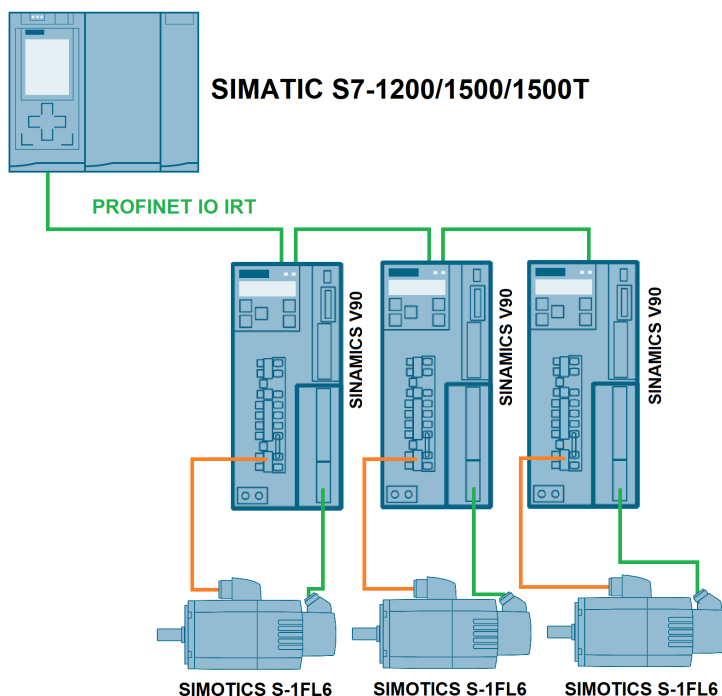
W takim rozwiązaniu zyskujemy z pewnością dużą dokładność ze względu na to, że realizacja funkcji pozycjonowania w napędzie działa bardzo szybko i nie wymaga wymiany informacji przez magistralę komunikacyjną z systemem nadrzędnym. Z drugiej jednak strony wymagana jest znajomość dedykowanego (licencjonowanego) środowiska inżynierskiego, koszt napędu jest dość wysoki, a obsługa wielu osi możliwa jest tylko przez napędy z górnej półki. Może okazać się, że zastosowanie prostszych napędów ze sterownikiem nadrzędnym będzie wygodniejsze oraz tańsze.

Dochodzą tutaj również kwestie diagnostyki, które są ograniczone w porównaniu do rozwiązania ze sterowaniem scentralizowanym (np. ograniczony dostęp do wartości zmiennych). Można sobie również wyobrazić sytuację gdzie napęd ulega uszkodzeniu, a wraz z nim całość projektu oraz nastaw. Przy rozwiązaniu lokalnym wymagane będzie odtworzenie oraz parametryzacja napędu od początku, przy strukturze scentralizowanej wszystkie informacje przechowywane są w sterowniku PLC.

1.2 Struktura scentralizowana (PLC)

Zastosowanie systemu nadrzędnego daje wiele zalet ale w przypadku sporej części aplikacji będzie po prostu zbędnym kosztem lub okaże się przerostem funkcjonalnym. Z pewnością przy wspomnianych w poprzednim rozdziale prostych aplikacjach gdzie logika sterownia nie jest wymagana – nie ma sensu stosować nadrzędnego sterownika. W przypadku aplikacji bardziej zaawansowanych gdzie występuje sterowanie zależne wieloma silnikami indukcyjnymi lub mechanizmami serwo – można już rozważać system nadrzędny. Zdecydowana większość takich aplikacji realizowana jest właśnie w takiej strukturze.

Zyskujemy tutaj przede wszystkim transparentność systemu – konfiguracja napędów w spójnym środowisku inżynierskim razem z programem sterowania oraz urządzeniami HMI daje bardzo dużą elastyczność, standard programistyczny oraz zdecydowanie więcej możliwości – zarówno jeśli chodzi o kwestie funkcyjne jak i diagnostyczne. Warto również wspomnieć, iż systemy zintegrowane *Motion Control* (SIMATIC + SINAMICS) oprogramowane są przez tzw. obiekty technologiczne czyli przez zmapowanie fizycznego obiektu do sterownika. Jest to bardzo wygodny mechanizm, gdyż wszystkie parametry oraz funkcje sterujące realizujemy na odwzorowanym w PLC napędzie z uwzględnieniem zastosowanej mechaniki. Wszystkie etapy konfiguracji wykonujemy tutaj w jednym spójnym środowisku inżynierskim. Więcej o obiektach technologicznych w dalszej części dokumentu. Prostota obsługi oraz centralna lokalizacja konfiguracji systemu jest również kluczowa we wspomnianym wcześniej przypadku krytycznym, gdzie napęd ulega awarii.



Zadania napędów serwo jesteśmy w stanie zrealizować na systemach z niższej półki, zyskując taką samą jakość i dokładność. Można ewentualnie zastanowić się nad mechanizmami pozycjonowania, gdzie najwyższą dokładność oraz prędkość uzyskamy realizując pętlę kontroli pozycji bezpośrednio w napędzie. Biorąc jednak pod uwagę obsługę trybu izochronicznego (IRT, o którym więcej w dalszej części) – przykładowo w systemie, gdzie masterem będzie jednostka centralna z rodziny S7-1500, natomiast napędem SINAMICS V90 – okazuje się, że różnica w dokładności ma znaczenie tylko w niewielkiej ilości bardzo specyficznych aplikacji.

Przy aplikacjach niewymagających wysokiej dokładności możemy zastosować inne magistrale komunikacyjne (np. Modbus lub USS) lub wydawać polecenia bezpośrednio przez sygnały I/O zintegrowane napędzie, co zminimalizuje koszt aplikacji.

Podsumowując można więc stwierdzić, iż wybór struktury systemu zawsze zależy od wymogów aplikacji. W zdecydowanej jednak większości przypadków wygodniej, a być może i taniej będzie zrealizować aplikację w strukturze scentralizowanej. Taka struktura daje największą elastyczność oraz możliwości integracji systemów napędowych różnej klasy. Obsługa całościowa projektu staje się prosta począwszy od konfiguracji, przez programowanie, a kończąc na diagnostyce.

2 Komunikacja napęd–sterownik

2.1 Protokoły komunikacyjne

Nasza struktura systemu proponuje napęd zarządzany przez kontroler nadrzędny. Zastanówmy się, jakie mamy do dyspozycji metody wymiany informacji pomiędzy jednostką nadrzędną a napędem.

Podchodząc do zagadnienia komunikacji w układzie napęd–sterownik bardzo ogólnie, mamy do dyspozycji magistralę komunikacyjną lub w przypadkach uproszczonych – komunikację przez sygnały I/O. W tym drugim wariantcie istnieje możliwość połączenia napędu z systemem sterowania interfejsem elektrycznym – gdzie polecenia do napędu wydawane będą przez sygnały aktywujące.

Jeśli jednak system będzie wyposażony w standard komunikacyjny w postaci protokołu wymiany danych najczęściej spotkamy:

- PROFIBUS DP,
- PROFINET,
- EtherNet/IP,
- USS,
- Modbus RTU,
- CANopen.

W zależności od wymagań aplikacji – dostępnego interfejsu, oczekiwanej prędkości wymiany danych czy norm zakładowych – dobieramy konfigurację interfejsów PLC oraz napędu. Najbardziej pożądanym standardem, zwłaszcza w przypadku aplikacji szybkich, jest PROFINET. Standard jest nie tylko najbardziej wydajną drogą wymiany danych ale również jest bardzo łatwy w konfiguracji oraz obsłudze.

Dodatkowo w przypadku aplikacji, gdzie prędkość oraz synchronizacja odczytu/zapisu informacji jest kluczowa, wspiera on mechanizmy deterministyczne PROFINET RT (real-time) oraz IRT (isochronous real-time).

W zależności od protokołu komunikacyjnego sterownik będzie wysyłał do, oraz odbierał informacje z napędu przez tzw. słowa sterujące oraz statusowe.

W większości aplikacji komunikujących się przez PROFINET lub PROFIBUS kwestia przesyłanych wartości parametrów będzie sprowadzała się do ustandaryzowanych struktur wymiany informacji – napędy oraz enkodery komunikują się przez tzw. ramki PROFIdrive (niezależny standard opracowany przez organizację PI International – www.profibus.com – dla sieci PROFINET IO oraz PROFIBUS DP).

2.2 Telegramy PROFIDrive

Sterownik wysyła do napędu komendy sterujące, wartości zadane oraz inne parametry określone przez funkcje programu użytkownika. Napęd z kolei zwrótnie dostarcza m.in. informacji statusowych oraz wartości aktualnego położenia.

W zależności od konfiguracji systemu oraz ilości informacji potrzebnych w programie użytkownika dobierzemy odpowiednią ramkę komunikacyjną PROFIDrive.

Poniższa tabela prezentuje charakterystykę doboru odpowiedniej ramki komunikacyjnej. Najczęściej stosowane to telegramy standardowe 1, 2, 3 oraz 5:

PZD: Słowo danych procesowych									
Telegram	PZD 1	PZD 2	PZD 3	PZD 4	PZD 5	PZD 6	PZD 7	PZD 8	PZD 9
1 Prędkość zadana 16 bit	STW 1	NSOLL							
	ZSW 1	NIST							
2 Prędkość zadana 32 bit, bit życia	STW 1	NSOLL		STW 2					
	ZSW 1	NIST		ZSW 2					
3 Prędkość zadana, bit życia, pozycja enkodera	STW 1	NSOLL		STW 2	G1_STW				
	ZSW 1	NIST		ZSW 2	G1_ZSW	G1_XIST1		G1_XIST2	
5 Prędkość zadana, bit życia, pozycja enk., DSC	STW 1	NSOLL		STW 2	G1_STW	XERR		KPC	
	ZSW 1	NIST		ZSW 2	G1_ZSW	G1_XIST1		G1_XIST2	

STW: Słowo sterujące
ZSW: Słowo statusowe

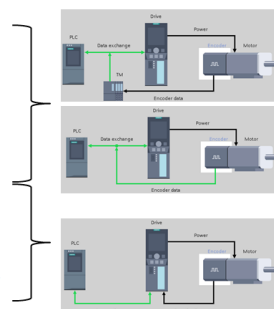
NSOLL: Prędkość zadana
NIST: Prędkość aktualna

G1_STW: Słowo sterujące Enkodera 1
G1_ZSW: Słowo statusowe Enkodera 1
G1_XIST: Pozycja aktualna Enkodera 1

XERR: Odchylenie pozycji
KPC: Wzmocnienie regulatora pozycji

Przenosząc powyższe informacje na strukturę systemu, możemy stwierdzić powiązania pomiędzy konfiguracją a wyborem odpowiedniego telegramu – zgodnie z ilustracją poniżej:

- Telegram 1:
- sterowanie prędkością osi,
 - oś pozycjonująca, jeśli enkoder podłączony jest przez zewnętrzny moduł technologiczny (TM) lub przez PROFIDrive
 - napęd bez synchronizacji do zegara czasu rzeczywistego
- Telegram 2:
- podobnie jak Telegram 1, dodatkowo synchronizacja napędu do zegara czasu rzeczywistego
- Telegram 3:
- oś pozycjonująca, jeśli enkoder podłączony jest do napędu
 - możliwa synchronizacja napędu w czasie rzeczywistym
- Telegram 5:
- oś pozycjonująca, jeśli S120/V90 w trybie *Dynamic Servo Control*
 - wymagana synchronizacja napędu w czasie rzeczywistym



Siemens proponuje do swoich systemów Motion Control rozszerzone telegramy PROFIDrive. Przykładowo, standardowe telegramy 3/4/5 mają odnośnik w formie telegramów Siemens 102/103/105. Różnica polega na tym, że telegramy rozbudowane są o dwa dodatkowe bajty przewidziane na funkcje monitorowania siły/momentu obrotowego, np. przy realizacji zadania Fixed stop detection, które pozwala na bezbłędne zatrzymanie obrotu osi w momencie wykrycia określonego poziomu momentu obrotowego.

2.3 Tryb izochroniczny czasu rzeczywistego (IRT)

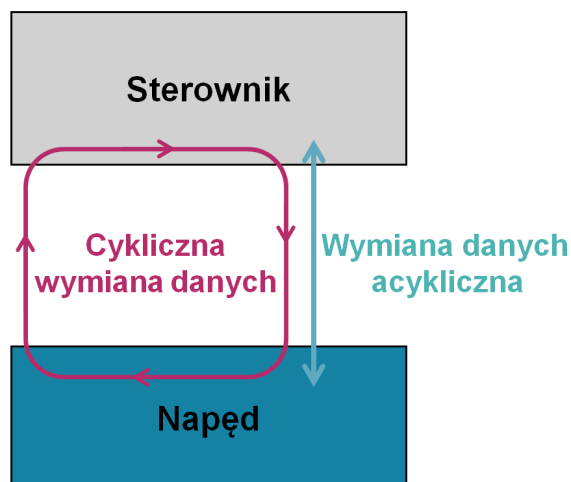
Wymiana informacji pomiędzy napędem a sterownikiem nadrzędnym odbywa się dwoma kanałami.

Kanał cykliczny

Zachodzi tutaj izochroniczna wymiana danych procesowych przez ramki *PROFIDrive* (obiekt technologiczny) – sterownik wysyła słowa sterujące oraz wartości zadane, natomiast od napędu odbiera słowa statusowe oraz wartości aktualne. Wymiana małych ilości informacji w jednym cyklu programu.

Kanał acykliczny

Zapewnia dostęp do parametrów napędu (odczyt/zapis) przez środowisko inżynierskie lub przez funkcje z biblioteki systemowej (*LAcycCom*). W tym wypadku wymiana dużej ilości informacji przez kilka cykli programu.



Wymiana danych w trybie Real-Time (RT)

W zakresie sieci PROFINET IO komunikacja (wymiana danych procesowych czy komunikatów alarmowych) zawsze odbywa się w trybie czasu rzeczywistego.

Tryb Real-Time pozwala na przyspieszenie wymiany informacji pomiędzy urządzeniami produkcyjnymi dzięki odrębnej warstwie komunikacyjnej – niezależnej od protokołów internetowych czy UDP(TCP)/IP – dane RT mają wyższy priorytet. Komunikacja urządzeń rozproszonych w trybie RT zapewnia deterministyczną wymianę istotnych informacji (odczyt i zapis informacji w ściśle określonym czasie) w cyklach kilku-milisekundowych.

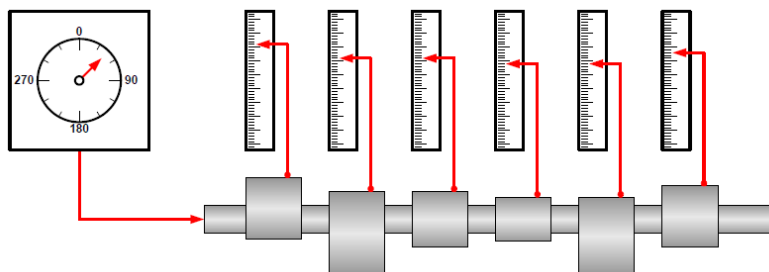
Krokiem dalej jest zastosowanie trybu izochronicznego Real-Time co daje jeszcze więcej możliwości w zakresie prędkości komunikacyjnej dla danych procesowych, a także zaawansowanych mechanizmów harmonogramu wymiany informacji przez zarezerwowane pasma komunikacyjne.

Tryb IRT

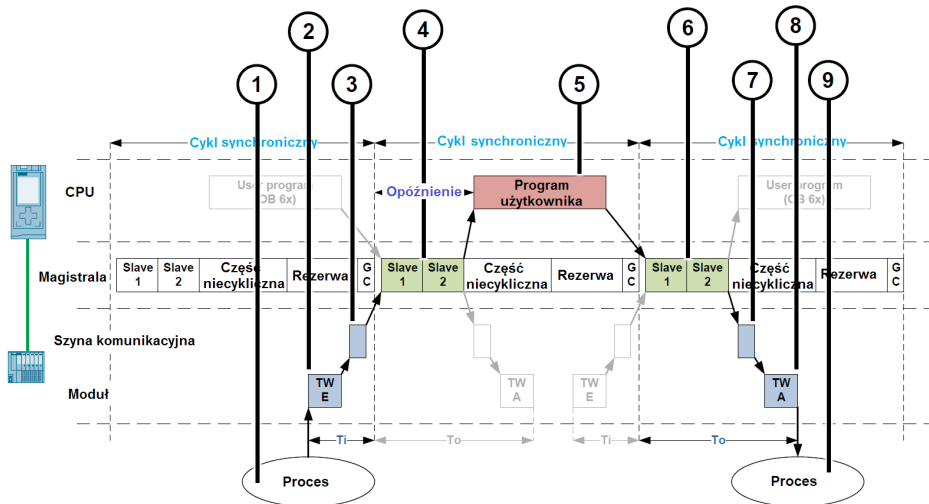
W sieci PROFINET IO oraz PROFIBUS DP istnieje możliwość konfiguracji trybu izochronicznego, czyli wymiany danych w czasie rzeczywistym – o ściśle określonym i wysoce powtarzalnym czasie cyklu. W wielu aplikacjach krytycznych czasowo konieczne jest aby parametry zależne względem siebie, a pochodzące z różnych źródeł zostały przetworzone w jednym cyklu programu sterownika. Przykładowo:

- równoczesny odczyt wielu wartości pomiarowych z niezależnych modułów wejść,
- wykonanie programu sterowania w sterowniku S7-1500, gdzie odczyt wartości mierzonych oraz sygnały wyjściowe pochodzą z różnych modułów,
- skoordynowane sterowanie napędami przez funkcje systemowe sterownika S7-1500.

Przykładem może być system precyzyjnego mierzenia wartości pozycji na wałku krzywkowym – gdzie mierzone odchylenia na poszczególnych krzywkach oraz kąt obrotu wału muszą być odczytane przez różne moduły w tym samym czasie i przetworzone/porównane w tym samym cyklu programu.



Zastosowanie trybu izochronicznego w sterowniku SIMATIC pozwoli na odczyt wartości w tym samym czasie. Rozproszone moduły I/O zbiorą informacje w tym samym momencie i dostarczą je do nadrzędnej jednostki CPU w określonym przedziale czasu. Odpowiedź systemu na otrzymane informacje zostanie wysłana w podobny sposób. Poniższy diagram pokazuje mechanizm przesyłania informacji w trybie izochronicznym.



Krok Opis

- 1 Wartość pomiaru zostaje odczytana z procesu.
- 2 Pomiar zostaje przetworzony na formę cyfrową w module wejściowym.
- 3 Cyfrowa wartość pomiaru zostaje przetransferowana z modułu wejść do modułu interfejsu przez magistralę wewnętrzną wyspy I/O.
- 4 Wartość zmierzona zostaje przesłana na początku cyklu magistrali komunikacyjnej do jednostki centralnej CPU SIMATIC w trybie izochronicznym.
- 5 Po otrzymaniu danych, CPU SIMATIC uruchamia synchroniczny blok OB, w którym otrzymana wartość zostanie przetworzona.
- 6 Wynik przetwarzania pomiaru lub reakcja na otrzymaną wartość zostaje przesłana na początku cyklu magistrali do modułów rozproszonych wejść/wyjść w trybie izochronicznym.
- 7 Sygnał wyjściowy zostaje przesłany z modułu interfejsu do modułu wyjść.
- 8 Wartość wyjściowa zostaje przetworzona – przykładowo konwersja na wartość analogową.
- 9 Wartość sygnału zostaje przekazana do procesu.

Izochroniczna wymiana danych daje pewność, że wszystkie wartości pomiarowe zostaną odczytane z procesu w określonym czasie T_i zanim zostaną przetransferowane przez magistralę komunikacyjną, oraz w zdefiniowanym czasie T_o zostaną przesłane do procesu po transferze zwrotnym.

Czasy T_i oraz T_o mogą zostać obliczone przez TIA Portal automatycznie w zależności od liczby modułów izochronicznych w każdej ze stacji rozproszonych IO. Czasy te można również ustawić statycznie dla każdej ze stacji pomiarowych, co jest zdecydowanie zalecane w przypadku gdy wartości pomiarowe pochodzą z wielu źródeł.

Uwaga

W przypadku sterowników z rodziny S7-1500 tryb izochroniczny wspierany jest przez jednostkę centralną w połączeniu z system rozproszonych wejść/wejść ET 200. Centralna konfiguracja sterownika S7-1500 (moduły I/O podłączone bezpośrednio do CPU) nie pozwala na konfigurację trybu izochronicznego – nie wspiera go wewnętrzna magistrala komunikacyjna sterowników S7-1500.

Moduły rozproszonych systemów I/O mogą być podłączone do CPU sterownika nadrzędnego przez PROFIBUS lub PROFINET. Trzeba się również upewnić, iż moduł interfejsu stacji rozproszonej wspiera tryb IRT.

Należy także pamiętać, iż w celu poprawnej konfiguracji sieci PROFINET IO pracującej w trybie IRT, komponenty pośredniczące w wymianie informacji również muszą obsługiwać komunikację izochroniczną. Zwłaszcza istotne jest to w przypadku switchy ethernetowych, które są standardowymi komponentami konfiguracji sieci przemysłowych. W zakresie rodziny SCALANCE znajdziemy wiele komponentów, które możemy zastosować w sieci PROFINET IO IRT, np. seria XB200 IRT.

3 Typowa konfiguracja systemu Motion Control ze sterowaniem nadrzędnym

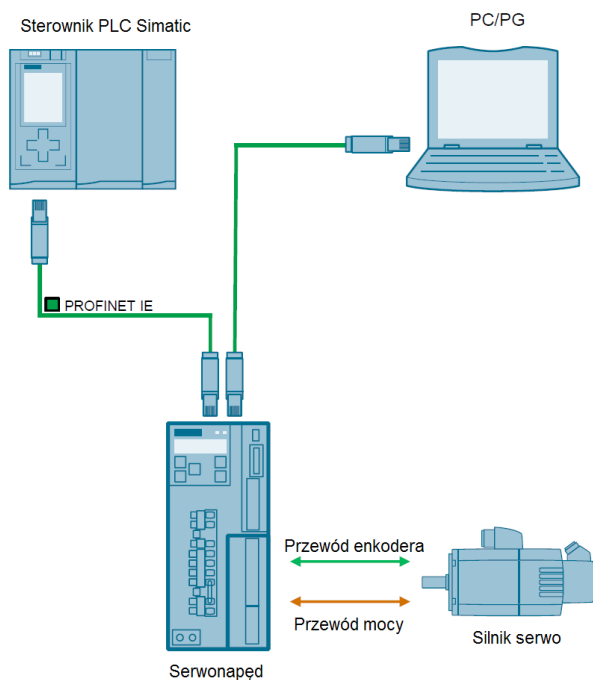
Bazą sprzętową proponowaną w niniejszej dokumentacji będzie sterownik z rodziny SIMATIC S7-1500(T), serwonapęd SINAMICS V90 oraz silnik serwo SIMOTICS.

Struktura systemu

Dostępne interfejsy komunikacyjne sterownika oraz napędu pozwolą nam na komunikację w zakresie całego systemu przez PROFINET, co ułatwi również podłączenie komputera/programatora.

Sterownik będzie zatem w naszej strukturze urządzeniem nadrzędnym – masterem (I/O Controller), natomiast napęd będzie urządzeniem wykonawczym – slave (I/O Device). PLC będzie przez program użytkownika wydawał polecenia do napędu.

Serwonapęd z kolei będzie sterował bezpośrednio pracą silnika ze sprzężeniem zwrotnym do napędu w postaci enkodera pozycji.



4

S7 Motion Control

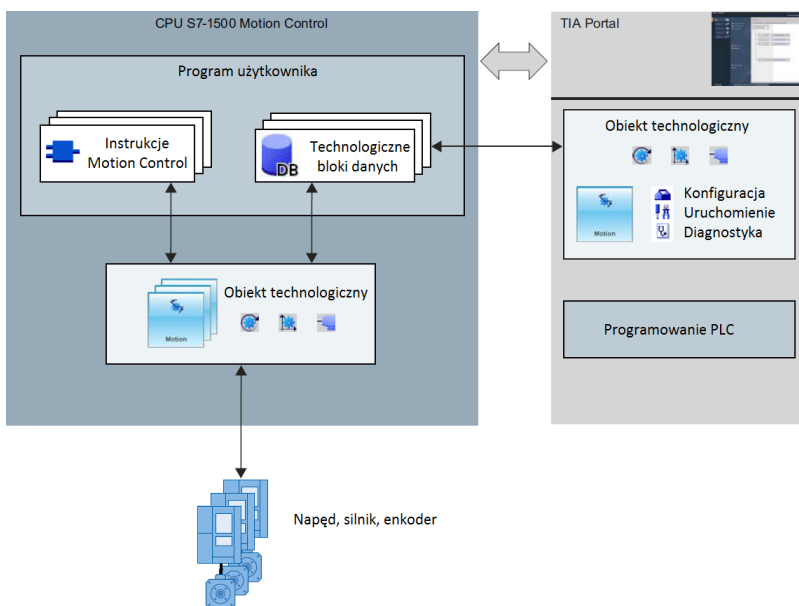
4.1 Zasada działania

Sterowanie systemem ruchu w środowisku SIMATIC sprowadza się do zaprogramowania oraz wysyłania komend sterujących (programu użytkownika) z poziomu sterownika PLC do napędu.

Fizyczna konfiguracja napędu (falownika, silnika oraz enkodera) mapowana jest poprzez narzędzie inżynierskie do sterownika przez tzw. obiekt technologiczny.

Konfiguracja obiektu technologicznego wykonana jest w środowisku TIA Portal, a po jej wgraniu – przetwarzana w jednostce centralnej PLC. Obiekt technologiczny sterowany jest instrukcjami (PLC) z grupy Motion Control. Dodatkowo TIA Portal posiada dodatkowe funkcje konfiguracyjne oraz uruchomieniowe, które umożliwiają optymalizację oraz diagnostykę napędu.

Konceptję sterowania przedstawia poniższy schemat:



4.2 TIA Portal

Środowisko inżynierskie pozwala na kompleksową obsługę systemu sterowania. Funkcjonalność *Motion Control* daje możliwość:

- integracji oraz konfiguracji komponentów sprzętowych,
- tworzenia oraz parametryzacji obiektów technologicznych,
- zaprogramowania logiki pracy systemu (program użytkownika),
- diagnostyki komponentów.

Instrukcje Motion Control

Dzięki instrukcjom *Motion Control* aktywujemy potrzebne funkcje obiektów technologicznych. Grupa instrukcji znajduje się w przyborniku narzędziowym w zakładce *Instructions* → *Technology* → *Motion Control*.

Instrukcje te są zgodne ze standardem PLCOpen (v2.0) – www.plcopen.org. Dzięki znormalizowanej konstrukcji bloków funkcyjnych zyskujemy wydajność oraz efektywność podczas tworzenia aplikacji.

Obiekt technologiczny

Odwzorowanie fizycznego urządzenia (np. napędu) w sterowniku odbywa się przez obiekty technologiczne.

Funkcje obiektu technologicznego wywoływane są przez instrukcje *Motion Control* w programie użytkownika. Obiekty technologiczne pozwalają na sterowanie ruchem zarówno w otwartej pętli programowej jak i ze sprzężeniem zwrotnym (*open-*, *closed-loop control*) dostarczając równocześnie informacji zwrotnej z obiektu (np. aktualna pozycja).

Konfiguracja obiektu technologicznego odzwierciedla właściwości fizycznego urządzenia. Dane konfiguracyjne przechowywane są w blokach danych obiektu technologicznego. Główne obiekty technologiczne w zakresie *Motion Control*:

- **Speed axis**  Obiekt technologiczny pozwala na zadanie prędkości do napędu. Przeszczenie się wywołane jest przez instrukcje programowe *Motion Control*.
- **Positioning axis**  Obiekt technologiczny pozwala na pozycjonowanie napędu. Zadania pozycjonowania realizowane są w programie użytkownika.
- **Synchronous axis**  Obiekt technologiczny realizujący zadania synchronizacji osi – tzw. gearing. Może być to synchronizacja prędkości lub prędkości i pozycji czy też inne zadania związane ze sprzężeniem osi – przesunięcie osi względem siebie, sprzężenie krzywkowe czy symulacja osi wirtualnej.
- **External encoder**  Obiekt technologiczny pozwana na detekcję pozycji oraz przekazanie jej wartości do programu użytkownika.

Technologiczne bloki danych

Bloki danych obiektów technologicznych reprezentują parametry obiektu technologicznego, a także zawierają wszelkie dane konfiguracyjne, wartości zadane, wartości aktualne czy też informacje statusowe obiektu. Technologiczny blok danych tworzony jest automatycznie przez system podczas dodawania do projektu obiektu technologicznego. Zawartość bloku danych dostępna jest z poziomu programu użytkownika.

Program użytkownika

Motion Control oraz technologiczne bloki danych tworzą interfejs programistyczny obiektu technologicznego. Wykorzystanie instrukcji Motion Control w programie sterownika pozwalają zainicjalizować oraz śledzić ścieżkę zadań sterowania napędem przez obiekty technologiczne.

Napęd

Napęd jest urządzeniem wykonawczym zwanym również akuatorem. Pozwala na obracanie osi. Napędy zintegrowane są w konfiguracji systemu jako urządzenia podrzędne (slave). Wydając polecenie Motion Control w programie użytkownika obiekt technologiczny przejmuje kontrolę nad napędem oraz odczytuje pozycję z enkodera.

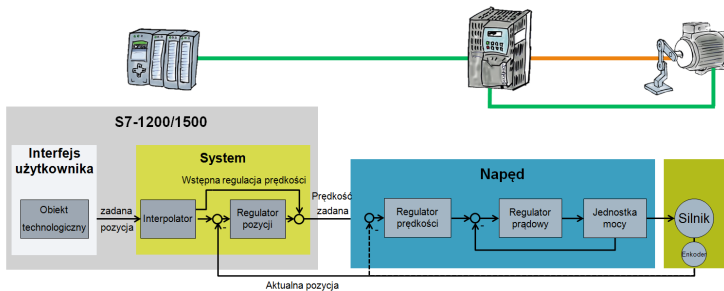
5 Pozycjonowanie

5.1 Struktura systemu

Pozycjonowanie

Poza zadawaniem prędkości do napędu – podstawowym zagadnieniem w systemach z serwonapędami jest precyzyjne pozycjonowanie, czyli przemieszczenie osi na określoną pozycję lub o określoną liczbę wyznaczonych jednostek.

Poniższy schemat ilustruje mechanizm wewnętrzny generowania wartości zadanych do pozycjonującego systemu napędowego z uwzględnieniem analizy sprzężenia zwrotnego z enkodera.



Interfejs użytkownika

Na część programowalną składa się tutaj wspomniana już wcześniej logika programu oraz konfiguracja obiektu technologicznego. Dalsza część zadań wykonywana jest przez system w cyklicznym bloku przerzaniowym.



Program użytkownika

Logika programu oraz komendy *Motion Control*



Obiekt technologiczny

Charakterystyka oraz konfiguracja aplikacji *Motion Control*



System – bloki OB – MC

Wykonanie poleceń *Motion Control*, interpolacja, pozycjonowanie

Bloki OB – MC wywoływane są w zadanym cyklu = cykl MC = cykl regulatora pozycji = cykl interpolacji (lub jako isochroniczne przerwanie cykliczne).

Napęd

Przekształtnik częstotliwości wykonuje regulację prędkości oraz poprzez moduł mocy przekazuje odpowiednią wartość natężenia prądu na zaciski silnika. Wartość pozycji aktualnej, odczytywana z enkodera, zwracana jest w sprzężeniu zwrotnym pętli sterującej.

5.2 Interpolator

W programie użytkownika generujemy wartość pozycji, na którą chcemy przemieścić obiekt technologiczny (oś pozycjonująca/synchroniczna) lub odległość, o jaką ma zostać on przesunięty – pozycjonowanie relatywne lub absolutne. Funkcja *Motion Control* określa również parametry przemieszczenia takie jak prędkość, przyspieszenie czy opóźnienie.

Aby zadane przemieszczenie osi przebiegło z odpowiednią dynamiką – płynnie, a zarazem precyzyjnie – system musi wygenerować zestaw wartości zadanych pozycji na całą „zaplanowaną” trasę. Innymi słowy, jeśli chcemy przemieścić się z pozycji startowej na pozycję docelową, to należy obliczyć (interpolować) wartości pośrednie, które będą w każdym kolejnym cyklu zadawane do napędu już podczas jego pracy.

Zadanie to wykonuje interpolator, czyli pierwszy element wewnętrznego przetwarzania wartości zadanej, w systemie sterowania nadrzędnego.

5.3 Regulator pozycji

Chwilowa wartość zadana pozycji – będąca wynikiem kalkulacji interpolatora – zostaje następnie przekazana do regulatora położenia.

Element ten odpowiada za wygenerowanie w każdym cyklu prędkości zadanej i przekazanie jej do napędu (regulator pozycji – w przypadku aplikacji o skrajnie wysokiej dynamice – może zostać przeniesiony do napędu).

Prędkość zadana jest wynikiem różnicy pozycji obliczonej przez interpolator oraz pozycji aktualnej otrzymanej z enkodera, pomnożonej przez wzmocnienie regulatora. W odniesieniu do zagadnień regulacji w automatyce – mamy tutaj do czynienia z regulatorem typu P – proporcjonalnym (z dodatkową matematyczną pre-regulacją prędkości przez interpolator). Wspomniane wzmocnienie jest więc standardowym parametrem regulatora P (zamkniętej pętli sterującej), które prowadzi do określenia znormalizowanego zakresu wyjściowej wartości układu – w naszym przypadku prędkości zadanej.

Wartość wzmocnienia (K_v) powinna zostać dostrojona do układu mechanicznego podczas uruchomienia systemu. Parametr wpływa na płynność ruchu, a także dokładność oraz czas pozycjonowania.

Wynikowa wartość prędkości zadanej regulatora położenia jest dodatkowo kompensowana przez wynik obliczeń modelu matematycznego prędkości zadanej zawartego w interpolatorze. Zadaniem regulacji wstępnej jest zwiększenie dynamiki układu oraz minimalizacja błędów nadążania.

5.4 Regulator prędkości

Ostatnim etapem przetwarzania parametrów zadanych jest regulator prędkości. Ta część układu realizowana jest po stronie napędu. W tym kroku następuje wiele obliczeń oraz analiz wartości parametrów aktualnych, których wynikiem jest sterowanie silnikiem w taki sposób aby moment obrotowy był znamionowy, natomiast uchyb prędkości jak najmniejszy. Analizie podlegają prądy, moment obrotowy, przesunięcie faz, etc. Regulacja odbywa się przy pomocy regulatora PI oraz zaawansowanego modelu matematycznego zastosowanego silnika. Analiza szczegółowego zachowania regulatora prędkości jest bardzo rozległym tematem – nie będziemy zgłębiać tych detali w niniejszym dokumencie.

II. Układ pomiarowy

W niniejszym dokumencie zajmiemy się omówieniem układu pomiarowego serwonapędu. Pomiar pozycji oraz – wynikających z jej zmiany w czasie – parametrów ruchu najczęściej odbywa się przez enkoder cyfrowy. Sporadycznie stosuje się inne elementy pomiarowe, np. resolver (analogowy odpowiednik enkodera) czy tachoprądnica (pomiar prędkości obrotowej). W zależności od konfiguracji struktury systemu oraz wymogów aplikacji, enkoder podłączony zostanie do przekształtnika częstotliwości lub (w strukturze scentralizowanej) bezpośrednio do sterownika PLC. Oczekiwana rozdzielczość pomiaru oraz jego zakres, dostępność wartości mierzonej w przypadku zaniku napięcia zasilania, a także typ komunikacji – determinować będą wybór odpowiedniego typu enkodera.

W dalszej części zajmiemy się kategoryzacją oraz omówieniem zasad działania poszczególnych typów enkoderów dostępnych powszechnie na rynku, a także możliwościami podłączenia ich do systemu sterowania opartego o sterownik SIMATIC S7-1500T oraz napęd SINAMICS.

Enkoder

Enkoder jest urządzeniem przetwarzającym przesunięcie lub pozycję kątową na sygnał elektryczny. W układach mechanicznych maszyn oraz linii produkcyjnych enkoder pozwala na precyzyjny pomiar prędkości, odległości czy też przebytej drogi z uwzględnieniem kierunku ruchu. Jest to podstawowy (najczęściej stosowany) element układu pomiarowego serwonapędu.

W zależności od zastosowanej technologii, rodzaju zasilania czy możliwości zapamiętania aktualnej pozycji enkodery dzielimy na inkrementalne oraz absolutne.

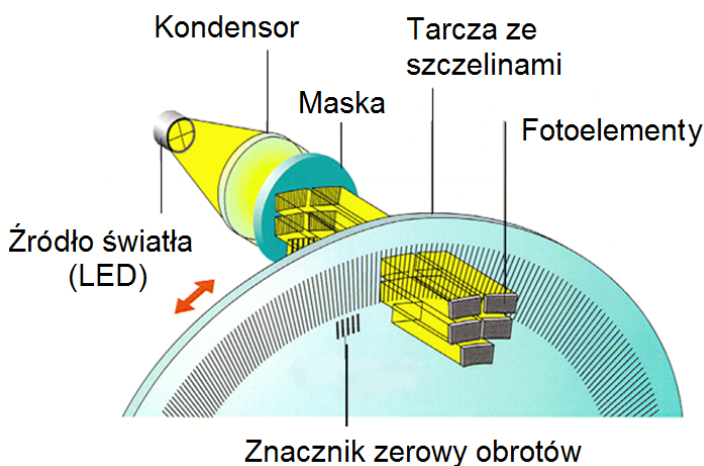
6

Enkodery inkrementalne

Budowa

Enkoder inkrementalny (zwany również przyrostowym lub impulsowym) jest przetwornikiem generującym impulsy elektryczne określające przyrost kątowy w ruchu obrotowym. Cechą charakterystyczną enkodera inkrementalnego jest stała liczba impulsów na wyjściu, która determinuje dokładność układu pomiarowego.

Poniższa ilustracja przedstawia budowę optycznego enkodera inkrementalnego:

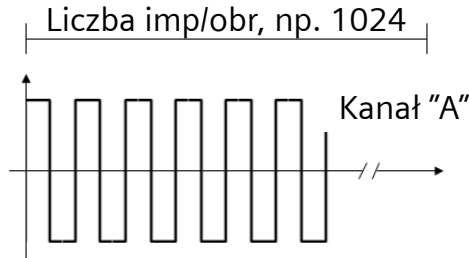


Zasada działania

Impulsy generowane są przez układ mechaniczny tarczy ze szczelinami, przez które przepuszczone jest światło w kierunku elementów światłoczułych. Obecność szczeliny powoduje oświetlenie fotelementu, a co za tym idzie – wygenerowanie napięcia, które interpretowane jest jako sygnał binarny przez układ elektroniczny. Alternatywnie stosowane są enkodery wykorzystujące technologię magnetyczną lub pojemnościową. Wybór typu enkodera uwarunkowany będzie dostosowaniem do warunków pracy urządzenia oraz oczekiwanych parametrów pomiaru.

Sygnały wyjściowe

Zmiana pozycji w ruchu obrotowym powoduje wygenerowanie sygnału elektrycznego, który przesyłany jest do systemu nadrzędnego (napędu lub sterownika), który interpretuje przesłane wartości narastającego zbocza sygnału cyfrowego. Encoder wyposażony jest w przynajmniej jeden kanał wyjściowy przesyłu informacji A, który ma postać binarną jak na poniższym wykresie:

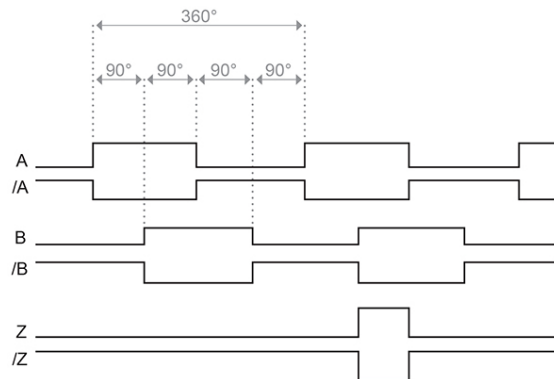


Zazwyczaj jednak kanały wyjściowe są dwa A oraz B – przesunięte względem siebie o 90° , co przez interpretację kolejności pojawiania się sygnałów – pozwala dodatkowo określić kierunek obrotów wału enkodera. W trakcie obrotu wałka enkodera w kierunku zgodnym ze wskazówkami zegara pierwszy pojawi się sygnał A, w kierunku przeciwnym – najpierw odczytane zostanie zbocze sygnału B.

Dodatkowo często spotykamy również trzeci kanał Z, który jest tzw. impulsem zerowym enkodera – generuje on jeden sygnał na pełen obrót w celu zachowania punktu referencyjnego. Wewnętrzny znacznik zerowy wykorzystywany jest również podczas procedury bazowania systemu, o czym szerzej będziemy mówić w kolejnych rozdziałach.

Niektóre modele enkoderów inkrementalnych wyposażone są również w sygnały oznaczane jako $\overline{A}$, $\overline{B}$ i $\overline{Z}$. Są to sygnały będące negacją wyjść kanałów A, B oraz Z. Ich analiza przez układ elektroniczny pozwala na potwierdzenie poprawności komunikacji. W celu redukcji zakłóceń elektrycznych sygnały te zazwyczaj przesyłane są parami ($A+\overline{A}$, $B+\overline{B}$, $Z+\overline{Z}$) przez skrętkę.

Przebieg sygnałów w przypadku bardziej rozbudowanej konfiguracji będzie wyglądał zgodnie z poniższym wykresiem:



Rozdzielczość

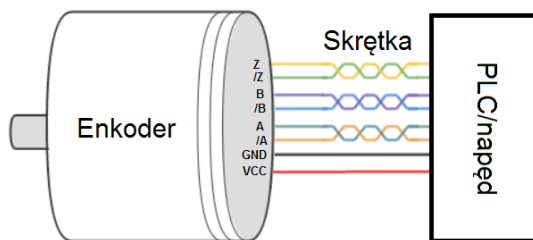
Podstawowym parametrem enkodera inkrementalnego jest jego rozdzielczość. Omówione na poprzedniej stronie impulsy wysyłane do systemu nadrzędnego wytwarzane są dzięki konstrukcji mechanicznej enkodera – bezpośrednio odpowiada za nią liczba szczelin na tarczy wewnętrznej urządzenia. Ilość impulsów generowanych przez układ enkodera na jeden obrót jest ściśle określona dla jego typu. Im większa ilość impulsów, tym dokładniejszy pomiar – mniejsza zmiana pozycji obrotowej powoduje powstanie impulsu wyjściowego. Rozdzielczości enkoderów inkrementalnych zaczynają się od kilkuset impulsów na obrót. Taka dokładność pomiaru zazwyczaj jest wystarczająca – enkodery inkrementalne zazwyczaj stosowane są w układach regulacji prędkości jako impulsatory w układach silników indukcyjnych. Bardziej dokładne pomiary realizowane są w aplikacjach serwo z wykorzystaniem silników synchronicznych – tutaj dokładności enkoderów inkrementalnych sięgają wartości nawet kilkuset tysięcy imp/obrot.

Rozdzielczość układu pomiarowego jest zazwyczaj bezpośrednim przeniesieniem ilości szczelin z wewnętrznej tarczy enkodera. Biorąc jednak pod uwagę możliwości wewnętrznego układu elektronicznego – niektóre enkodery potrafią wykonać programową ewaluację zbocz narastających obu sygnałów (A oraz B), co zwiększy dokładność pomiaru dwukrotnie. Idąc dalej tym tropem – istnieją również enkodery wykonujące analizę zbocz narastających oraz opadających sygnałów A oraz B, co pozwala zwiększyć dokładność elektronicznie czterokrotnie.

Warto również tutaj wspomnieć o enkoderach programowalnych, które – dzięki uniwersalnej konstrukcji oraz interfejsie użytkownika – mogą zostać zaprogramowane w zależności od potrzeb aplikacji. Parametrem programowalnym jest nie tylko ilość impulsów generowanych na obrót wałka ale również typ wyjścia czy kierunek zliczania impulsów.

Transmisja danych

W przypadku enkodera inkrementalnego transmisja danych odbywa się przez wyjścia cyfrowe – niezależnie dla każdego z sygnałów elektrycznych (np. w standardzie RS422). Ze względu na stałą ilość linii sygnałowych oraz przesyłane dane (binarne) okablowanie będzie miało zazwyczaj podobną konfigurację. Standardowo będziemy mieli więc zasilanie (np. 24V, GND) oraz maksymalnie 6 żył do przesyłu sygnałów A, B, Z oraz /A, /B, /Z.



Enkoder inkrementalny nie posiada możliwości zapamiętania pozycji – jest to jego właściwość, która zasadniczo odróżnia go od enkodera absolutnego. W zawiązku z powyższym, zanik napięcia zasilania pociąga za sobą utratę pozycji aktualnej, a co za tym idzie, konieczność bazowania systemu mechanicznego (np. na podstawie sygnału zewnętrznego z wyłącznika krańcowego lub wewnętrznego sygnału zerowego enkodera).

Rodzaj wyjścia

Mechaniczna część enkodera w zależności od zastosowanej technologii (optyczna, pojemnościowa czy magnetyczna) służy konwersji ruchu obrotowego na sygnał impulsowy. Część elektroniczna z kolei to konwerter sygnału na cyfrowy standard komunikacyjny. W przypadku enkoderów inkrementalnych najczęściej spotykane to RS422 (TTL) oraz Push-Pull (HTL).

Cechy enkoderów inkrementalnych

- Stosunkowo niski koszt,
- wysoka rozdzielczość
- nieduże gabaryty,
- wymagane bazowanie system przy pozycjonowaniu absolutnym,
- łatwa wymiana w razie awarii,
- odporność na niekorzystne warunki pracy,
- szybka transmisja danych.

6.1 TTL

Wyjście enkodera w standardzie TTL (RS422) ma wartość napięcia w przybliżeniu 5 VDC (minimalnie 3V dla logicznej „1” oraz maksymalnie 0.5V dla logicznego „0”). Napięcie wyjścia nie jest zależne od poziomu napięcia zasilania. Zasilanie enkodera może być dostarczone w dwóch standardach: 4.75 – 5.5 VDC lub 8 – 30 VDC. Niezależnie od opcji zasilania, wyjście pozostaje zgodnie ze standardem elektrycznym RS422. Wyjście TTL zapewnia stosunkowo wysoką częstotliwość odpowiedzi układu oraz bardzo dobrą odporność na zakłócenia. Standard TTL (Transistor-Transistor Logic) bierze nazwę od roli tranzystorów – w takim układzie cyfrowym zarówno funkcje logiczne (np. AND/OR) jak i wzmocnienie sygnału realizowane jest przez tranzystory (w przeciwieństwie do układów typu RTL czy DTL).

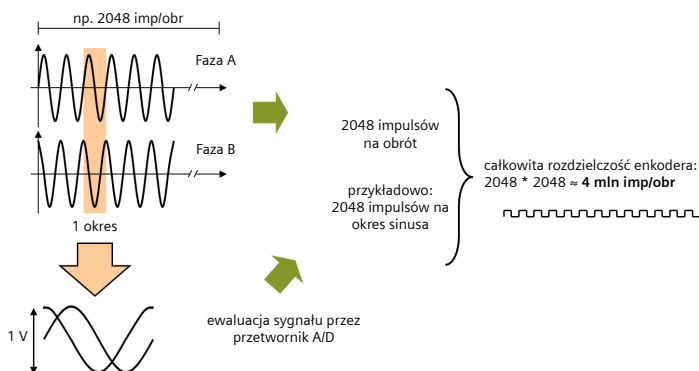
6.2 HTL

Wyjście enkodera w standardzie HTL (Push-pull) ma wartość napięcia proporcjonalną do poziomu napięcia zasilania (minimalnie 3V dla logicznej „1” oraz maksymalnie 0.5V dla logicznego „0”). Standardowo napięcie zasilające znajduje się w przedziale od 8 do 30 VDC. Struktura HTL (High Threshold Logic) jest odmianą układu diodowo-tranzystorowego (RTL – poprzednik TTL), gdzie funkcje logiczne (np. AND/OR) realizowane są przez sieć rezystorów, natomiast wzmocnienie przez tranzystory. Wysoki próg (high threshold) oznacza tutaj rozbieżność pomiędzy interpretacją poziomu sygnału wejściowego jako wartość logiczną 0 lub 1. Dzięki temu zyskujemy bardzo wysoką odporność na zakłócenia, tracimy jednak na prędkości wymiany informacji, zwiększony jest również pobór energii. Poniższa tabela prezentuje charakterystykę doboru odpowiedniej ramki komunikacyjnej. Najczęściej stosowane to telegramy standardowe 1, 2, 3 oraz 5.

6.3 SIN/COS

W obszarze rodziny enkoderów inkrementalnych znajdują się również urządzenia, które posiadają bardziej zaawansowaną metodę cyfrowego przetwarzania sygnału pozyskanego z układu mechanicznego. Enkodery inkrementalne SIN/COS pozyskują sygnał elektryczny podobnie jak te opisane w poprzedniej sekcji – mechaniczna część pozostaje bez zmian.

Zachodzi jednak różnica w zakresie przetwarzania sygnału impulsowego. Konkretnie jest on w przypadku tego typu urządzeń konwertowany na funkcję sinus (sygnał A) oraz cosinus (sygnał B). Konwersja odbywa się dzięki analizie czasu zmiany stanów sygnałów A/B. Następnie w zakresie jednego okresu funkcji trygonometrycznych zachodzi konwersja z próbkowaniem charakterystycznym dla danego układu elektronicznego. Przykładowo – założmy, że mamy optyczny enkoder inkrementalny z tarczą posiadającą 2048 szczelin. 2048 impulsów sygnału A (B) zostaje więc konwertowane na 2048 faz funkcji okresowej sinus (cosinus). Założmy również, iż w jednym okresie funkcji trygonometrycznej przetwornik analogowo/cyfrowy będzie w stanie wygenerować 2048 jednoznacznych stanów zależności funkcji sinus/cosinus. W związku z powyższym uzyskamy dokładność na wyjściu enkodera na poziomie 4 milionów impulsów na obrót.



Wyjściem enkodera typu SIN/COS może być sygnał impulsowy gotowy do przetworzenia w układzie sterownia lub analogowy sygnał SIN/COS, który konwertowany będzie dopiero po stronie systemu automatyki.

6.4 Połączenie z systemami SIMATIC/SINAMICS

Elementy sterowania SIMATIC można wyposażać w moduły dedykowane dla połączeń urządzeń pomiarowych dla systemów *Motion Control*. Enkoder inkrementalny może zostać połączony, np. ze sterownikiem S7-1200/1500 lub rozproszonymi

SIMATIC S7-1200

Jednostki centralne sterowników S7-1200 posiadają zintegrowane szybkie wejścia (6xHSC), które mogą zostać wykorzystane jako wejścia do podłączenia enkoderów inkrementalnych – maksymalna częstotliwość zliczania 100 kHz. Jednostka centralna pozwoli również zasilić enkodery 24V. Sterownik nie wspiera pracy w trybie izochronicznym. Nie ma również możliwości obsługi enkodera przez obiekt technologiczny.

SIMATIC S7-1500/ET 200MP				
Właściwość	Moduł technologiczny		Moduł DI	Compact CPU
	TM Count 2x24V	TM PosInput 2	DI 32x24VDC HF	CPU 1511C-1 PN, CPU 1512C-1 PN
Liczba kanałów	2	2	2	6
Maksymalna częstotliwość sygnału	200 kHz	1 MHz	1 kHz	100 kHz
Maksymalna częstotliwość zliczania dla enkoderów inkrementalnych z ewaluacją kwadraturową (A, /A, B, /B)	800kHz	4 MHz	-	400 kHz
Maksymalny zakres zliczania	32-bit	32-bit	32-bit	32-bit
Enkoder inkrementalny/pulsowy RS422/TTL	-	●	-	-
Enkoder inkrementalny 24 V	●	-	-	●
Enkoder pulsowy 24 V	●	-	●	●
Zasilanie enkoderów 5 V	-	●	-	-
Zasilanie enkoderów 24 V	●	●	-	●
Uproszczona konfiguracja <i>Motion Control</i>	●	●	-	●
Praca w trybie izochronicznym	●	●	●	-

SIMATIC ET 200SP			
Właściwość	Moduł technologiczny		Moduł DI
	TM Count 1x24V	TM PosInput 1	DI 8x24VDC HS
Liczba kanałów	1	1	4
Maksymalna częstotliwość sygnału	200 kHz	1 MHz	10 kHz
Maksymalna częstotliwość zliczania dla enkoderów inkrementalnych z ewaluacją kwadraturową (A, /A, B, /B)	800kHz	4 MHz	-
Maksymalny zakres zliczania	32-bit	32-bit	32-bit
Enkoder inkrementalny/pulsowy RS422/TTL	-	●	-
Enkoder inkrementalny 24 V	●	-	-
Enkoder pulsowy 24 V	●	-	●
Zasilanie enkoderów 5 V	-	-	-
Zasilanie enkoderów 24 V	●	●	●
Uproszczona konfiguracja <i>Motion Control</i>	●	●	-
Praca w trybie izochronicznym	●	●	●

SINAMICS

Podłączenie enkodera inkrementalnego do przekształtnika częstotliwości SINAMICS jest możliwe zarówno dla rodziny G120 jak i dla serwonapędów V90 (enkodery dedykowane) lub S1x0. W zależności od konfiguracji sprzętowej wspierane są enkodery inkrementalne typu HTL, TTL oraz SIN/COS, a także z interfejsami sieciowymi lub DRIVE-CLiQ, o czym w dalszej części.

7

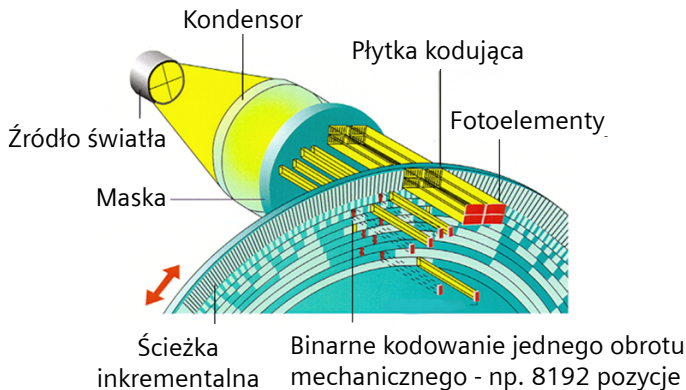
Enkodery absolutne

Budowa

Impulsy generowane są przez układ mechaniczny tarczy ze szczelinami, przez które przepuszczane jest światło w kierunku elementów światłoczułych. Obecność szczeliny powoduje oświetlenie fotoelementu, a co za tym idzie wygenerowanie napięcia, które interpretowane jest jako sygnał binarny przez układ elektroniczny. Alternatywnie stosowane są enkodery wykorzystujące technologię magnetyczną lub pojemnościową. Wybór typu enkodera uwarunkowany będzie dostosowaniem do warunków pracy urządzenia oraz oczekiwanych parametrów pomiaru.

Enkoder absolutny jest cyfrowym przetwornikiem kąta obrotu. Dla określonej pozycji kątowej enkoder absolutny generuje jednoznaczną wartość sygnału wyjściowego. Sygnał wyjściowy kodowany jest w taki sposób aby jego wartość została utrzymana nawet przy zaniku napięcia zasilającego. Przesunięcie wału enkodera bez jego zasilenia – również spowoduje wygenerowanie sygnału wyjściowego (po zasileniu) o wartości odzwierciedlającej aktualną pozycję. Dokładność układu pomiarowego zależy od jego wewnętrznej konstrukcji mechanicznej.

Poniższa ilustracja przedstawia budowę optycznego enkodera absolutnego:



Zasada działania

W przypadku enkodera absolutnego każda z pozycji (dokładność pomiaru wynika z parametrów mechanicznych) posiada swój unikatowy kod binarny, który interpretowany jest przez układ cyfrowy i wysyłany na wyjście enkodera. Kod binarny aktualnej pozycji generowany jest przez kręgi tarczy enkodera analizowane przez układ elektroniczny.

Podobnie jak w przypadku enkoderów inkrementalnych spotykamy różnego rodzaju technologie detekcji aktualnej pozycji. Najbardziej popularna to opisana wcześniej technika wykorzystująca zjawisko fotoelektryczne (rysunek). Kod binarny odczytywany jest z aktualnego układu szczelin tarczy enkodera, przez które przepuszczane jest światło w kierunku elementów światłoczułych. Obecność szczeliny powoduje oświetlenie fotoelementu, a co za tym idzie wygenerowanie napięcia, które interpretowane jest jako sygnał binarny (część kodu binarnego) przez układ elektroniczny. Alternatywnie stosowane są enkodery wykorzystujące

technologię magnetyczną lub pojemnościową. W przypadku enkoderów absolutnych stosowane są również sporadycznie układy w pełni mechaniczne, gdzie styki elektryczne podążają ścieżkami na tarczy enkodera – zależnie od ich pozycji na kręgach będą przepuszczać prąd lub będą odizolowane.

Kodowanie binarne realizowane jest zazwyczaj przez dwójkowy kod Graya. Cechuje się on tym, iż zmiana na wartość kolejną lub poprzednią powoduje podmiannę stanu tylko jednego bitu w całym kodzie. Jest to niewątpliwą zaletą, gdyż zapobiega występowaniu dużych błędów pomiaru. W przypadku standardowego kodu binarnego zmiana wartości na sąsiednią – może spowodować nawet zmianę wszystkich bitów kodu.

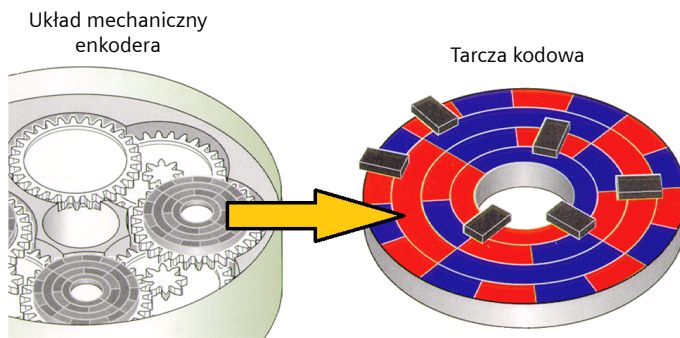
W celu zwiększenia dokładności pomiaru stosuje się również dodatkową ścieżkę inkrementalną.

Wybór typu enkodera uwarunkowany będzie dostosowaniem do warunków pracy urządzenia oraz oczekiwanych parametrów pomiaru.

Enkoder jedno-obrotowy / wielo-obrotowy

Zależnie od wymogów aplikacji możemy zastosować enkoder absolutny z tarczą kodową – jednoobrotowy (single-turn) lub z wieloma tarczami kodowymi oraz sprzęgającym je układem przekładni mechanicznych – tzw. enkoder absolutny wieloobrotowy (multi-turn). Różnica polega na zdolności zapamiętywania ilości wykonanych obrotów. Enkoder wieloobrotowy posiada możliwość zliczania oraz zapamiętywania pozycji dla więcej niż jednego obrotu.

Enkoder absolutny jednoobrotowy z konstrukcyjnego punktu widzenia wyposażony jest w tarczę z zakodowanymi (w formie binarnej lub w kodzie Gray'a) liczbami – bezpośrednio reprezentującymi wartość kąta obrotu. Obrócenie wału o dokładnie 360 stopni spowoduje wygenerowanie na wyjściu tej wartości początkowej. W przypadku enkodera absolutnego wieloobrotowego najczęściej część mechaniczna zostaje wzbogacona o system przekładni, które tworzą relację pomiędzy wieloma tarczami kodowymi, dając tym samym możliwość zapamiętania ilości wykonanych obrotów.



Podtrzymanie wartości pomiaru

Wartość aktualnej pozycji w technologii mechanicznej dostępna jest zawsze, niezależnie od tego czy enkoder jest zasilony – zgodnie z powyższym opisem. Stosowane są również rozwiązania poboczne, np. zastosowanie enkodera

inkrementalnego z baterijnym podtrzymaniem wartości lub enkoder inkrementalny ze zintegrowaną prądnicą oraz pamięcią nieulotną.

Zaletą mechanicznego układu przekładni w stosunku do rozwiązań elektronicznych jest brak konieczności posiadania podtrzymania baterijnego. Z drugiej jednak strony zwiększa się rozmiar urządzenia oraz wnosi konieczność zastosowania elementów, które się zużywają lub mogą ulec uszkodzeniu.

Rozdzielczość

Dokładność wyznaczenia pozycji wału enkodera zależy bezpośrednio od omówionej konstrukcji mechanicznej. Im więcej kręgów kodujących zostanie umieszczonych na tarczy enkodera, tym mniejsza zmiana pozycji kątowej zostanie zarejestrowana przez układ elektroniczny. Dokładność pomiaru enkoderów absolutnych określamy więc jako ilość kręgów, czyli ilość bitów jakie służą do zakodowania wartości aktualnej pozycji.

Zwiększenie rozdzielczości enkodera wiąże się bezpośrednio z jego konstrukcją, a co za tym idzie wpływa na jego gabaryty. Istotną kwestią jest również metoda transmisji danych do systemu automatyki. Rozdzielczości enkoderów absolutnych kodowania jednego obrotu sięga wartości 34-bitowych. Parametrem uzupełniającym w przypadku enkoderów wieloobrotowych jest liczba obrotów jakie mogą zostać zapamiętane.

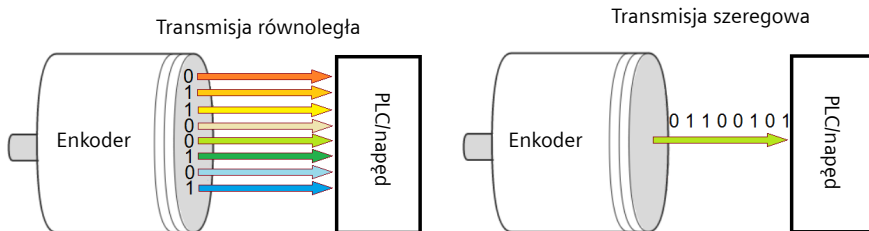
Przykładowo dla enkodera absolutnego wieloobrotowego o rozdzielczości 16-bitów oraz 12-bitów na zapamiętanie liczby obrotów uzyskamy wartość $216 \cdot 212 \approx 270$ mln jednoznacznych pozycji bez przeliczania w układzie sterowania lub napędzie.

Transmisja danych

Informację o aktualnej pozycji (w zależności od standardu oraz potrzeb aplikacji) może być wysyłana przez układ elektroniczny enkodera jako sygnał binarny (równoległe lub szeregowe), przez wyjście analogowe lub magistralę komunikacyjną.

Komunikacja równoległa jest bardzo szybka, gdyż każdy bit informacji ma swój elektryczny interfejs. Ta metoda komunikacji wymaga jednak zastosowania osobnego przewodu dla każdego z bitów kodu danej pozycji. Przy dużych rozdzielczościach rozwiązanie to niesie za sobą konieczność stosowania wielożyłowych kabli transmisyjnych. Przy dużych odległościach generuje to wysoki koszt oraz zmniejsza wygodę montażową. Aktualną tendencją w automatyce przemysłowej jest odchodzenie od połączeń równoległych i stosowanie w miarę możliwości jedynie transmisji szeregowej lub standardowych protokołów sieci przemysłowych.

Na schemacie przykład równoległej transmisji informacji 8-bitowej.



Transmisja szeregową jest wolniejsza od równoległej ale ogranicza ilość przewodów, a tym samym przy dużych odległościach daje możliwość redukcji kosztów.

Obecnie najczęściej spotykamy enkodery z szeregowymi interfejsami wyjściowymi (np. SSI, ISI, EnDat, BiSS czy Hiperface) lub współpracujące bezpośrednio z sieciami przemysłowymi (np. PROFIBUS czy PROFINET).

Firma Siemens wprowadziła również niezależny standard komunikacji na linii enkoder-napęd DRIVE-CLiQ, który wprowadza sporo udogodnień w konfiguracji systemu napędowego.

Cechy enkoderów absolutnych

- Koszt wyższy od enkoderów inkrementalnych,
- gabaryty wzrastają wraz z rozdzielczością,
- pozycja absolutna,
- stosunkowo niska prędkość transmisji danych,
- nie jest wymagane bazowanie systemu po zaniku zasilania.

7.1 Siemens DRIVE-CLiQ

DRIVE-CLiQ to otwarty protokół komunikacyjny dedykowany dla połączeń komponentów napędowych z systemem SINAMICS. Jest to autorska sieć opracowana przez firmę Siemens bazująca na Industrial Ethernet 100Mbit/s. Standard pozwala na podłączenie zarówno podzespołów Siemens (silniki serwo, moduły napędów, jednostki sterujące CU, enkodery lub moduły terminalowe I/O) jak i urządzeń producentów trzecich, którzy wprowadzili w ich zakresie kompatybilność z interfejsem DRIVE-CLiQ.



Zalety standardu DRIVE-CLiQ:

- Wysoka wydajność komunikacyjna,
- standardowy interfejs (niezależny od producenta),
- zintegrowane funkcje bezpieczeństwa (SINAMICS Safety Integrated),
- łatwa konfiguracja,
- automatyczna konfiguracja przez elektroniczną tabliczkę znamionową,
- kompaktowe oraz uniwersalne okablowanie dla wszystkich enkoderów,
- szybka oraz przejrzysta diagnostyka systemu pomiarowego (status urządzenia).

7.2 SSI

Synchronous Serial Interface (SSI) jest powszechnie stosowanym w aplikacjach przemysłowych szeregowym (cyfrowym) interfejsem komunikacyjnym typu point-to-point. Wymiana informacji zachodzi tutaj pomiędzy urządzeniem typu master (np. sterownik bądź napęd) a slave (np. czujnik, enkoder). SSI bazuje na standardzie elektrycznym RS-422 i cechuje się wysoką efektywnością komunikacyjną, co więcej jest prosty w implementacji w szerokim spektrum platform sprzętowych, co czyni go bardzo popularnym wśród producentów urządzeń pomiarowych. Protokół znajduje zastosowanie w aplikacjach wymagających dużej niezawodności oraz odporności na niekorzystne warunki otoczenia.

Wymiana informacji odbywa się jednokierunkowo zgodnie z taktom zegara systemowego urządzenia master. Zazwyczaj w okablowaniu stosowane są dwie skrętki – jedna na transfer danych (*enkoder* → *napęd*), druga na sygnał zegara (*napęd* → *enkoder*) oraz dwa przewody zasilające. Prędkość wymiany danych zależy od długości przewodów transmisyjnych, maksymalnie sięga 10Mbit/s (dla przewodu o długości maksymalnie 5m).

7.3 EnDat

Encoder Data (EnDat) jest szeregowym interfejsem komunikacyjnym punkt-punkt dedykowanym do wymiany danych z enkoderami. Standard został opracowany przez niemiecką firmę Heidenhain i jest powszechnie stosowany w przemyśle. Podobnie jak SSI jest protokołem szeregowym ale daje możliwość wysyłania informacji dwukierunkowo – odczyt informacji (o pozycji/prędkości, statusie, typie urządzenia z enkoderów inkrementalnych oraz absolutnych), oraz wysyłanie danych do enkodera (aktualizacja parametrów enkodera lub zapisywanie nowych danych).

Komunikacja szeregową ogranicza okablowanie do 4 przewodów transmisyjnych (plus zasilanie). Dane przesłane są synchronicznie zgodnie z taktom generatora zegara po stronie urządzenia nadrzędnego. Typ transmisji (informacja o pozycji, ustawianie parametrów czy diagnostyka urządzenia) określany jest przez komendy wysyłane z PLC lub napędu. Prędkość wymiany danych zależy od długości przewodów transmisyjnych, maksymalnie sięga 4Mbit/s

7.4 Hiperface

Interfejs komunikacyjny enkoderów opracowany przez firmę Max Stegmann GmbH (obecnie SICK). Standard bazuje na standardzie elektrycznym RS-485. Wykorzystuje przewód 8-żyłowy – dwa komunikacyjne RS-485, dwa do zasilania oraz cztery na sygnał inkrementalny SIN/COS. Po zasileniu enkodera – informacja o aktualnej pozycji transferowana jest przez RS-485. Podczas dalszej pracy – zmiana pozycji rejestrowana jest przez sygnał inkrementalny. Aktualna pozycja nie jest korygowana przez sygnał absolutny, a co za tym idzie może wystąpić usterka, która spowoduje niepoprawny odczyt pozycji. W przeciwieństwie do większości standardów szeregowych Hiperface nie ma specyfiki synchronicznej – nie ma taktu zegara generującego cykliczną wymianę danych.

Standard umożliwia komunikację point-to-point oraz pracę w sieci przemysłowej (jeden master dla wielu urządzeń pomiarowych). Ułatwia to kwestie okablowania. Dane przesyłane są stosunkowo powoli (38.4kbit/s) jednakże otrzymujemy znacznie więcej informacji niż, np. w przypadku interfejsu SSI. Podobnie jak w przypadku EnDat, możemy uzyskać dostęp do pamięci wewnętrznej enkodera – np. odczytać parametry silnika – napięcie, prąd czy typ; odebrać dane statusowe lub zapisać informacje w pamięci urządzenia.

7.5 BiSS

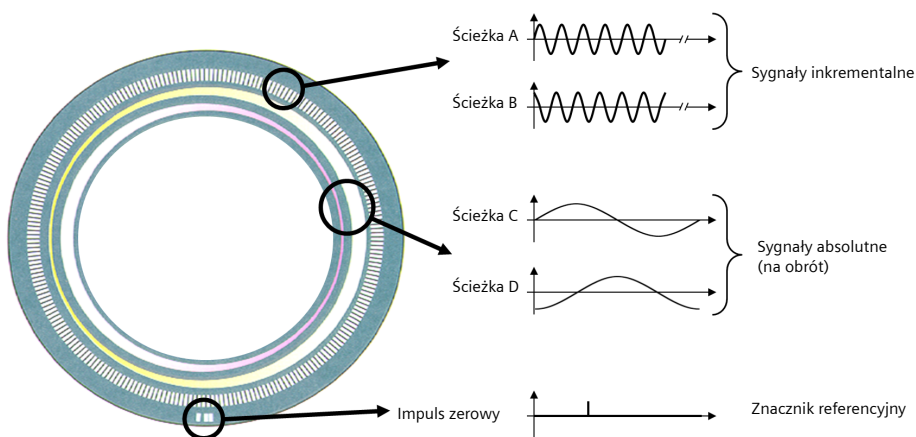
Bidirectional Serial Synchronous (BiSS) jest stosunkowo nowym standardem komunikacyjnym (opracowanym przez niemiecką firmę iC-Haus GmbH) dedykowanym dla czujników, enkoderów oraz urządzeń wykonawczych. BiSS został stworzony jako otwarta alternatywa dla standardów EnDat oraz Hiperface – oba kompatybilne sprzętowo z BiSS (RS-422).

Jako interfejs dwukierunkowy podobnie jak wspomniane protokoły – pozwoli na odczyt szeregu informacji z nadajnika (pozycja absolutna, alarmy, ostrzeżenia, diagnostyka, parametry silnika czy temperatura) oraz zapis parametrów w pamięci wewnętrznej enkodera. Wymiana danych wywoływana jest synchronicznie.

Urządzenia w standardzie BiSS mogą pracować w strukturze punkt-punkt oraz w sieci przemysłowej. Przewód 6-żyłowy pozwala na komunikację z prędkością nawet 100Mbit/s.

7.6 SIN/COS ze ścieżką C/D

Specyficznym interfejsem komunikacyjnym stosowanym w enkoderach absolutnych jest omówiony już wcześniej standard inkrementalny SIN/COS, z tym, że wzbogacony o dodatkowe ścieżki sinusoidalne pozwalające zidentyfikować pozycję absolutną. Konstrukcja tarczy enkodera (rysunek) pozwala na dokładną interpretację przyrostu pozycji (sygnał A/B konwertowany na sinus oraz cosinus), natomiast dodatkowe ścieżki umieszczone na tarczy enkodera (C/D) reprezentujące (w jednym obrocie mechanicznym) pełny okres funkcji sinus oraz cosinus pozwalają na identyfikację unikalnej pozycji kątowej.

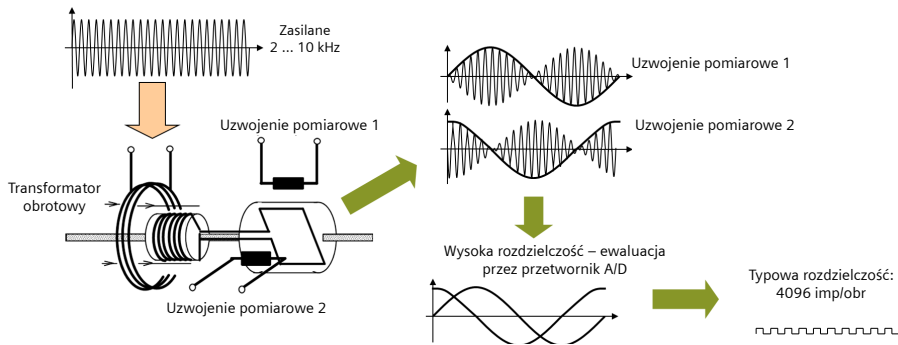


Zaletą takiego rozwiązania jest możliwość próbkowania sygnałów analogowych ścieżek C/D z bardzo wysoką częstotliwością, co pozwala na osiągnięcie adekwatnej rozdzielczości pomiaru przy zachowaniu dobrych parametrów transmisji danych.

7.7 Resolver

Resolver to rodzaj przetwornika kąta obrotu, który można określić jako analogową wersję enkodera absolutnego. Przez analogowy sygnał wyjściowy dostarcza on informacji o aktualnym położeniu wału.

Zasada działania jest podobna jak w silniku elektrycznym – urządzenie składa się z wirnika oraz stojana. Uzwojenie pierwotne (wirnika) zasilane jest (przez transformator obrotowy) prądem zmiennym, natomiast w uzwojeniach wtórnych (stojana) – przesuniętych względem siebie o kąt 90° – indukują się napięcia, które są proporcjonalnymi do sinusa oraz cosinusa kąta obrotu. Sinusoidalne sygnały wyjściowe jednoznacznie określają pozycję kątową wału.



Resolver jest urządzeniem indukcyjnym, a co za tym idzie w praktyce nie wymaga żadnej dodatkowej elektroniki w swojej konstrukcji. Sprawia to, że jest on stosunkowo tani oraz odporny na niekorzystne warunki pracy. Uzyskamy tutaj również relatywnie wysoką dokładność pomiaru.

7.8 Sieć przemysłowa

Najbardziej zaawansowaną oraz najwygodniejszą metodą komunikacji jest wykorzystanie sieci przemysłowej. Standard komunikacyjny dla wszystkich urządzeń systemu automatyki pozwala na łatwą konfigurację, szybkie uruchomienie oraz kompleksową diagnostykę.

Wśród najpopularniejszych sieci przemysłowych, w których komunikować mogą się enkodery absolutne możemy wyszczególnić:

- wysoka wydajność komunikacyjna,
- standardowy interfejs (niezależny od producenta),
- zintegrowane funkcje bezpieczeństwa (SINAMICS Safety Integrated),
- łatwa konfiguracja,
- automatyczna konfiguracja przez elektroniczną tabliczkę znamionową,
- kompaktowe oraz uniwersalne okablowanie dla wszystkich enkoderów,
- szybka oraz przejrzysta diagnostyka systemu pomiarowego (status urządzenia).

Zgodność z określonym standardem będzie kwestią charakterystyczną dla danego producenta.

Specyfika każdej z sieci jest inna – wybór odpowiedniego standardu będzie zależny od dostępnych protokołów, odległości oraz prędkości wymiany danych, warunków pracy urządzeń, typu okablowania, struktury komunikacyjnej, maksymalnej ilości urządzeń w sieci czy po prostu preferencji użytkownika.

Poniższa tabela przedstawia poglądowe porównanie podstawowych parametrów wybranych sieci przemysłowych, w których enkodery absolutne mogą występować jako urządzenia pomiarowe.

Sieć	Prędkość transmisji	Maks. rozmiar segmentu	Liczba stacji
Profinet IO	1 Gbit/s	100 m	256/segment
Profibus DP	12 Mbit/s	1200 m	126
Modbus RTU	19,2 Kbit/s	350 m	250
InterBus	500 Kbit/s	12.8 km	256
CanOpen	1 Mbit/s	1000 m	127
DeviceNet	500 Kbit/s	500 m	64

7.9 Enkoder liniowy

Specyficzną odmianą enkodera jest enkoder liniowy (inkrementalny bądź absolutny). Zasada działania jest podobna jak w przypadku enkoderów obrotowych – występują analogiczne technologie pomiaru (np. optyczne, magnetyczne lub pojemnościowe). Wielkość mierzona przetwarzana jest na sygnał elektryczny – analogowy lub cyfrowy. Zasadniczą różnicą w stosunku do enkodera obrotowego jest fakt, iż celem urządzenia nie jest wykrycie obrotu wału, a przesunięcie liniowe względem wzorca np. magnetycznej taśmy pomiarowej umieszczonej na ruchomym elemencie.

7.10 Połączenie z systemami SIMATIC/SINAMICS

Specyficzną odmianą enkodera jest enkoder liniowy (inkrementalny bądź absolutny). Zasada działania jest podobna jak w przypadku enkoderów obrotowych – występują analogiczne technologie pomiaru (np. optyczne, magnetyczne lub pojemnościowe). Wielkość mierzona przetwarzana jest na sygnał elektryczny – analogowy lub cyfrowy. Zasadniczą różnicą w stosunku do enkodera obrotowego jest fakt, iż celem urządzenia nie jest wykrycie obrotu wału, a przesunięcie liniowe względem wzorca np. magnetycznej taśmy pomiarowej umieszczonej na ruchomym elemencie.

SIMATIC

Podłączenie enkodera absolutnego do systemów SIMATIC możliwe jest wyłącznie przez moduł technologiczny dla S7-1500/ET 200MP – TM PosInput 2 lub przez analogiczny moduł dla systemów ET 200SP – TM PosInput 1. Jedynym wspieranym standardem komunikacji szeregowej jest SSI.

Alternatywą w przypadku enkoderów absolutnych jest zastosowanie enkodera z interfejsem sieci przemysłowej – w takim przypadku wykorzystując odpowiedni moduł komunikacyjny możemy wykonać bezpośrednie połączenie, np. przez sieć PROFINET, PROFIBUS czy MODBUS.

Właściwość	Moduł technologiczny	
	TM PosInput 1 (ET 200SP)	TM PosInput 2 (S7-1500/ET 200MP)
Liczba kanałów	1	2
Maksymalna częstotliwość sygnału	1 MHz	1 MHz
Maksymalna częstotliwość zliczania dla enkoderów inkrementalnych z ewaluacją kwadraturową (A, /A, B, /B)	4 MHz	4 MHz
Enkoder inkrementalny/pulsowy RS422/TTL	●	●
Maksymalny zakres zliczania	32-bit	32-bit
Enkoder absolutny SSI	●	●
Maksymalny zakres wartości pozycji absolutnej	31-bit	31-bit
Enkoder inkrementalny 24 V	-	-
Enkoder pulsowy 24 V	-	-
Zasilanie enkoderów 5 V	-	●
Zasilanie enkoderów 24 V	●	●
Uproszczona konfiguracja <i>Motion Control</i>	●	●
Praca w trybie izochronicznym	●	●

Panel *Diagnostics* – dostępny dla osi pozycjonującej – podobnie jak w przypadku obiektu technologicznego *TO_SpeedAxis* dostarcza informacji statusowych oraz informuje o błędach pracy układu.

SINAMICS

W przypadku systemów SINAMICS mamy nieco więcej możliwości. Bezpośrednie połączenie enkodera absolutnego do przekształtnika częstotliwości jest możliwe zarówno dla rodziny G120 jak i dla serwonapędów V90 (enkodery dedykowane) oraz S1x0. W zależności od konfiguracji sprzętowej wspierane są enkodery z interfejsem DRIVE-CLiQ oraz szeregowo typu SSI,

EnDat, SIN/COS, resolvery lub z enkodery interfejsem sieci przemysłowej (np. PROFINET czy PROFIBUS). Poniższa tabela prezentuje możliwości połączeń enkoderów bezpośrednio do jednostek sterujących CU napędów SINAMICS.

			Enkodery inkrementalne			Enkodery absolutne		
Napędy SINAMICS		Resolver	HTL	TTL	Sin/cos 1Vpp	Endat	SSI	DRIVE CLiQ
G120 z CU250S-2		●	●	●	●	●	●	●
G120 z CU250S-2 poprzez moduł SMC lub SME		●	●	●	●	●	●	
S110*			●	●			●	●
S110 poprzez moduł SMC lub SME		●	●	●	●	●	●	
S120	CU310-2 lub CUA32		●	●			●	
	CU310-2 poprzez moduł SMC lub SME	●	●	●	●	●	●	
	CU320-2							●
	CU320-2 poprzez moduł SMC lub SME	●	●	●	●	●	●	

*umożliwia podłączenie tylko 1 enkodera

Informację zawarte w powyższej tabeli są ogólnym opisem obsługiwanych, przez poszczególne jednostki sterujące CU, enkoderów.

Dokładne informacje odnośnie możliwości łączenia enkoderów znajdują się w dokumentach technicznych dostępnych do pobrania ze strony:
<https://support.industry.siemens.com/>

W razie pytań lub wątpliwości zapraszamy również do bezpośredniego kontaktu ze wsparciem technicznym techniki napędowej: automatyka.pl@siemens.com.

III. Sterowanie pojedynczą osią

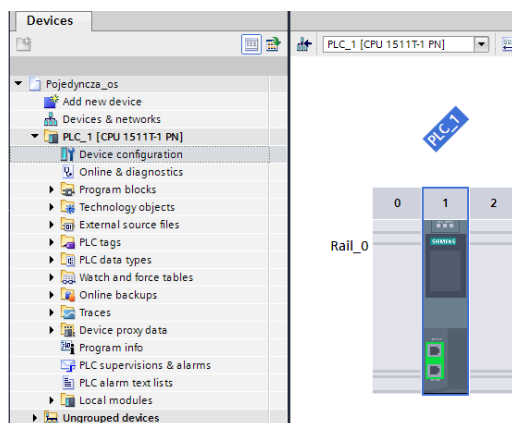
W kolejnym rozdziale zajmiemy się omówieniem zagadnień zadawania prędkości oraz pozycjonowania w strukturze scentralizowanej *Motion Control*, czyli w takiej, gdzie sterownik PLC będzie nadrzędnym urządzeniem wydającym polecenia bezpośrednio do napędu. Dzięki interfejsowi PROFINET, sterownik będzie kontrolerem I/O (*master*) – zawiera regulator pozycji, natomiast napęd będzie urządzeniem typu I/O device (*slave*) – tutaj odbywać się będzie regulacja prędkości.

8

Konfiguracja sprzętowa

8.1 Dodanie sterownika do projektu

W zakresie projektu TIA Portal V14 dodajemy jednostkę PLC
– w przykładzie zastosowano CPU 1511T (6ES7511-1TK01-0AB0).

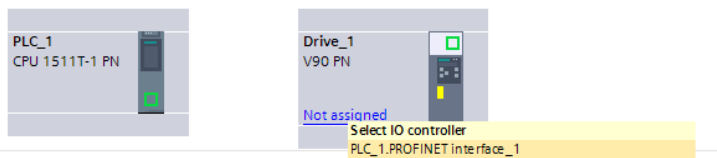


8.2 Wstawienie do projektu napędu oraz jego konfiguracja

Do projektu wstawiamy napęd SINAMICS V90. Aby podstawowa (jednocześnie wystarczająca z naszego punktu widzenia) konfiguracja napędu w wersji PN była możliwa w środowisku TIA Portal – należy doinstalować najnowszą aktualizację katalogu sprzętowego (dostępną w następującej lokalizacji sieciowej: <https://support.industry.siemens.com/cs/us/en/view/72341852>). Aktualizację instalujemy przez TIA Portal korzystając z menu *Options* → *Support packages*.

Po tym zabiegu z drzewka katalogu sprzętowego (*Drives & starters* → *SINAMICS drives* → *SINAMICS V90 PN*) wybieramy odpowiedni napęd. W naszym przykładzie jest to V90 PN, 1AC/3AC 200V-240V, 01kW (6SL3210-5FB10-1UF0).

W widoku sieci (*Devices & networks*) przypisujemy napęd (I/O device) do kontrolera sieci PROFINET (S7-1500 CPU).

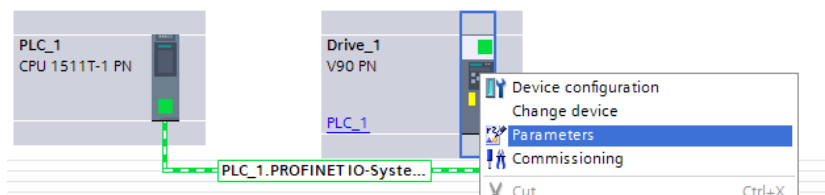


Uwaga – napęd SINAMCS V90 PN domyślnie pracuje w trybie izochronicznym, a do komunikacji z systemem nadrzędnym przypisany jest telegram Siemens 105. W związku z powyższym po wykonaniu połączenia ze sterownikiem nadrzędnym – kompilacja projektu PLC będzie zgłaszała błąd na poziomie telegramu. Wynika to z faktu, iż zastosowanie telegramu 105 determinuje również tryb komunikacji izochronicznej – nie da się go wyłączyć w tej konfiguracji. Ten z kolei wymaga przypisania zegara/generatora cyklicznego przerwania dla obsługi wymiany danych trybu IRT. Takim cyklicznym przerwaniem jest blok organizacyjny MC_Servo. Cykl bloku można określić w jego ustawieniach bądź dostosować do ustawień sieci komunikacyjnej PROFINET IO RT/IRT (o czym w dalszej części dokumentu).

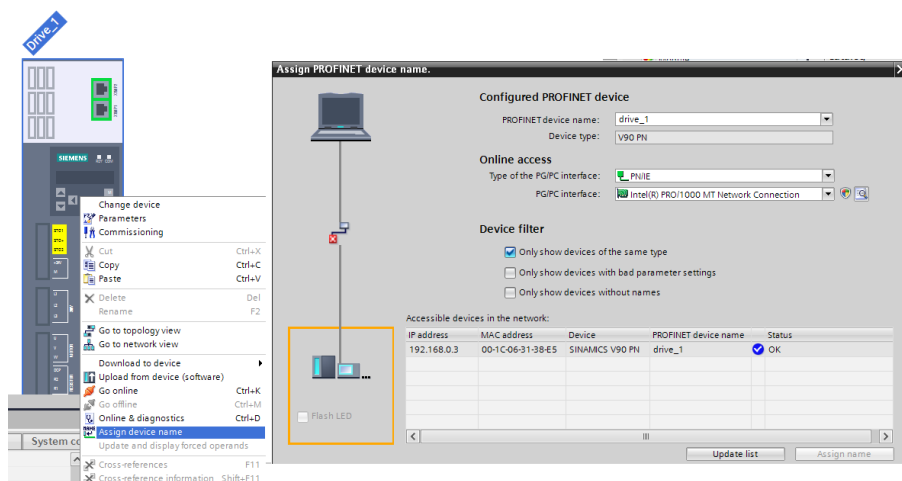
Blok ten jednak generowany jest przez system dopiero w momencie dodania do projektu sterownika – obiektu technologicznego. W chwili obecnej aby kompilacja projektu PLC przebiegła poprawnie – należy dodać obiekt technologiczny MC (co z robimy w kolejnych krokach) lub zmienić telegram PROFIdrive na niewymagający trybu izochronicznego, np. 102.

W systemach wspierających komunikację w trybie izochronicznym – jego zastosowanie jest zalecane.

W kolejnym kroku klikamy prawym przyciskiem myszy w napęd i wybieramy z menu podręcznego opcję Parameters. Tutaj określimy typ zastosowanego silnika/enkodera (w przykładzie SIMOTICS 1FL6024-2AF21-1AA1) oraz jego podstawowe parametry.



Jako, że napęd występuje w strukturze sieci PROFINET IO jako I/O device – konieczne jest przypisanie mu nazwy. W tym celu z poziomu okna konfiguracji napędu wybieramy opcję *Assign device name* i przypisujemy nazwę urządzenia.

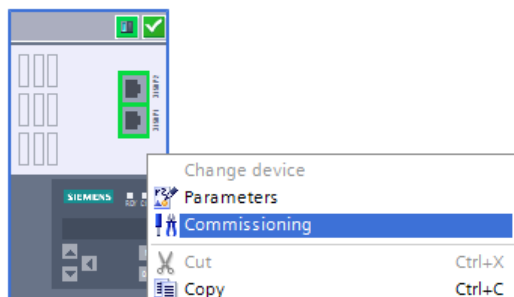


Wgrywamy konfigurację do napędu

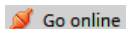


8.3 Uruchomienie napędu

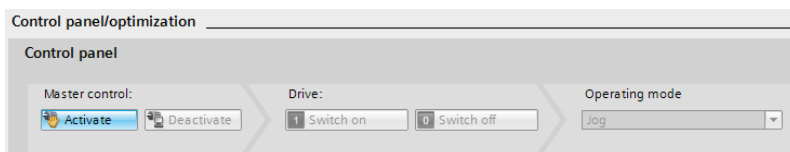
Na tym etapie konfiguracji możemy wykonać uruchomienie oraz wstępną optymalizację napędu z poziomu panelu sterowania dostępnego w TIA Portal. Przechodzimy po raz kolejny do konfiguracji sprzętowej napędu, tym razem z menu podręcznego wybieramy opcję Commissioning.



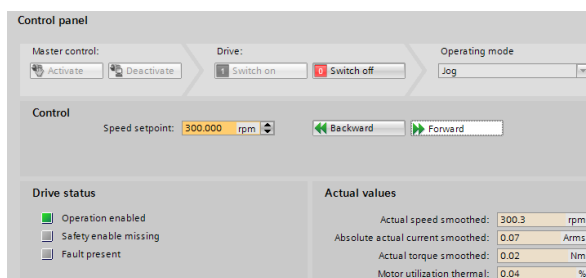
Dalsze kroki wymagają aktywnego połączenia z napędem – należy przejść w tryb online



W panelu sterowania lokalnego należy przejąć kontrolę nad napędem. W tym celu w polu *Master control* klikamy przycisk *Activate*. Wykonanie polecenia będzie jednoznaczne z odłączeniem ewentualnego sterowania nadrzędnego. Do momentu kliknięcia przycisku *Deactivate*, żadne komendy wydawane z systemu nadrzędnego poprzez magistralę komunikacyjną – nie będą w napędzie realizowane. Kontrola jest teraz lokalna przez urządzenie programujące i środowisko TIA Portal.



Aby uruchomić napęd – klikamy przycisk *Switch on* w polu *Drive*. Następnie wybieramy tryb pracy *Jog* (jazda gdy przycisk aktywny) lub *Continuosly* (jazda ciągła), a na końcu wysyłamy komendę jazdy w prawo lub w lewo przyciskami *Forward/Backward*.



W oknie panelu sterowania – poza podstawowymi parametrami pracy napędu – możemy również podejrzeć aktualny status funkcji bezpieczeństwa STO oraz bufor błędów.

Jeśli napęd poprawnie steruje ruchem silnika, możemy przejść do jego optymalizacji (zakładka *Optimization*). Operacja autotuningu dostarcza parametry regulatora prędkości napędu, aby pracował on optymalnie do zastosowanego układu mechanicznego maszyny. Analizie podlegają prądy, moment obrotowy, przesunięcie faz i ich reakcje na wartości zadane w określonym zakresie obrotu osi. Na podstawie tych testów system wyznacza optymalne parametry pętli regulacyjnej (wzmocnienie oraz czas całkowania – regulatora PI). Zaleca się przeprowadzenie procedury przez mechanicznym zmontowaniem maszyny oraz na etapie uruchomienia końcowego.

W celu inicjalizacji automatycznej optymalizacji systemu w panelu sterowania aktywujemy kontrolę nadrzędną (*Master control* → *Activate*). Ustawiamy parametry autotuningu (dynamikę, czas trwania sygnału testującego czy maksymalny dozwolony kąt odchylenia osi). Klikamy przycisk *Start optimization*. System przez jakiś czas będzie przeprowadzał analizę – po jej wykonaniu zostanie wygenerowana wartość parametrów regulatora prędkości.

The screenshot shows the 'Optimization' and 'Configure' sections of the SIMATIC Manager interface. In the 'Optimization' section, the 'Master control' is set to 'Deactivate'. The 'Configure' section includes parameters for tuning: 'Tuning: Dyanmic factor' (18), 'Duration of the test signal' (2000 ms), and 'Maximum angle of motion in each direction (absolute valu...' (360). A warning message states: 'Warning: The motor must be able to rotate freely irrespective of this value ±720°. For motors with incremental encoders, the motor must be able to rotate freely in optimization phase for two revolutions (720°) in the positive and in the negative c irrespective of the set maximum angle of motion.' Below this, the 'Perform (rotating measurement)' section shows a 'Start optimization' button and a table of parameters:

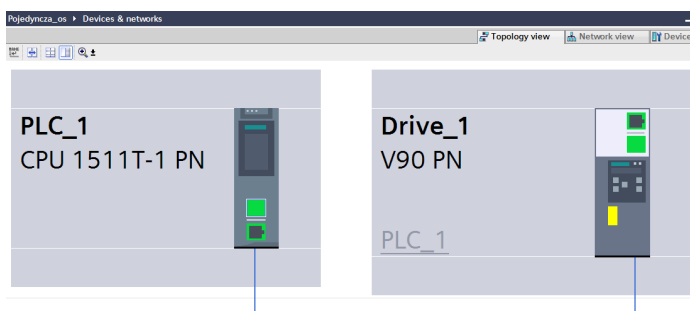
	New value	Previous value
Speed Loop Gain:	0.0029 Nms/rad	0.0029 Nms/rad
Speed Loop Integral time:	9.94 ms	9.93 ms
Ratio of the total moment of inertia to the motor moment of inertia:	1.01	1.00

Buttons for 'Reset to previous values' and 'Save data in the drive' are also visible.

Obliczone parametry są zapisane chwilowo w środowisku inżynierskim. W celu przepisania nastaw do napędu należy wykonać polecenie *Save data in the drive*. Od tego momentu napęd jest sparametryzowany oraz uruchomiony. Nie będzie konieczności powracania do tych ustawień – w naszym przykładzie dalsze funkcje programowe realizować będziemy przez obiekty technologiczne sterownika nadrzędnego. Aby pobrać wyznaczone wartości parametrów z napędu do projektu – na koniec wykonujemy upload do TIA Portal.

8.4 Topologia sieci PROFINET IO IRT

Przy założeniu, że nasz system będzie zawsze pracował w zalecanym trybie IRT – podstawowym wymaganiem sieci gdzie komunikacja odbywa się izochronicznie jest określenie jej topologii. Oznacza to ściśle przepisanie w projekcie fizycznych numerów portów, przez które połączone są urządzenia. Jest to konieczne do poprawnej pracy systemu, samo połączenie urządzeń w układzie sieci nie jest wystarczające. Przechodzimy zatem w widok topologii (*Topology view*) i zgodnie z fizycznymi relacjami wykonujemy połączenia w projekcie.



9

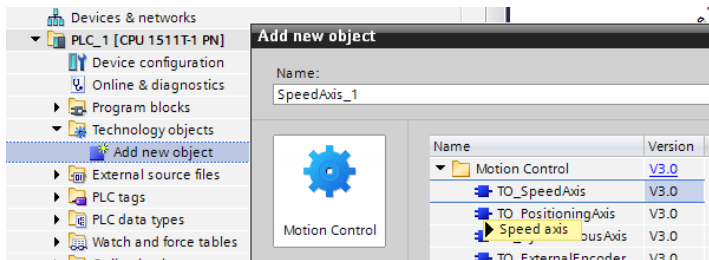
Zadawanie prędkości

Sterowanie prędkością osi jest podstawową funkcją układów napędowych. Interfejs PROFINET w napędach SINAMICS V90 daje wiele zaawansowanych możliwości funkcjonalnych oraz sprawia, że konfiguracja systemu jest bardzo szybka i prosta. W niniejszym rozdziale opiszemy kolejne etapy konfiguracji funkcji zadawania prędkości w strukturze scentralizowanej (zgodnie z opisem z poprzedniej sekcji).

Zadanie sterownia prędkością w sieci PROFINET przez obiekty technologiczne sterownika nadrzędnego może zostać zrealizowane przez jednostkę z S7-1200, S7-1500 lub S7-1500T.

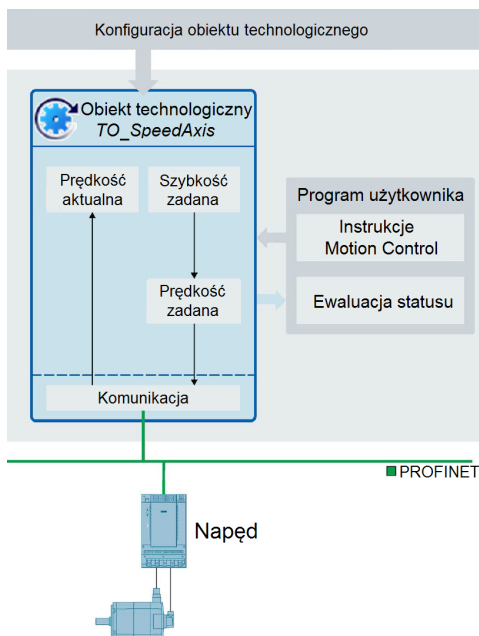
9.1 Konfiguracja obiektu technologicznego

Wykorzystany w naszym projekcie napęd SINAMICS V90 PN pracuje domyślnie w trybie IRT. W związku z powyższym – w celu poprawnej kompilacji projektu – po dodaniu i połączeniu PLC z napędem, konieczne jest przypisanie urządzeń podrzędnych do cyklu izochronicznego mastera IRT. Cykl ten generowany jest przez izochroniczny blok przerwań OB (MC_Servo). Aby blok został wygenerowany w projekcie – należy dodać do sterownika PLC obiekt technologiczny *Motion Control*. W naszym przykładzie chcemy sterować prędkością, zatem dodamy obiekt technologiczny *TO_SpeedAxis* zgodnie z poniższą ilustracją.

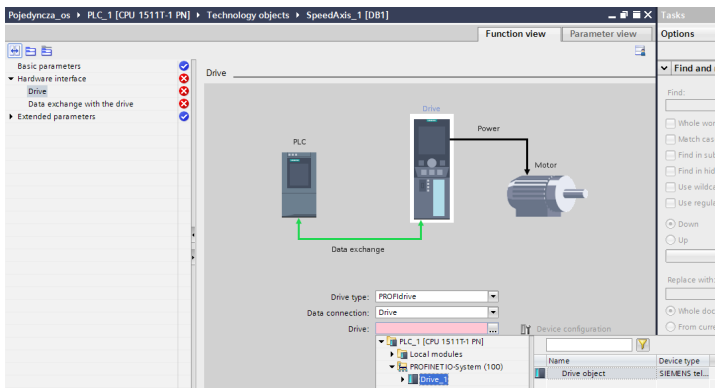


Obiekt technologiczny – oś prędkościowa

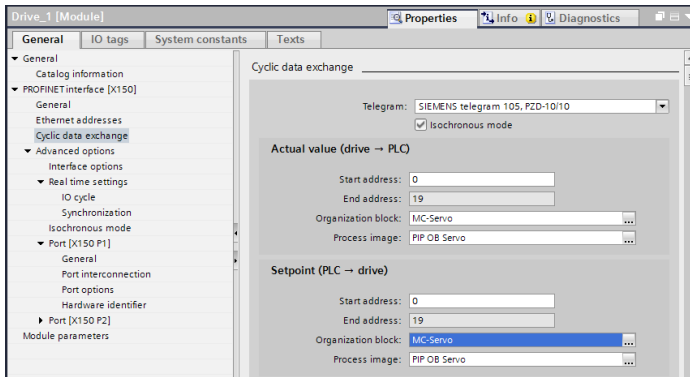
Na poniższym schemacie przedstawiono zasadę działania obiektu technologicznego *TO_SpeedAxis*.



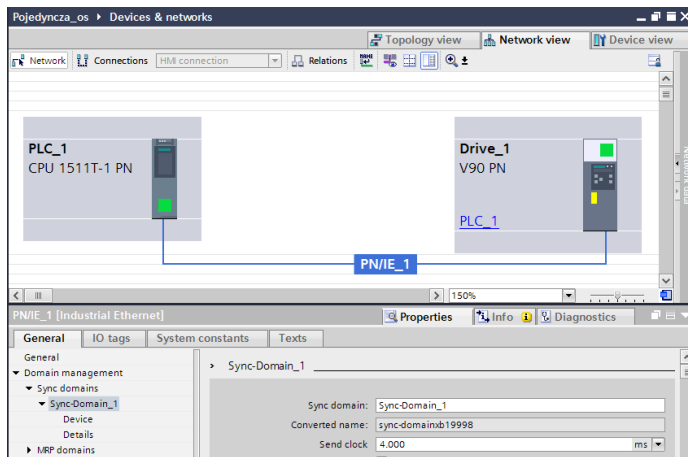
Pierwszym i bezwzględnie wymaganym krokiem jaki należy wykonać po dodaniu obiektu technologicznego, jest przypisanie do niego fizycznego napędu. W tym celu w zakładce konfiguracyjnej obiektu technologicznego *Hardware interface* → *Drive* wskazujemy skonfigurowany wcześniej napęd SINAMICS V90. Komunikacja z napędem odbywa się u nas przez *PROFIdrive* (alternatywnie sygnał analogowy).



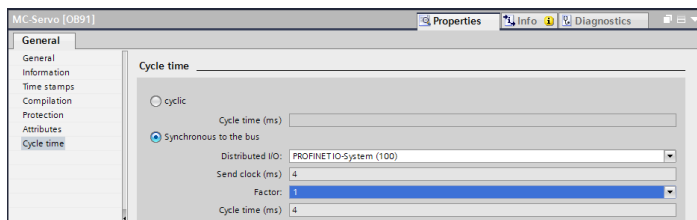
Po tym kroku – system automatycznie przypisze blok organizacyjny *MC_Servo* jako generator cyklu wymiany danych pomiędzy napędem a PLC. Możemy sprawdzić poprawność w panelu konfiguracyjnym napędu *PROFINET interface* → *Cyclic data exchange* → *Actual value/Setpoint* → *Organization Block*.



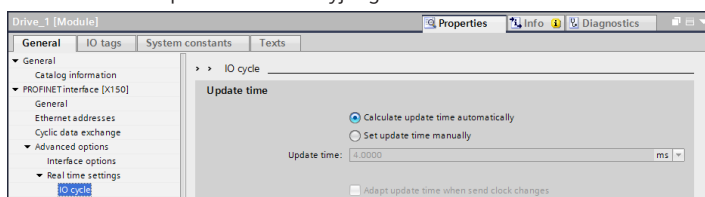
Cykl przetwarzania danych jest parametrem sieci PROFINET IO IRT. Zatem nadrzędnym ustawieniem dla całego projektu jest określenie cyklu domeny synchronizacyjnej w ustawieniach *Devices&networks*. Stamtąd zarówno sterownik jak i napęd będzie pobierać informacje dotyczące cyklu wymiany informacji w sieci komunikacyjnej.



Przejęcie wartości parametru *Send clock* po stronie sterownika można sprawdzić w ustawieniach bloku przerwań *MC_Servo*.



Możemy również tutaj ustawić cykl niezależny od cyklu zegara sieci PROFINET IRT lub powielić zaadoptowaną z ustawień sieciowych wartość przez ustawienie odpowiedniego współczynnika. Po stronie napędu wartość cyklu widnieć powinna w ustawieniach portu komunikacyjnego.



Musi on w tym przypadku być zgodny z cyklem zegara sieci

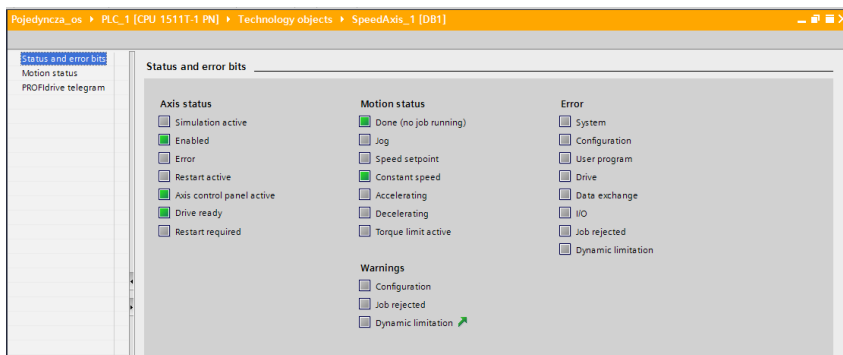
W trybie IRT sterownik S7-1500 może przyjąć minimalny cykl 1 ms, natomiast napęd SINAMICS V90 obsługuje cykl do 2 ms. W zależności od wymagań aplikacji wybierzemy odpowiedni cykl. Nawet w przypadku aplikacji o wysokiej dynamice w zupełności powinno wystarczyć ustawienie cyklu sieci na 4 ms.

Dalsze ustawienia obiektu technologicznego *TO_SpeedAxis* określają jednostki parametrów jakie będą stosowane dalej w projekcie użytkownika (*Basic parameters*), mechanikę systemu – ewentualna przekładnia (*Mechanics*), domyślne oraz graniczne wartości prędkości/przyspieszenia/zrywu, a także charakterystykę ich zmian w czasie (*Extended parameters*). Domyślne wartości stosowane są przez system automatycznie w programie użytkownika jeśli nie zostanie określone inaczej w parametrach funkcji sterujących.

Przy zastosowaniu ramki PROFIdrive Siemens (np. 105) możemy również określić limit momentu obrotowego (*Torque limit*).

W zakładce *Commissioning* obiektu technologicznego *TO_SpeedAxis*, podobnie jak na etapie konfiguracji napędu, jesteśmy w stanie wywołać próbny przejazd osi już przez obiekt technologiczny, czyli przez napęd zmapowany na obszar pamięci sterownika. Wgrrywając projekt do symulatora PLCSim możemy również wykonać symulację procedury bez konieczności posiadania fizycznych urządzeń.

Panel *Diagnostics* daje sporo możliwości diagnostycznych, które znacznie usprawniają uruchomienie systemu oraz diagnozowanie ewentualnych działań nieoczekiwanych. Obiekt technologiczny dostarcza informacji statusowych osi, trybu pracy, błędów oraz ostrzeżeń.



Bardziej szczegółowych informacji diagnostycznych dostarczają funkcje użytkownika wykorzystane w projekcie – przez zwracane ID błędów.

9.2 Podstawowe funkcje interfejsu użytkownika

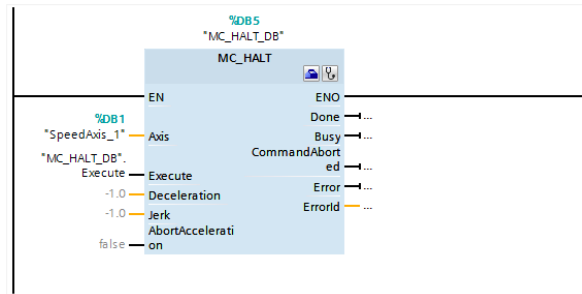
Obsługa programowa obiektu technologicznego *TO_SpeedAxis* wymaga zastosowania funkcji podstawowych interfejsu użytkownika.

Funkcje *Motion Control* znajdują się w przyborniku instrukcji → zakładka *Technology* → *Motion Control*.

W każdej z funkcji konieczne jest wskazanie obiektu technologicznego (można przeciągnąć go metodą *drag&drop* do parametru *Axis*) oraz podłączenie sygnału aktywującego na wejście funkcji *Enable/Execute*.

MC_Power

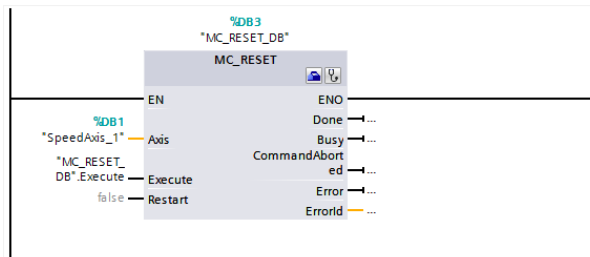
Aktywacja osi – przełączenie obiektu technologicznego w tryb pracy (załączenie przekształtnika częstotliwości).



Oś jest aktywna gdy wartość parametru *Enable* = 1.

MC_Reset

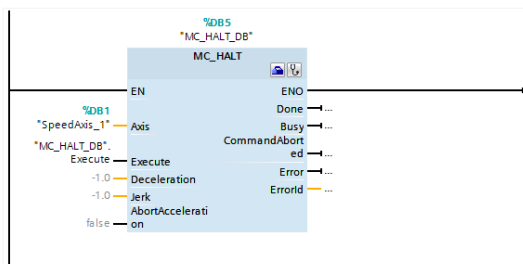
Pokwitowanie alarmów obiektu technologicznego. Funkcja potwierdza również bity błędów oraz ostrzeżeń w technologicznym bloku danych obiektu.



Funkcja aktywowana jest zboczem narastającym parametru *Execute*. Aby funkcja mogła zostać wykonana, wymagana jest aktywacja osi (*MC_Power*).

MC_Halt

Hamowanie osi – spowalnianie do momentu osiągnięcia statusu zatrzymania (*stanstill*).



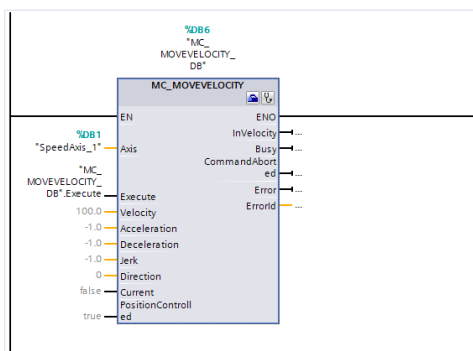
Funkcja aktywowana jest zboczem narastającym parametru *Execute*. Dynamikę hamowania określamy przez parametry *Deceleration* oraz *Jerk*. Jeśli zostawimy ustawienia na wartości domyślnej (-1), system pobierze specyfikę dynamiki z ustawień obiektu technologicznego.

9.3 Ruch z zadaną prędkością

Określiśmy już podstawowe funkcje obsługi obiektu technologicznego – aktywację osi, potwierdzenie błędów napędu oraz zatrzymanie osi. Teraz przejdźmy do funkcji, określające parametry ruchu naszej osi w trybie pracy aplikacji. Proste zadawanie prędkości możemy zrealizować przez dwie podstawowe funkcje użytkownika.

MC_MoveVelocity

Podstawowa funkcja zadawania prędkości z określoną dynamiką. Podobnie jak poprzednio, w parametrze *Axis* wskazujemy oś, którą chcemy sterować, natomiast zbocze narastające podane na wejście funkcji *Execute* spowoduje uruchomienie ruchu osi z określoną prędkością (*Velocity*) – jednostka określona została w parametrach obiektu technologicznego, jeśli przykładowo wybraliśmy 1/min, podanie wartości 100 spowoduje wygenerowanie stu obrotów na minutę.



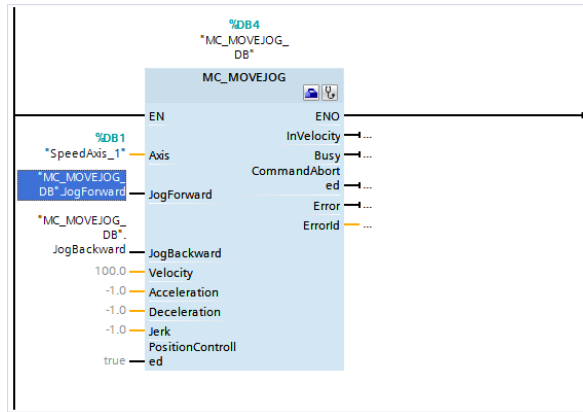
Wartość parametrów Acceleration/Deceleration/Jerk ustawiona na – 1 spowoduje pobranie dynamiki pracy osi z ustawień obiektu technologicznego.

Dodatkowo możemy określić kierunek ruchu (*Direction*), a także reakcję osi na zmiany parametru *Velocity (Current)*. Więcej informacji na temat parametrów funkcji oraz ich wartości odszukać można w tematach pomocy systemowej.

Zatrzymanie osi wykonujemy przez podanie wartości 0 na parametr *Velocity* oraz ponowne wygenerowanie zbocza na wejście *Execute* lub przez wywołanie funkcji *MC_Halt*.

MC_MoveJog

Funkcja zadawania prędkości w trybie Jog. Główna różnica w stosunku do *MC_MoveVelocity* polega na tym, że możemy przez dwa wejścia aktywujące (*JogForward/JogBackward*) podać kierunek obrotu osi. Wejścia aktywujące działają tylko gdy podany sygnał ma wartość 1, zmiana na wartość 0 powoduje zatrzymanie osi.

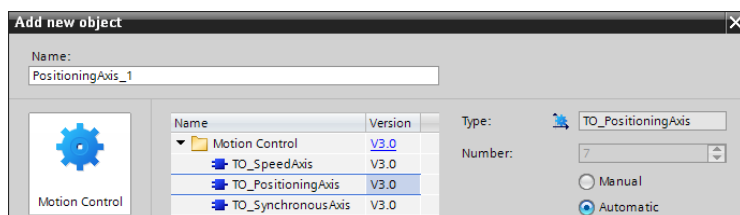


10 Pozycjonowanie względne

Wiemy już jak zadawać prędkość osi przez obiekt technologiczny *TO_SpeedAxis*, czyli wykonaliśmy zarządzanie napędem w otwartej pętli sterującej. W kolejnym kroku przejdźmy już do typowej funkcjonalności serwomechanizmu, czyli pozycjonowania. Pozycjonowanie jest sterowaniem w pętli ze sprzężeniem zwrotnym (pozycja z enkodera) gdzie parametrem naszej osi nie będzie bezpośrednio prędkość jej ruchu, a przemieszczenie. Będziemy chcieli więc przesunąć nasz obiekt o zdefiniowaną liczbę jednostek odległości lub na określoną pozycję. Te dwa zadania serwomechanizmu określane są pozycjonowaniem względnym (relatywnym) oraz absolutnym. W tej sekcji zajmiemy się omówieniem pierwszej z tych metod pozycjonowania osi.

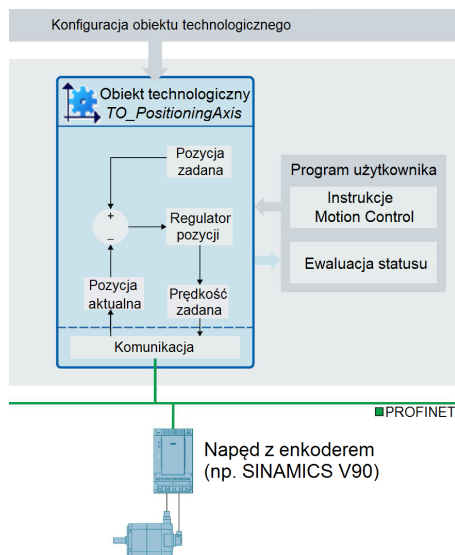
10.1 Konfiguracja obiektu technologicznego

W omawianej strukturze *Motion Control* – pozycjonowanie realizowane jest przez odrębny obiekt technologiczny, tzw. oś pozycjonującą. Dodajmy więc do naszego projektu nowy obiekt technologiczny MC – *TO_PositioningAxis*.



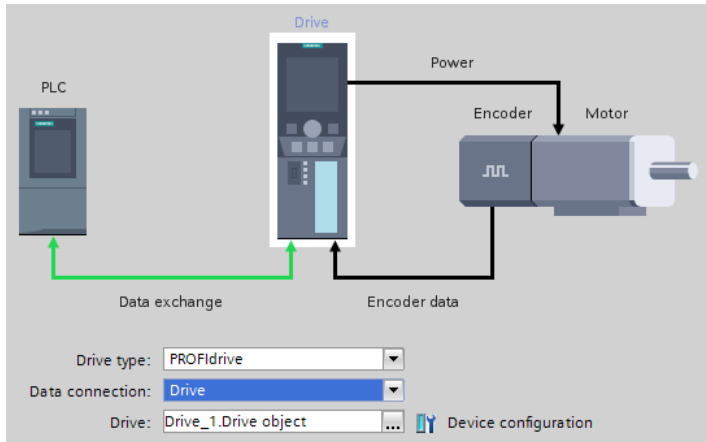
Obiekt technologiczny – oś pozycjonująca

Na poniższym schemacie przedstawiono zasadę działania obiektu technologicznego *TO_PositioningAxis*.



W porównaniu do wcześniej wykorzystanego obiektu *TO_SpeedAxis* od razu widać większą ilość parametrów, które możemy skonfigurować.

Zaczynamy podobnie jak w poprzednim przypadku od wskazania napędu, który chcemy zmapować do naszej konfiguracji jako oś pozycjonującą.



Tym razem również wykorzystamy komunikację PLC<>napęd przez telegramy *PROFIdrive* czyli przez magistralę komunikacyjną PROFINET (alternatywnie można wykonać komunikację przez sygnał analogowy). Sprzężenie zwrotne (sygnał z enkodera pozycji) tworzy przekształtnik częstotliwości. Oś pozycjonująca nie może być równocześnie połączona z napędem zdefiniowanym jako inny obiekt technologiczny.

W konfiguracji obiektu technologicznego definiujemy podstawowe własności mechaniczne naszego systemu (*Basic parameters*) – oś obrotowa/liniowa, jednostki pomiarowe przemieszczenia/prędkości/momentu/siły, ewentualnie definiujemy oś modułu (stały powtarzający się zakres pozycji – np. 0-360° dla osi rotacyjnej).

W zakładce *Extended parameters* określamy szczegółowo parametry mechaniczne układu – przekładnię, lokalizację enkodera, czy (dla osi liniowej) przemieszczenie liniowe na obrót wału silnika. Podobnie jak w przypadku obiektu *TO_SpeedAxis* definiujemy domyślną dynamikę systemu oraz parametry zatrzymania awaryjnego.

Obszar *Limits* konfiguracji obiektu technologicznego pozwala zdefiniować ograniczenia programowe osi. Mamy tutaj do dyspozycji pozycje krańcowe osi (o czym dalej), maksymalne wartości dynamiki, czy ograniczenia momentu obrotowego.

Ustawienia bazowania systemu dostępne są w panelu *Homing*. Ogólnie rzecz biorąc bazowanie układu mechanicznego jest określeniem pozycji, na której znajduje się oś obiektu technologicznego w odniesieniu do mechaniki systemu. Oznacza to, że tworzymy bezpośrednią (znaną) relację pomiędzy pozycją fizyczną osi, a obiektem technologicznym. Zadanie możemy zrealizować na kilka sposobów – programowo lub sprzętowo.

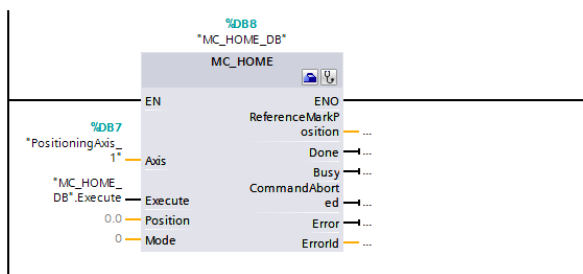
W przypadku osi pozycjonującej dochodzi nam zakres parametrów związanych z kontrolą pozycji (*Position monitoring*). Definiujemy tutaj dopuszczalne opóźnienia czasowe przy zadaniach pozycjonowania, a także błędy nadążania (*Following error*) czyli akceptowalne odchylenie pozycji aktualnej od zadanej, czy też parametry osiągnięcia statusu zatrzymania osi (*Standstill signal*).

Końcowym etapem konfiguracji osi pozycjonującej jest dostosowanie parametrów pętli sterującej.

10.2 Podstawowe funkcje interfejsu użytkownika

Opisane przy konfiguracji osi *TO_SpeedAxis* funkcje *MC_Power*, *MC_Reset* oraz *MC_Halt* znajdują również analogiczne zastosowanie w przypadku osi pozycjonującej. Dodatkowo pojawia się nam tutaj funkcja służąca do bazowania systemu.

MC_Home

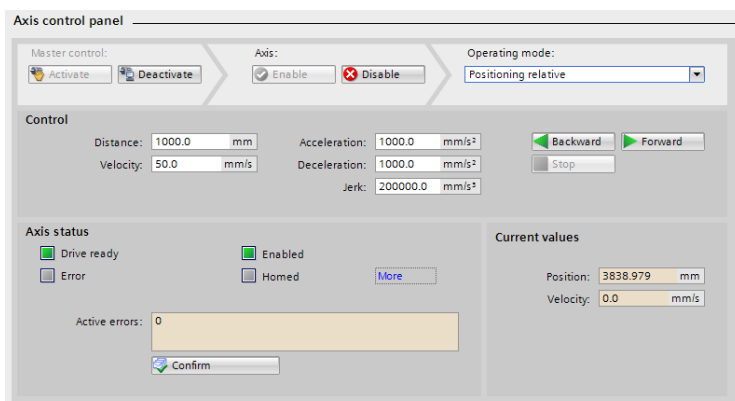


Bazowanie systemu (tzw. *homing*) jest funkcją tworzącą relację pomiędzy fizyczną pozycją osi, a jej pozycją w programie sterownika. Ustawienie układu mechanicznego w określonej pozycji, a następnie wywołanie funkcji bazowania, pozwoli precyzyjnie monitorować pozycję osi oraz pozycjonować układ z poziomu nadrzędnego systemu sterowania.

Pozycjonowanie relatywne nie wymaga użycia funkcji *MC_Home*. W przypadku pozycjonowania absolutnego (o czym w dalszej części) – system musi zostać wybazowany, aby zrealizowane mogły być funkcje pozycjonujące. Więcej informacji na temat trybów bazowania systemu znajdują się w sekcji dotyczącej pozycjonowania absolutnego.

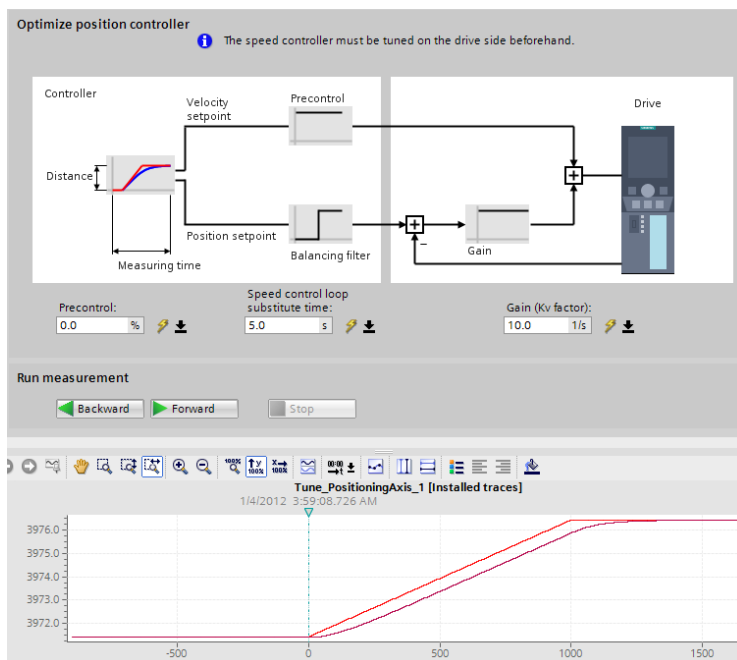
Na chwilę obecną wystarczy nam funkcja, która wyzeruje nam aktualną pozycję systemu w programie użytkownika. Zgodnie z powyższą ilustracją, podłączamy – analogicznie jak w przypadkach pozostałych funkcji podstawowych – naszą oś oraz wywołujemy funkcję *MC_Home* podając zbocze narastające sygnału binarnego na wejście bloku *Execute*. Wartość aktualnej pozycji zostanie ustawiona dla obiektu technologicznego zgodnie z wartością podaną na wejście *Position*. W naszym przypadku może być to pozycja 0.

Podobnie jak w przypadku osi sterowania prędkością, w obiekcie technologicznym osi pozycjonującej, również mamy do dyspozycji panel uruchomienia oraz testowania funkcji osi z poziomu obiektu technologicznego.

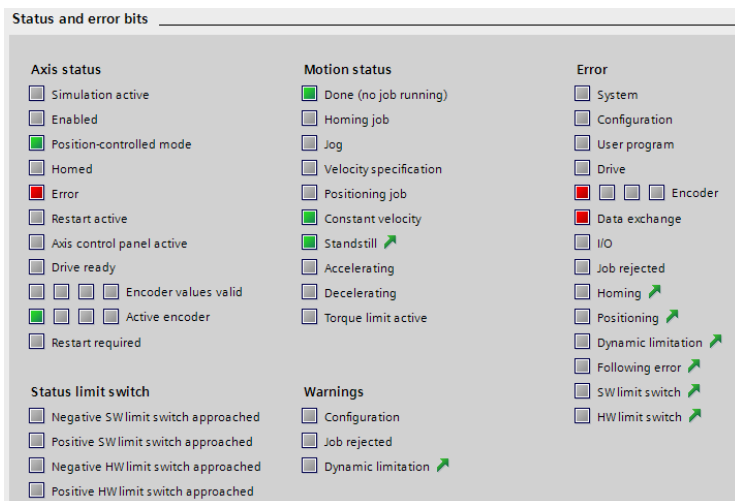


Poza zadawaniem prędkości możemy wykonać testowanie funkcji bazowania systemu, a także pozycjonowania. Bezpośrednio z panelu *Comissioning* zyskujemy również dostęp do wartości aktualnych parametrów osi (pozycja/prędkość).

W zakładce *Tuning* możemy dostosować dynamikę naszego układu poprzez modyfikację parametrów regulatora pozycji – przy założeniu, że regulator prędkości został zoptymalizowany po stronie konfiguracji napędu. Możemy przykładowo porównać wartość pozycji zadanej w stosunku do aktualnej.



Panel *Diagnostics* – dostępny dla osi pozycjonującej – podobnie jak w przypadku obiektu technologicznego *TO_SpeedAxis*, dostarcza informacji statusowych oraz informuje o błędach pracy układu.



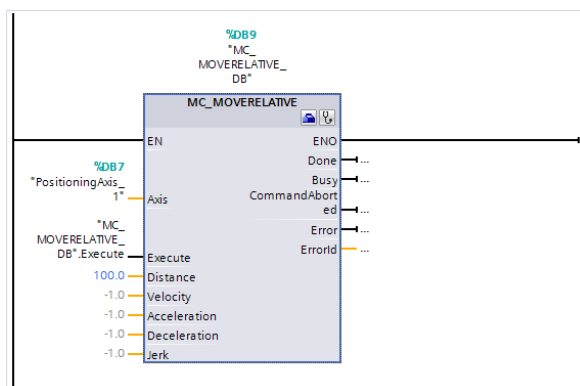
Ze względu na bardziej zaawansowaną charakterystykę obiektu – zakres parametrów dostępnych dla osi pozycjonującej jest szerszy.

10.3 Pozycjonowanie

Zgodnie ze wstępem bieżącego rozdziału – pozycjonowaniem względnym nazywamy funkcję, która analizując pozycję osi fizycznej (zmapowaną na obiekt technologiczny w programie sterownika) pozwoli na przemieszczenie jej o określoną liczbę jednostek. Jednostki odległości wynikają z konfiguracji mechaniki systemu w obiekcie technologicznym (*Extended parameters* → *Mechanics*).

MC_MoveRelative

Funkcją *Motion Control* realizującą powyższe zadanie jest *MC_MoveRelative*.



Parametrem wejściowym jest obiekt technologiczny (*Axis*), a także sygnał aktywujący *Execute* – zboczne narastające. Parametry przemieszczenia osi definiujemy przez podanie: odległości przejazdu, prędkości, przyspieszenie oraz zrywu (*Distance*, *Velocity*, *Acceleration*, *Deceleration* oraz *Jerk*). Dynamika przejazdu określona przez wartości – 1.0 podobnie jak w poprzednim przykładzie – spowoduje pobranie wartości domyślnych z ustawień obiektu technologicznego.

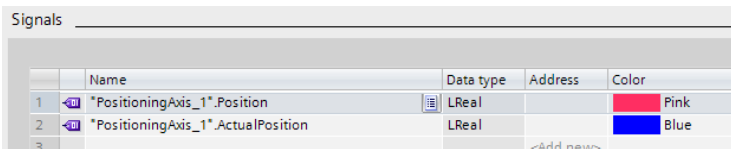
10.4 Trace

Zarówno na etapie uruchomienia i parametryzacji systemu, jak i podczas zadań diagnostycznych, bardzo przydatna (zwłaszcza w przypadku szybkozmiennych parametrów układów MC) jest standardowa funkcja sterownika S7-1500 – *Trace*.

Funkcja ta pozwala nagrywać wartości zmiennych, czy też parametry obiektów technologicznych w sposób ciągły oraz bardzo dokładny – z cyklem dowolnego bloku OB. Oznacza to, iż nawet w przypadku dynamicznych aplikacji sterowania ruchem (np. tryb IRT z cyklem 2ms) jesteśmy w stanie przeanalizować wartość dowolnego parametru w każdym cyklu programu. Przykładowo rejestrując wartości pozycji zadawanej przez interpolator w stosunku do aktualnej pozycji, na której znajduje się oś, możemy zaobserwować najdrobniejsze zmiany oraz reakcje rzeczywistego systemu na zmianę parametrów pętli regulacyjnej. Funkcja umożliwi nam również wykrywanie, wszelkich niewychwytywanych dla ludzkiego oka, właściwości mechanicznych urządzenia, czy też diagnostykę zużycia elementów, niepożądanych przeciążeń, etc.

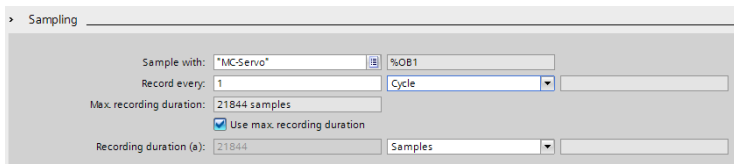
Aktywacja funkcji odbywa się przez zakładkę *Trace* w drzewku projektu → *Add new trace*.

W pierwszym kroku wybieramy parametry, które chcemy zarejestrować (np. pozycja osi zadana vs. aktualna).



	Name	Data type	Address	Color
1	"PositioningAxis_1".Position	LReal		Pink
2	"PositioningAxis_1".ActualPosition	LReal		Blue
+ Add new				

Następnie wskazujemy cykl nagrywania – w naszym przypadku może być to blok przerwań dla cyklu izochronicznego *MC_Servo*.



Sampling

Sample with: "MC-Servo" %OB1

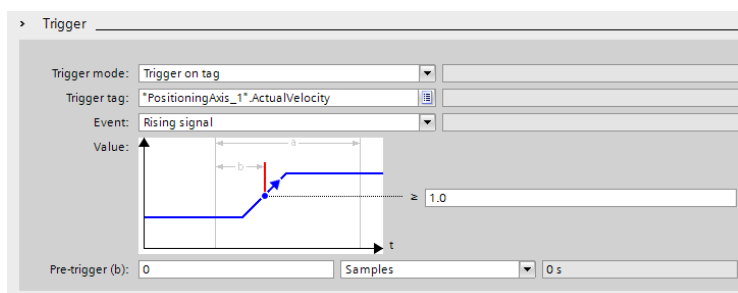
Record every: 1 Cycle

Max. recording duration: 21844 samples

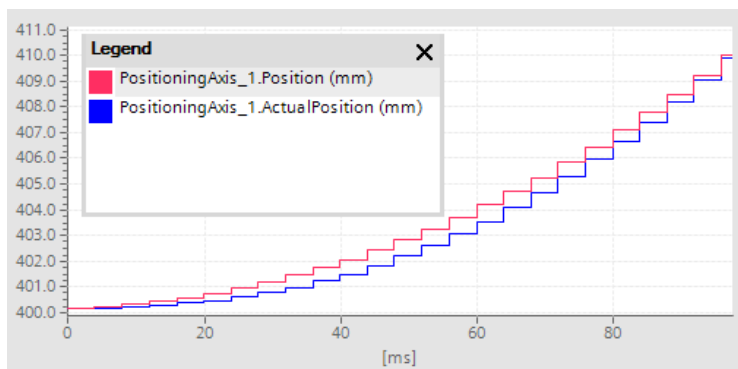
☒ Use max. recording duration

Recording duration (s): 21844 Samples

Dodatkowo określamy parametry wyzwolenia rejestratora.



Wgrywamy konfigurację do sterownika oraz aktywujemy funkcję Trace .
Przykładowy zarejestrowany przebieg może wyglądać następująco:



11

Położenia krańcowe osi

W przypadku obiektu technologicznego *TO_PositioningAxis* istotnym parametrem jest położenie skrajne osi w układzie mechanicznym. Aby nie spowodować uszkodzenia maszyny, bądź nie narazić produkcji na straty materiału, a jednocześnie zapewnić bezpieczeństwo pracownikom – warto skonfigurować ograniczenia dopuszczalnej pozycji osi. Położenie krańcowe osi możemy zdefiniować w konfiguracji obiektu technologicznego przez statyczne wartości liczbowe lub sprzętowo – przez sygnał z wejścia cyfrowego sterownika.

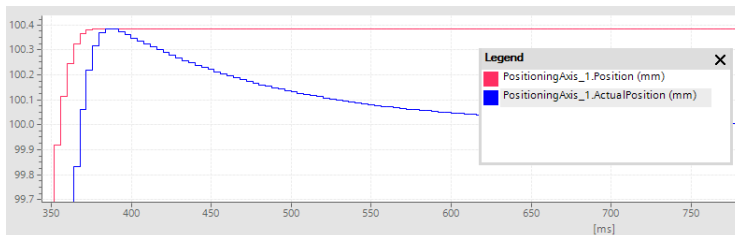
11.1 Ograniczenie programowe

Położenia krańcowe osi określamy w ustawieniach obiektu technologicznego *Configuration* → *Extended parameters* → *Limits* → *Position limits*.

Krańcówka programowa definiowana jest przez zakres aktualnej pozycji podanej w jednostce wynikającej z konfiguracji obiektu technologicznego. Przykładowo możemy określić dopuszczalny zakres pozycji jak pokazano poniżej:

Uwaga – programowe położenie krańcowe osi może zostać zastosowane tylko dla systemu, który uprzednio został wybazowany (funkcja *MC_Home*). Po bazowaniu systemu, aktualna pozycja musi zawierać się we wskazanym przedziale.

Przekroczenie wartości granicznej powoduje wywołanie błędu obiektu technologicznego o numerze 534 (który można odczytać z technologicznego bloku danych przez parametr *ErrorDetail.Number* lub w panelu diagnostycznym osi pozycjonującej). Błąd oznacza osiągnięcie programowego ograniczenia położenia osi. Obiekt technologiczny przejdzie w status błędu oraz zostanie zatrzymany zgodnie z maksymalnym określonym opóźnieniem wynikającym z konfiguracji osi. Jeśli pozycja krańcowa zostanie przekroczona, system automatycznie powróci do określonej granicy. Można zaobserwować to na przykładowym przebiegu *Trace* (zależność pozycji zadanej względem aktualnej), gdzie system natrafia na granicę programową:



Aby przywrócić system do pracy, należy wykonać potwierdzenie błędów napędu (np. przez funkcję *MC_Reset*). System może zostać ponownie przywrócony do pracy w kierunku zgodnym z dopuszczalnym limitem pozycji.

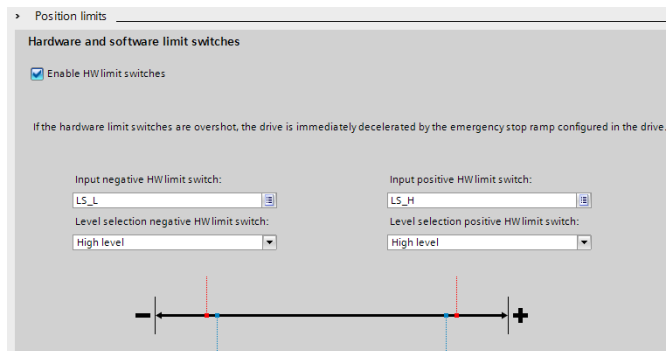
11.2 Ograniczenie sprzętowe

Alternatywą dla ograniczeń programowych jest zdefiniowanie krańcówek sprzętowych. Może być to przykładowo wyłącznik pozycyjny lub zbliżeniowy czujnik położenia, który podłączony będzie bezpośrednio pod wejście cyfrowe sterownika, napędu lub system rozproszonych wejść/wyjść.

W celu poprawnej identyfikacji sprzętowej położenia krańcowego osi, należy stosować mechanizmy wykrywania pozycji, które w momencie przekroczenia pozostaną w ustalonym stanie – w zależności od konfiguracji wejścia cyfrowe będą na stałe aktywne (zbrocze narastające) lub nieaktywne (zbrocze opadające).

Drugim istotnym aspektem wykrywania pozycji krańcowych przez sprzętowe wejścia cyfrowe jest ich dokładność – głównie chodzi o czas odczytu sygnału cyfrowego. Przy aplikacjach wolnozmiennych nie będzie to istotne – możemy zastosować standardowe wejścia cyfrowe z dowolnym opóźnieniem sygnału wejściowego (filtrowanie sygnału). W przypadku dynamicznych aplikacji serwo musimy wziąć pod uwagę: szybkość odczytu sygnału, czas filtrowania oraz ewentualnie dostępność komunikacji w trybie izochronicznym. Wybierając moduł wejść cyfrowych możemy uzyskać minimum 3.2 ms dla modułu standardowego, 0.05 ms dla modułu typu *High Feature*, lub nawet kilka mikrosekund w przypadku dedykowanych modułów technologicznych (*TM Timer*). Ten ostatni wspiera również pracę w trybie izochronicznym zapewniając, że sygnał zostanie zawsze przetworzony w jednym cyklu bloku przerwań *MC_Servo*. Wybór modułu zależy od konkretnych wymagań aplikacji.

Konfigurację wykonujemy podobnie jak w przypadku ograniczeń programowych – w ustawieniach obiektu technologicznego (*Extended parameters* → *Limits* → *Position Limits*)



Aktywujemy opcję sprzętowych położenia krańcowych osi. Następnie wskazujemy wejścia cyfrowe, które monitorowane będą pod kątem wystąpienia położenia granicznego – przykładowe zmienne *LS_L* oraz *LS_H* muszą być zaadresowanymi wejściami cyfrowymi, a co za tym idzie w konfiguracji sprzętowej projektu już na tym etapie musi znaleźć się odpowiedni moduł wejść cyfrowych. Dodatkowo dla każdego z ograniczników określamy, czy system reagować ma na zbrocze narastające (*High level*) czy opadające (*Low level*). Wykrycie określonego stanu na wyłączniku pozycyjnym spowoduje zatrzymanie osi zgodnie z ustawieniami dynamiki osi pochodzącymi z konfiguracji zatrzymania awaryjnego (*Extended parameters* → *Emergency stop*). Zatrzymanie osi zostanie również zwieńczzone błędem obiektu technologicznego o ID: 531 (*ErrorDetail.Number*), który sugeruje osiągnięcie sprzętowego ogranicznika położenia krańcowego osi.

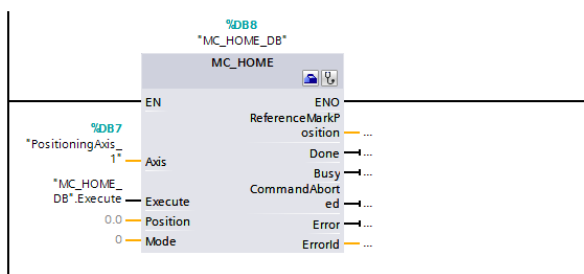
12 Bazowanie systemu

Bazowaniem systemu pozycjonującego (tzw. *homिंगiem*) nazywamy określenie znanej relacji pomiędzy fizycznym układem mechanicznym, a obiektem technologicznym w sterowniku. Wartość aktualnej pozycji zostaje przypisana do znacznika bazowania, który reprezentuje znaną pozycję mechaniczną. Bazowanie możemy zastosować w przypadku osi pozycjonującej, synchronicznej oraz dla zewnętrznego enkodera. Procedura musi zostać zrealizowana w przypadku zadań pozycjonowania absolutnego.

12.1 Funkcje interfejsu użytkownika

Opisana wstępnie w poprzednim rozdziale, funkcja *MC_Home* posiada szerokie możliwości w zakresie bazowania układu.

MC_Home



MC_Home jest funkcją tworzącą relację pomiędzy fizyczną pozycją osi, a jej pozycją w programie sterownika. Ustawienie układu mechanicznego w określonej pozycji, a następnie wywołanie funkcji bazowania, pozwoli precyzyjnie monitorować oraz pozycjonować układ z poziomu nadrzędnego systemu sterowania. Można również wymusić w pełni zautomatyzowane bazowanie systemu. W zależności od potrzeb danej aplikacji będziemy realizować odpowiedni tryb bazowania.

Pozycjonowanie relatywne nie wymaga użycia funkcji *MC_Home*. W przypadku pozycjonowania absolutnego (o czym w dalszej części dokumentu) – system musi zostać wybazowany, aby zrealizowane mogły być funkcje pozycjonujące.

Podobnie jak w przypadku pozostałych funkcji MC podstawowym parametrem funkcji *MC_Home* jest oś (*Axis*), dla której funkcja ma zostać zrealizowana. Inicjacja procedury odbywa się przez podanie zbocza narastającego na wejście *Execute*. Pozostałe parametry określają pozycję jaka ma zostać przypisana do obiektu technologicznego po zakończeniu bazowania (*Position*) oraz tryb bazowania układu (*Mode*). Wybrany tryb bazowania może zostać zastosowany dla wybranych obiektów technologicznych, o czym w dalszej części.

12.2 Status wybazowania systemu

Realizacja bazowania systemu może zostać wykonana na wiele sposobów, o czym w kolejnej sekcji. Każdy z algorytmów kończy się jednak tym samym efektem – obiekt technologiczny zyskuje status *homed*, czyli ustawiony zostaje bit statusowy, informujący, iż system mechaniczny znajduje się w znanej pozycji z punktu widzenia obiektu technologicznego. Status można odczytać z panelu diagnostycznego osi z poziomu środowiska inżynierskiego TIA Portal lub przez słowo statusowe obiektu technologicznego (*HomingDone*).

Istotną kwestią jest świadomość kiedy status *homed* zostaje skasowany. Poza tym, że możemy odczytać tę informację ze słowa statusowego (`<TO>.StatusWord.X5`) warto również wiedzieć, przy wystąpieniu jakich warunków system wyzeruje znaczący bit. Poniżej przedstawiamy wykaz zdarzeń pozbawiających obiekt technologiczny statusu wybazowania.

Obiekt technologiczny z enkoderem inkrementalnym:

- wywołanie funkcji *MC_Home* w trybie 3, 5, 8 lub 10,
- zakończenie procedury bazowania (chwilowe skasowanie statusu),
- błąd odczytu lub usterka enkodera,
- reset obiektu technologicznego,
- wyłączenie i ponowne uruchomienie zasilania CPU,
- skasowanie pamięci operacyjnej sterownika (memory reset),
- zmiana konfiguracji enkodera.

Obiekt technologiczny z enkoderem absolutnym:

- błąd czujnika lub usterka enkodera,
- wymiana jednostki centralnej sterownika,
- zmiana konfiguracji enkodera,
- przywrócenie CPU do ustawień fabrycznych,
- wgranie innego programu do sterownika.

12.3 Metody bazowania systemu

W zależności od wymogów aplikacji oraz możliwości konstrukcyjnych urządzenia bazowanie systemu możemy przeprowadzić w różnych trybach. Poniżej znajduje się podstawowy podział dostępnych procedur.

Bazowanie aktywne (Mode = 3, 5)

Tryb, który spowoduje wywołanie procedury bazowania systemu (poprzez fizyczne przemieszczenie osi) zgodnie z konfiguracją pobraną z ustawień obiektu technologicznego (*Extended parameters* → *Homing* → *Active homing*).

Zainicjowanie procedury bazowania przez funkcję *MC_Home* spowoduje przejście osi w fazę dojeżdżania (*approach*). Następnie po wykryciu znacznika bazowania system przejdzie w fazę *homingu* – przemieszczenie osi z określoną prędkością na pozycję zgodną z konfiguracją obiektu technologicznego.

Znacznikiem bazowania może być wewnętrzny znacznik zerowy enkodera lub impuls wejścia cyfrowego.

Wywołanie bazowania aktywnego automatycznie anuluje wszelkie aktywne komendy *Motion Control* w zakresie danej osi. Bazowanie aktywne może zostać zaaplikowane dla osi pozycjonującej lub synchronicznej z enkoderem inkrementalnym.

W karcie ustawień bazowania aktywnego określamy źródło znacznika zerowego.

Select the homing mode

☒ Use zero mark via PROFIdrive telegram

☐ Use zero mark via PROFIdrive telegram and reference cam

☐ Use homing mark via digital input

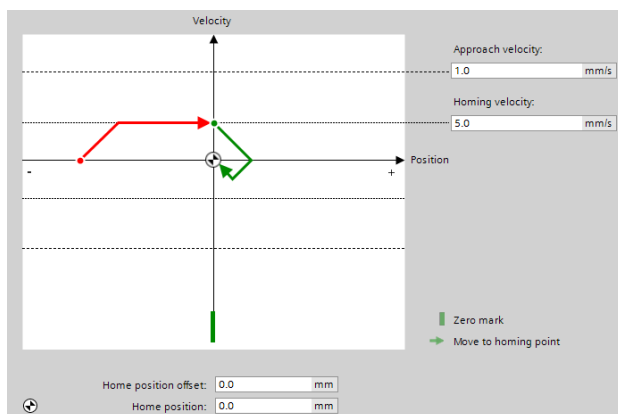
Digital input homing mark/cam: <Select tag>

Level selection: High level

☐ Enable direction reversal at HW limit switch

Use zero mark via PROFIdrive telegram

Wybór tej opcji oznacza, że znacznikiem bazowania będzie wskaźnik pozycji zerowej znajdujący się bezpośrednio na tarczy enkodera. Metoda ta pozwala osiągnąć dużą dokładność bazowania. W tym przypadku, procedura zostanie zrealizowana zgodnie z następującym przebiegiem:



Parametrami algorytmu są tutaj prędkości – **najazdu** (premieszczenie do momentu wykrycia znacznika) oraz **bazowania** (premieszczenie po wykryciu znacznika). Dodatkowo określić możemy offset pozycji domowej oraz jej wartość.

Home position offset

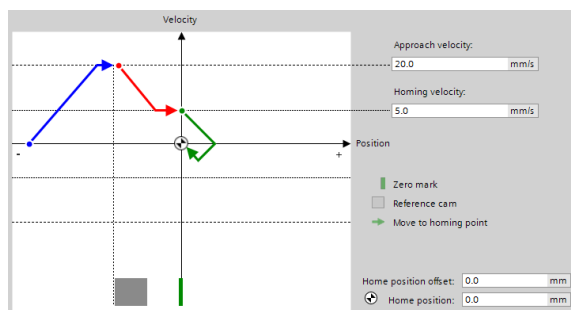
W momencie wykrycia znacznika zerowego aktualna pozycja obiektu technologicznego zostaje wyzerowana. Parametr *Home position offset* mówi nam o tym, na jaką pozycję po wykryciu znacznika oś ma się przesunąć (wartość dodatnia lub ujemna określa kierunek przesunięcia). Wartość parametru ma znaczenie zarówno w trybie *MC_Home* 3 jak i 5.

Home position

Parametr określa jaka zostanie przypisana aktualna pozycja obiektu technologicznego po zakończeniu procedury bazowania. Parametr ma znaczenie tylko w przypadku trybu *MC_Home* = 5. W przypadku trybu 3, wartość pozycji aktualnej po zakończeniu procedury bazowania zostanie pobrana z parametru *Position* funkcji *MC_Home*.

Use zero mark via PROFIdrive telegram and reference cam

Wybierając tę metodę bazowania systemu uzyskamy efekt podobny jak w poprzednim przypadku, aczkolwiek dodatkowo analizowany będzie sygnał binarny.

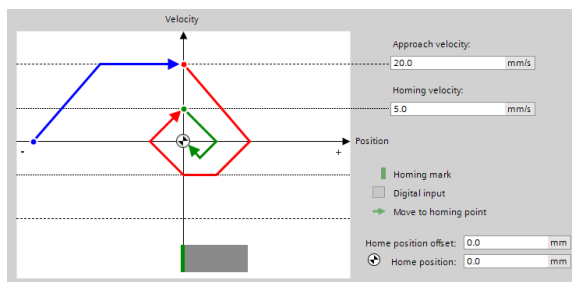


Aktywacja funkcji *MC_Home* (w trybie 3 lub 5) spowoduje przejście obiektu technologicznego w tryb poszukiwania (**approach**) znacznika referencyjnego (*Reference cam*), czyli fizycznego wejścia binarnego definiowanego przez zmienną w polu *Digital input homing mark/cam*. Gdy sygnał zostanie aktywowany (w zależności od konfiguracji zboczem narastającym lub opadającym – *Level selection*) np. sygnałem 1 – obiekt technologiczny przejdzie w tryb bazowania (**homing**). Tryb ten będzie trwał do momentu powrotu sygnału binarnego do stanu źródłowego (np. 0). Po zakończeniu trwania impulsu, na wejściu binarnym system zacznie poszukiwanie (dalej w trybie bazowania) znacznika zerowego enkodera. Wykrycie znacznika spowoduje **zakończenie procedury homingu** zgodnie z ustawieniami bazowania obiektu technologicznego.

Parametry *Home position offset* oraz *Home position* mają takie samo znaczenie jak w przypadku procedury poprzedniej – bez ingerencji zewnętrznego wejścia cyfrowego sterownika.

Use homing mark via digital input

Ostatni z trybów bazowania aktywnego uruchamiany jest wyłącznie przez funkcję *MC_Home* oraz wejście cyfrowe sterownika – znacznik zerowy enkodera nie jest tutaj analizowany. Zgodnie z systemowym schematem procedura może wyglądać następująco (może również wyglądać identycznie jak w poprzednim przypadku – zależy to od ustawień kierunków poszukiwania znaczników):



Aktywując procedurę przez funkcję *MC_Home* (tryb 3 lub 5) algorytm przechodzi w tryb poszukiwania (*approach*) sygnału binarnego (w zależności od ustawień aktywującym będzie zbocze narastające lub opadające). Ta faza trwa do momentu wykrycia odpowiedniego stanu wejścia binarnego, jeśli zostanie wykryte zbocze aktywuje się tryb bazowania (*homing*). Powrót do stanu początkowego sygnału binarnego aktywuje **zakończenie procedury bazowania**.

Parametry *Home position offset* oraz *Home position* mają takie samo znaczenie jak w przypadku procedur poprzednich.

Kierunek najazdu, poszukiwania znacznika oraz bazowania definiujemy dla każdego z powyżej opisanych trybów przez adekwatne ustawienie kierunku pracy osi w danej fazie – pozytywny lub negatywny, czyli zgodny z przyrostem wartości pozycji lub jej zmniejszaniem.

 Approach direction	 Homing direction	 Homing mark
☒ Positive	☒ Positive	☐ Positive side
☐ Negative	☐ Negative	☒ Negative side

Enable direction reversal at HW limit switch

Znacznik ten określa zachowanie systemu w sytuacji, gdy skonfigurowane są sprzętowe krańcowe położenia osi. Aktywacja krańcówki, w przypadku aktywacji powyższej opcji, spowoduje odwrócenie kierunku jazdy osi – podobnie jak wykrycie znacznika bazowania. Gdy opcja nie będzie aktywna, a skonfigurowane sprzętowe krańcowe położenie osi zostanie aktywowane podczas procedury bazowania – napęd zatrzyma się i zgłosi błąd.

Bazowanie pasywne (Mode = 2, 8, 10)

W przeciwieństwie do bazowania aktywnego – bazowanie pasywne nie powoduje fizycznego przemieszczenia osi. W tym trybie ruch osi inicjowany jest przez funkcję interfejsu użytkownika. Algorytm bazowania wywołany jest zgodnie z konfiguracją pobraną z ustawień obiektu technologicznego (*Extended parameters* -> *Homing* -> *Passive homing*).

Zainicjowanie procedury bazowania przez funkcję *MC_Home* spowoduje przejście osi w fazę dojeżdżania (*approach*). Ruch osi generowany jest przez funkcję użytkownika w programie sterownika. W momencie wykrycia znacznika bazowania system wykona *homing*, czyli – w przypadku bazowania pasywnego – przypisanie odpowiedniej wartości pozycji (wynikającej z konfiguracji obiektu technologicznego oraz trybu bazowania) do pozycji aktualnej osi.

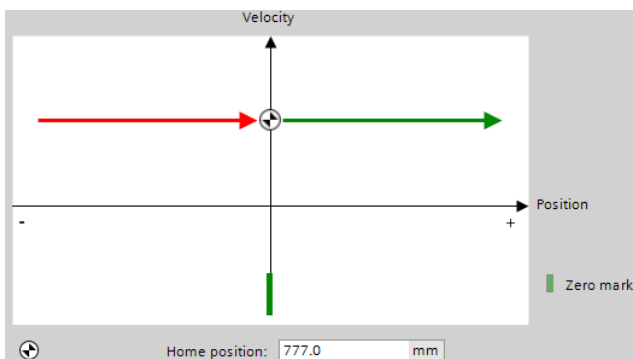
Znacznikiem bazowania może być wewnętrzny znacznik zerowy enkodera lub impuls wejścia cyfrowego.

Wywołanie bazowania pasywnego nie powoduje anulowania aktywnych komend zadanej prędkości lub pozycji w zakresie danej osi. Bazowanie pasywne może zostać zaaplikowane dla osi pozycjonującej lub synchronicznej z enkoderem inkrementalnym lub dla zewnętrznego enkodera inkrementalnego.

W karcie ustawień bazowania aktywnego określamy źródło znacznika zerowego – analogicznie jak w poprzednim przypadku, do wyboru mamy te same opcje.

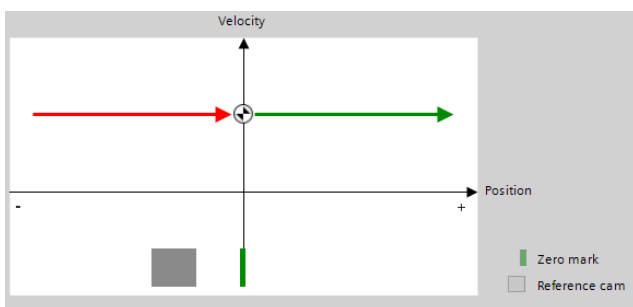
Use zero mark via PROFIdrive telegram

Aktywacja bazowania pasywnego powoduje przejście osi w tryb poszukiwania znacznika zerowego enkodera (*approach*). Najeżdżenie osi na znacznik spowoduje wykonanie *homingu*, czyli w tym przypadku przypisanie do aktualnej pozycji obiektu technologicznego wartości parametru *Position* funkcji *MC_Home* (tryb 8) lub wartości parametru *Home position* (tryb 10) określonego w konfiguracji bazowania pasywnego obiektu technologicznego.



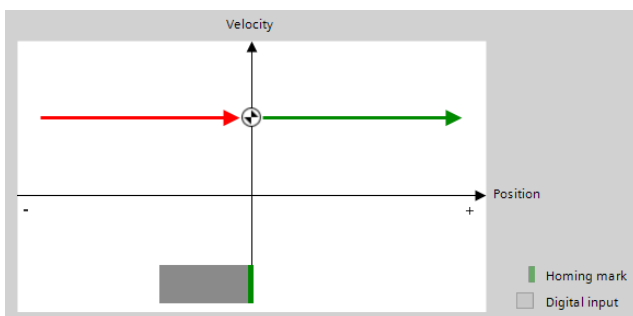
Use zero mark via PROFIdrive telegram and reference cam

W tym przypadku algorytm działa analogicznie jak przy bazowaniu aktywnym – aktywacja funkcji *MC_Home* w trybie bazowania pasywnego powoduje przejście systemu w fazę zbliżania (*approach*). Na tym etapie system oczekuje zbocza (zgodnie z konfiguracją obiektu technologicznego – narastającego lub opadającego) na określonym wejściu cyfrowym (*Digital input homing mark/cam*). Po jego wykryciu i powrocie do stanu początkowego (zakończenie impulsu), poszukiwany jest znacznik zerowy enkodera. W momencie jego detekcji, pozycja aktualna osi zostaje ustawiona na wartość określoną w konfiguracji bazowania pasywnego (tryb 10) lub na wartość parametru *Position* funkcji *MC_Home* (tryb 8).



Use homing mark via digital input

Bazowanie pasywne przez znacznik wejścia cyfrowego powoduje przypisanie wartości pozycji aktualnej określonej w konfiguracji bazowania pasywnego (tryb 10) lub na wartość parametru *Position* funkcji *MC_Home* (tryb 8). W tym wypadku znacznik zerowy enkodera nie ma znaczenia – istotne jest jedynie wskazane w konfiguracji wejście cyfrowe. W zależności od ustawień system oczekuje zbocza narastającego lub opadającego. Po zakończeniu trwania impulsu (powrót do stanu początkowego) system zostaje wybazowany.



Tryb 2

Wywołanie bazowania pasywnego przez funkcję *MC_Home* w trybie 8 lub 10 powoduje automatyczne skasowanie statusu *homed* obiektu technologicznego. Jedynie w trybie 2 (który funkcjonalnie działa identycznie jak tryb 8) status nie zostaje skasowany w momencie aktywacji bazowania.

Kierunek poszukiwania znacznika w trybie zbliżania dla każdego z powyżej opisanych trybów definiujemy przez adekwatne ustawienie kierunku pracy osi w danej fazie – pozytywny, negatywny lub aktualny. Poniższa konfiguracja pozwoli również określić, czy bazowanie z uwzględnieniem wejścia cyfrowego sterownika zostanie zrealizowane po zakończeniu trwania impulsu, czy przy jego rozpoczęciu.

➔ Homing direction ☐ Positive ☐ Negative ☒ Current	█ Homing mark ☐ Positive side ☒ Negative side
---	---

Programowe anulowanie bazowania pasywnego odbywa się przez wywołanie funkcji *MC_Home* w trybie 9.

Bazowanie bezpośrednie – relatywne (Mode = 1)

Bazowanie bezpośrednie relatywne nie uwzględnia detekcji żadnych fizycznych znaczników bazowania. Aktywne zadania pozycjonowania lub prędkości nie są anulowane. Wykonanie funkcji *MC_Home* w trybie 1 spowoduje bezpośrednie dodanie wartości parametru *Position* do aktualnej pozycji obiektu technologicznego.

Funkcja może zostać zastosowana dla osi pozycjonującej lub synchronicznej z dowolnym typem enkodera lub dla dowolnego enkodera zewnętrznego.

Bazowanie bezpośrednie – absolutne (Mode = 0)

Bazowanie bezpośrednie absolutne nie uwzględnia detekcji żadnych fizycznych znaczników bazowania. Aktywne zadania pozycjonowania lub prędkości nie są anulowane. Wykonanie funkcji *MC_Home* w trybie 0 spowoduje bezpośrednie przepisanie wartości parametru *Position* do aktualnej pozycji obiektu technologicznego.

Funkcja może zostać zastosowana dla osi pozycjonującej lub synchronicznej z dowolnym typem enkodera lub dla dowolnego enkodera zewnętrznego.

Pozycja enkodera absolutnego (Mode = 6, 7)

Podobnie jak opisany powyżej tryb bazowania bezpośredniego, bazowanie enkodera absolutnego odbywa się w dwóch trybach: relatywnym (Mode = 6 – pozycja aktualna zostaje przesunięta o wartość parametru *Position*) lub absolutnym (Mode = 7 – wartość parametru *Position* zostaje przypisana do pozycji aktualnej).

Różnica w stosunku do bazowania bezpośredniego polega na tym, że wyliczone przesunięcie aktualnej pozycji absolutnej zostaje zapisane w nieulotnej pamięci sterownika. Wartość tę odczytać można przez parametr obiektu technologicznego `<TO>.StatusSensor[n].AbsEncoderOffset`.

Funkcja może zostać zastosowana dla osi pozycjonującej lub synchronicznej z enkoderem absolutnym lub dla zewnętrznego enkodera absolutnego.

13

Pozycjonowanie absolutne

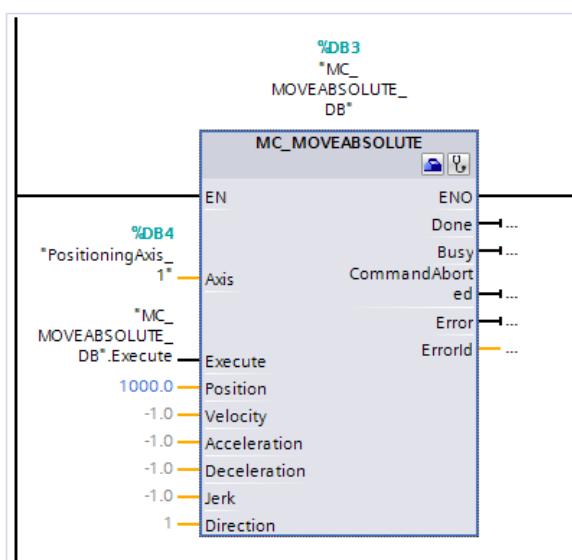
W rozdziale *Pozycjonowanie względne* dowiedzieliśmy się jak działa mechanizm pozycjonowania relatywnego, gdzie określaliśmy o jaką ilość jednostek odległości chcemy przemieścić nasz obiekt technologiczny. W przypadku pozycjonowania absolutnego zamiast określać odległość przejazdu, wskażemy systemowi bezpośrednio na jaką pozycję oś ma zostać przemieszczona.

13.1 Warunki dopuszczające

Aby można było zrealizować funkcję pozycjonowania absolutnego, należy spełnić podstawowe kryteria technologiczne. Przede wszystkim należy utworzyć odpowiedni obiekt technologiczny oraz wykonać jego parametryzację. Pozycjonowanie absolutne można zastosować dla osi pozycjonującej lub synchronicznej. Kolejnym krokiem jest wywołanie odpowiednich funkcji podstawowych interfejsu użytkownika w celu aktywacji osi. Dodatkowo (w przeciwieństwie do funkcji pozycjonowania relatywnego), aby pozycjonowanie absolutne było możliwe, system musi zostać wybazowany, czyli status *homed* obiektu technologicznego musi mieć wartość 1 – zgodnie z opisem poprzedniego rozdziału. Metoda bazowania jaką zastosujemy, nie ma znaczenia z punktu widzenia obiektu technologicznego.

13.2 Funkcje użytkownika

Pozycjonowanie absolutne realizowane jest przez systemową funkcję *Motion Control* – *MC_MoveAbsolute*. Funkcja dostępna jest bezpośrednio z biblioteki systemowej środowiska TIA Portal (*Instructions* → *Technology* → *Motion Control*).



Parametryzację wykonujemy podobnie jak w przypadku opisanych uprzednio funkcji. Pod parametr *Axis* podpinamy obiekt technologiczny. Parametr *Position* definiuje na jaką pozycję oś ma zostać przesunięta. Parametry przejazdu określamy przez pozostałe wejścia funkcji (prędkość, przyspieszenie, zryw oraz kierunek przejazdu). Pozostawienie na wejściach *Velocity*, *Acceleration/Deceleration* oraz *Jerk* wartości domyślnej (-1.0) spowoduje pobranie właściwości dynamiki bezpośrednio z ustawień obiektu technologicznego.

Kierunek przemieszczenia osi (*Direction*) znajduje zastosowanie tylko w przypadku obiektu technologicznego, gdzie skonfigurowana została oś moduło. Oznacza to, że pozycja obiektu jest cyklicznie kasowana przez system, tak aby jej wartość nie wychodziła poza ramy określonych wartości – np. oś rotacyjna z wartościami pozycji w przedziale od 0 do 360 stopni. W przypadku osi moduło kierunek oznacza przemieszczenie zgodnie z przyrostem wartości (*Direction* = 1) lub przeciwnie (*Direction* = 2). Można również wybrać przejazd najkrótszą drogą (*Direction* = 3).

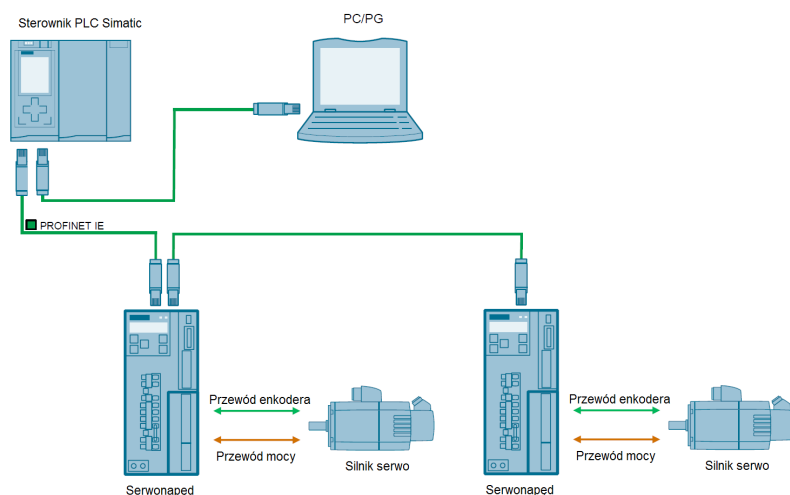
Wywołanie pozycjonowania odbywa się przez podanie zbocza narastającego na wejście *Execute*.

IV. Ruch skoordynowany

W kolejnym rozdziale zajmiemy się omówieniem zagadnień synchronizacji wieloosiowych układów napędowych. System pracuje w strukturze scentralizowanej *Motion Control*, czyli w takiej, gdzie sterownik PLC będzie nadrzędnym urządzeniem wydającym polecenia bezpośrednio do napędów. Dzięki interfejsowi PROFINET, sterownik będzie kontrolerem I/O (*master*), natomiast napędy będą urządzeniami typu I/O device (*slave*).

14 Opis systemu

Poniższa grafika ilustruje rozpatrywany system.



W zależności od wymaganej funkcjonalności, struktura systemu będzie wymagała jednostki nadrzędnej obsługującej odpowiednie procedury. W niniejszym przykładzie zastosujemy zaawansowaną jednostkę technologiczną S7-1511T, aczkolwiek zadanie synchronizacji względnej można wykonać również oparciu o standardową jednostkę centralną z rodziny S7-1500.

Przekształtniki częstotliwości wykorzystane w konfiguracji to SINAMICS V90 z interfejsem PROFINET, natomiast silniki to dedykowane serwomotory SIMOTICS S-1FL6.

Programator to Field PG M5 z oprogramowaniem STEP 7 Professional V14.

15 Sprężenie relatywne osi

Elektroniczna synchronizacja osi, w wieloosiowych systemach napędowych z nadrzędnym układem sterowania, oznacza doprowadzenie układu do pracy w określonej zależności prędkości i/lub pozycji pomiędzy wybranymi osiami – wiodącą oraz nadążną. Układy, które historycznie były realizowane przykładowo przez mechaniczne sprzężenie przekładnią, śrubą, krzyżem maltańskim, bądź dyskiem krzywkowym, w nowoczesnych systemach sterowania ruchem implementowane są przez funkcje programowe nadrzędnego systemu sterowania. Pozwala to na modyfikację oraz dostosowanie parametrów sprzężenia w sposób swobodny (również w locie) oraz minimalizuje ilość elementów mechanicznych, które przez swoją nieidealność oraz brak odporności na zużycie, generują wyższy koszt utrzymania aplikacji. Są również przyczyną zmniejszonej dokładności pracy urządzeń przez mechanikę przeniesienia napędu z silnika na element roboczy.

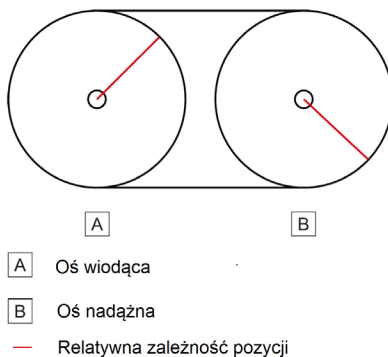
Sterownik technologiczny S7-1500T pozwala na realizację trzech podstawowych podejść do sprzężenia osi. W bieżącym oraz w kolejnych dokumentach rozważać będziemy synchronizację:

- liniową prędkościową (tzw. sprzężenie relatywne),
- liniową prędkościową z uwzględnieniem pozycji, w jakiej osie mają wejść w synchronizm (tzw. sprzężenie absolutne),
- krzywkową, czyli zależność pozycji dwóch osi, opisaną dowolną funkcją.

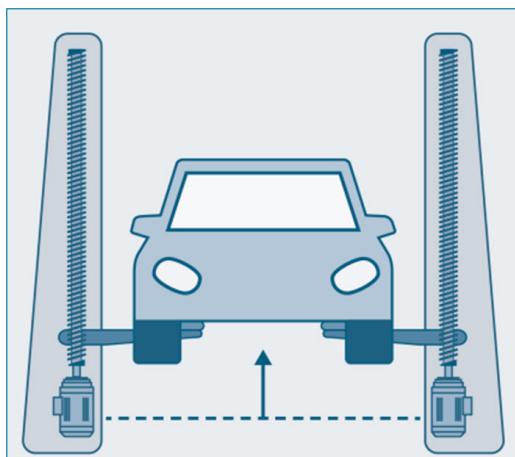
15.1 Obiekt technologiczny TO_Cam

W pierwszej kolejności zastanówmy się nad relatywną (względną) synchronizacją osi napędowych. Synchronizm względny oznacza wprowadzenie układu w tryb automatycznej pracy korygującej pozycję osi nadążnej względem osi wiodącej. System monitoruje oraz modyfikuje parametry osi aby zapewnić, że każda zmiana prędkości/ pozycji (zadanej lub aktualnej) osi master (*leading axis*) spowoduje podążanie za nią osi slave (*following axis*).

Sterownik nadrzędny S7-1500 odczytuje zmianę aktualnej pozycji osi wiodącej, mnoży ją przez elektroniczną przekładnię (może to być 1:1 lub inna konfigurowalna zależność), a następnie przekazuje jako wartość zadaną dla osi nadążnej.



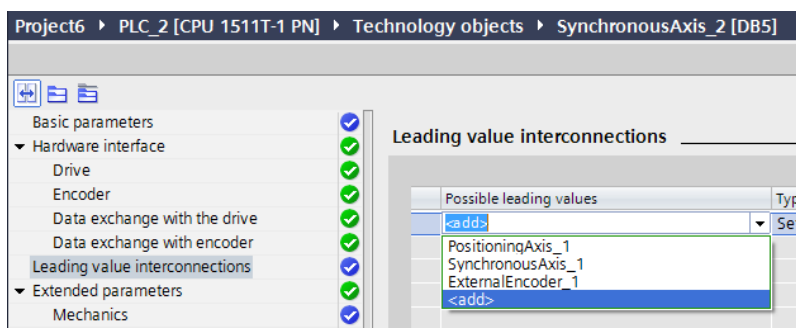
Charakterystycznym przykładem zastosowania sprzężenia relatywnego jest wieloosiowy napęd przenośnika taśmowego lub połączenie w ciąg wielu przenośników. Oczywiście przykładów zastosowania takiej funkcjonalności jest niezliczona ilość, np. podajniki, paletyzatory, manipulatory, bramy czy innego rodzaju aplikacje, gdzie układ wielu osi napędowych musi pracować w określonej relacji synchronicznej. Na załączonym obrazku widnieje przykładowa aplikacja synchronizacji relatywnej – podnośnik samochodowy.



15.2 Konfiguracja w TIA Portal

Od strony narzędzia inżynierskiego przede wszystkim musimy zapoznać się z nowym obiektem technologicznym. Znamy już oś regulacji prędkościowej w pętli otwartej (*speed axis*) oraz oś pozycjonującą (*positioning axis*) pracującą w układzie ze sprzężeniem zwrotnym.

Obiektem technologicznym, którego konfiguracja wymagana jest w celu sprzężenia dwóch osi, jest oś synchroniczna (*synchronous axis*), która w opisywanej relacji jest osią nadążną (może być również wiodącą dla innej osi nadążnej) – jest to oś pozycjonująca rozszerzona o możliwość podążania ścieżką innej osi. Parametryzacja tego obiektu różni się od poznanej osi pozycjonującej tylko jednym parametrem – wskaźnikiem do listy osi (*leading value interconnections*), z których – w programie użytkownika – jedna może zostać wybrana jako wiodąca.



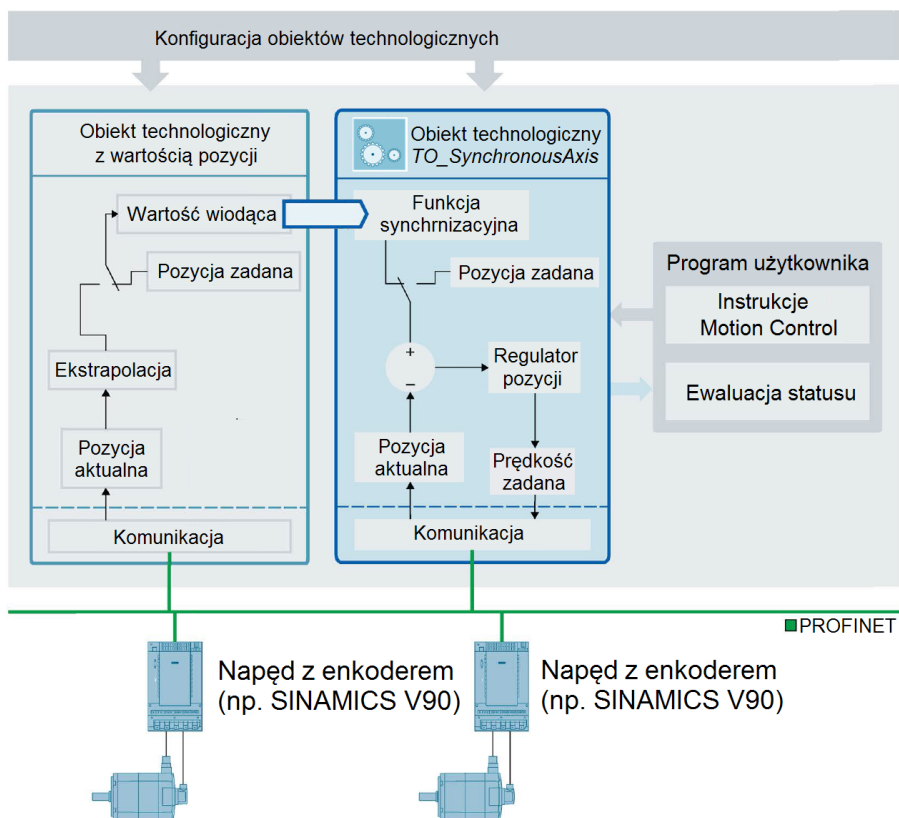
Osią wiodącą w takiej strukturze może być oś pozycjonująca, synchroniczna lub enkoder zewnętrzny. Należy zwrócić uwagę, że ze względu na to, że w synchronizmie analizowana jest pozycja osi wiodącej, nie może być to obiekt technologiczny typu speed axis.

Na liście osi wiodących wybieramy również typ sprężenia (*Setpoint* lub *Actual value*). Cechy tych dwóch opcji zostaną szczegółowo omówione w dalszej części dokumentu.

Osią wiodącą względnej pracy synchronicznej może być relatywna oś pozycjonująca, bądź synchroniczna (część istniejącego układu mechanicznego), może być to również oś wirtualna (wzorzec sygnału wiodącego generowany przez system nadrzędny) lub enkoder zewnętrzny (dodatkowy pomiar pozycji z istniejącej już części maszyny).

Obiekt technologiczny – oś synchroniczna

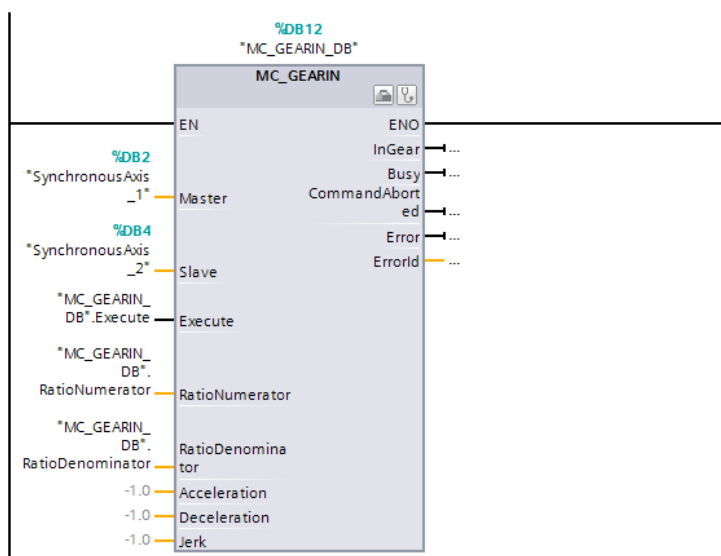
Poniższa ilustracja przedstawia schemat działania mechanizmu synchronizacji osi (obiektów technologicznych) przez nadrzędny sterownik S7-1500.



Interfejs użytkownika – funkcja *MC_GearIn*

Oś synchroniczna (podobnie jak prędkościowa bądź pozycjonująca) jest kompatybilna ze standardowym zestawem funkcji, które możemy zaaplikować bezpośrednio z poziomu programu użytkownika w celu jej aktywacji, zatrzymania, wyzazowania lub potwierdzenia błędów. Zastosowanie funkcji zgodnych ze standardem PLCOpen (*MC_Power*, *MC_Halt*, *MC_Home*, *MC_Reset*) nie wymaga analizowania ramki komunikacyjnej, odpowiednich bitów statusowych/sterujących czy metody komunikacji – za to odpowiada już prosta funkcja programowa.

Poza podstawową obsługą parametrów osi ważna jest funkcja, która służy do aktywacji sprzężenia – *MC_GearIn*.



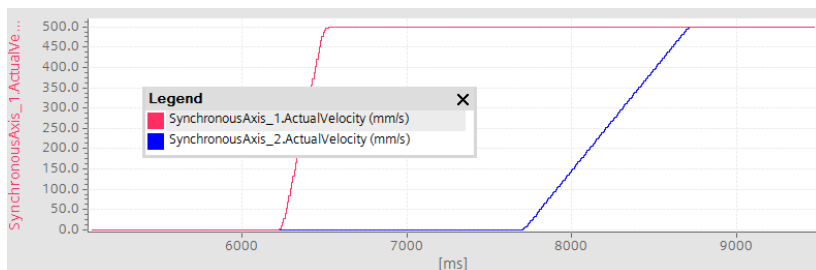
Pod parametry wejściowe *Master* oraz *Slave* podłączamy (np. poprzez przeciągnięcie – *drag&drop* – bezpośrednio z drzewka projektu) oś wiodącą oraz nadążną (synchroniczną). Podanie zbocza narastającego na wejście *Execute* spowoduje wywołanie sprzężenia osi.

W sytuacji, gdy aktywujemy funkcję *MC_GearIn*, podczas gdy oś wiodąca (lub nadążna) będzie w ruchu – współczynniki dynamiczne dojścia do sprzężenia relatywnego (dot. osi nadążnej) zostaną pobrane z parametrów funkcji *MC_GearIn* (*Acceleration*, *Deceleration*, *Jerk*). Jeśli wartość tych parametrów zostanie ustawiona na wartość 1, system pobierze domyślne wartości dynamiki z ustawień obiektu technologicznego (*Synchronous Axis* → *Configuration* → *Extended parameters* → *Dynamic default values*).

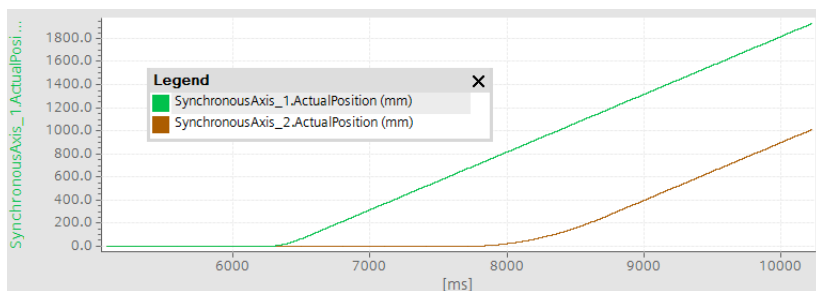
Jeśli funkcja *MC_GearIn* zostanie wywołana podczas postoju osi wiodącej oraz nadążnej, sprzężenie nastąpi natychmiastowo. Aby sprzężenie nastąpiło, wymagana jest aktywacja osi nadążnej (oś wiodąca nie musi być zasilona przez funkcję *MC_Power*).

Poniższy wykres prezentuje przykładowy przebieg wartości prędkości aktualnej osi wiodącej oraz nadążnej (w czasie) – przy dojściu do pracy synchronicznej. Oś *master* zostaje aktywowana oraz otrzymała komendę poruszania się ze stałą prędkością

500 mm/s. Niewiele ponad sekundę później wykonane zostało polecenie sprzężenia relatywnego dla osi synchronicznej.



Dla tego samego przypadku możemy porównać zależność pozycji osi **naładźnej** względem **wiodącej**.



Wykres zależności pozycji prezentuje przesunięcie fazowe wartości pozycji (ok. 900 mm) po dojściu do synchronizacji, które jest stałe w przypadku sprzężenia relatywnego i uwarunkowane momentem sprzężenia (wartościami aktualnymi pozycji obu osi). Aby usunąć przesunięcie fazowe, możemy zastosować inny typ synchronizacji: absolutna (o czym w następnym rozdziale) lub wykorzystać funkcje systemowe realizujące przesunięcie fazowe osi pracujących w synchronizmie (dalsza część dokumentu – *Przesunięcie w ruchu synchronicznym*). Rozbieżność pozycji można ewentualnie skorygować manualnie – przez ustawienie osi na wymaganych pozycjach (mechanicznie lub przez funkcje programu użytkownika) na postoju, a następnie wykonanie sprzężenia przed uruchomieniem trybu pracy systemu.

Interfejs funkcji `MC_GearIn` posiada opcjonalnie możliwość wprowadzenia przekładni w jakiej mają poruszać się osie sprzężone w ruchu synchronicznym – *RatioNumerator* oraz *RatioDenominator*. Czyli określamy ile obrotów ma wykonać oś naładźna na określoną ilość obrotów osi wiodącej. Przykładowo ustawiając przełożenie na wartość 1:3 – oś naładźna będzie pokonywać trzykrotność dystansu odczytanego z pozycji (zadanej lub aktualnej) osi wiodącej.

Wyjścia funkcji `MC_GearIn` dostarczają informacji na temat statusu sprzężenia (sprężony, zajęty, procedura przerwana) oraz ewentualnych błędów.

Odsprężenie osi nadążnej

Status sprężenia osi synchronicznej (*MC_GearIn.InGear* oraz *MC_GearIn.Busy*) pozostaje aktywny do momentu zakończenia pracy synchronicznej. Odsprężenie osi nadążnej następuje w momencie wywołania dowolnej innej funkcji *Motion Control* na jej obiekcie technologicznym (*TO_SynchronousAxis*), np. przez zatrzymanie, bazowanie (tylko aktywne) bądź zadanie prędkości, czy pozycjonowanie. Błąd osi nadążnej również spowoduje zakończenie pracy synchronicznej.

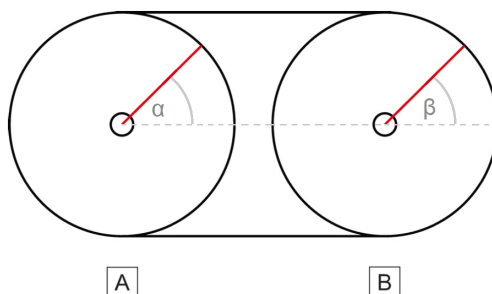
16 Sprężenie absolutne osi

16.1 Synchronizacja absolutna

Nieco bardziej zaawansowaną metodą sprężenia osi technologicznych jest synchronizacja absolutna. Synchronizm absolutny oznacza wprowadzenie układu w tryb automatycznej pracy korygującej pozycję osi nadążnej względem osi wiodącej z uwzględnieniem relacji pozycji pomiędzy nimi w momencie dojścia do sprężenia. Możemy powiedzieć, że wykonujemy synchronizację relatywną ze wskazaniem pozycji osi wiodącej oraz nadążnej, w których mają się znaleźć w momencie, w którym wejdą w synchronizm.

Po zaspężeniu, podobnie jak w przypadku synchronizacji relatywnej, system monitoruje oraz modyfikuje parametry osi, aby zapewnić, że każda zmiana prędkości/ pozycji (zadanej lub aktualnej) osi *master* (*leading axis*) spowoduje podążanie za nią osi *slave* (*following axis*).

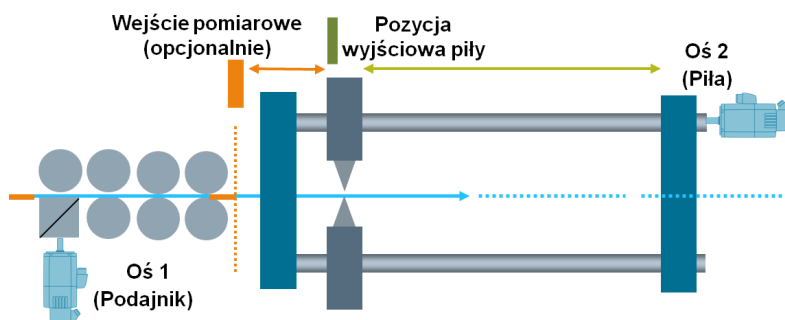
W przeciwieństwie do synchronizmu relatywnego – sprężenie absolutne możliwe jest tylko przez nadrzędny sterownik klasy S7-1500T z rozszerzonymi funkcjami *Motion Control*. Sterownik nadrzędny odczytuje zmianę aktualnej pozycji osi wiodącej, mnoży ją przez elektroniczną przekładnię (może być to 1:1 lub inna konfigurowalna zależność), a następnie przekazuje jako wartość zadaną dla osi nadążnej.



- A Oś wiodąca
- B Oś nadążna
- Absolutna zależność pozycji

Powyższa grafika ilustruje przykład synchronizacji absolutnej osi obrotowej, gdzie możemy przyjąć, że pozycja kątowa osi wiodącej i nadążnej jest taka sama ($\alpha = \beta$). Zależność ta może być oczywiście inna ($\alpha \neq \beta$), swobodnie konfigurowalna, ale znana. Ta cecha sprężenia jest charakterystyczna dla liniowej synchronizacji absolutnej. Dojście do zaspężenia określają parametry funkcji interfejsu użytkownika, o czym w dalszej części dokumentu.

Synchronizacja absolutna znajduje zastosowanie wszędzie tam, gdzie konieczne jest sprzężenie osi z uwzględnieniem punktu ich spotkania – mogą być to maszyny pakujące, drukujące, manipulatory czy też przenośniki. Charakterystycznym przykładem jest aplikacja tzw. latającej piły (rysunek poniżej) bądź noża obrotowego, które będą szczegółowo omówione w kolejnych rozdziałach.



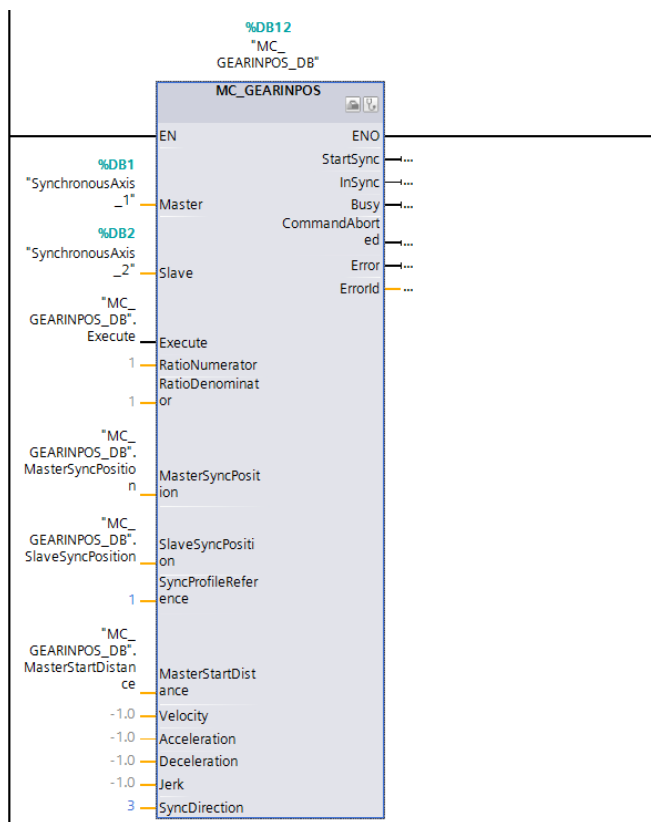
16.2 Konfiguracja w TIA Portal

Analogicznie jak w przypadku synchronizacji relatywnej osi, sprzężenie absolutne również wymaga zdefiniowania osi synchronicznej (obiekt technologiczny *TO_SynchronousAxis*), która będzie osią nadążną (choć i w tym przypadku może być osią wiodącą dla innej osi synchronicznej). Po raz wtóry definiujemy więc oś pozycjonującą, rozszerzoną o możliwość podążania ścieżką innej osi. Po zdefiniowaniu obiektu technologicznego pozostaje wskazać z listy oś (*leading value interconnections*), która – w programie użytkownika – może zostać wywołana jako wiodąca.

Podobnie jak poprzednio, osią wiodącą absolutnej pracy synchronicznej może być zależna pozycyjnie oś pozycjonująca bądź synchroniczna (część istniejącego układu mechanicznego). Może być to również oś wirtualna (wzorec sygnału wiodącego generowany przez system nadrzędny) lub enkoder zewnętrzny (dodatkowy pomiar pozycji z istniejącej już części maszyny).

Interfejs użytkownika – funkcja *MC_GearInPos*

Obsługa osi przez podstawowe funkcje interfejsu użytkownika (*MC_Power*, *MC_Halt*, *MC_Home*, *MC_Reset*) nie różni się w stosunku do synchronizmu relatywnego. Pozostaje więc jedynie kwestia bloku, który służy do aktywacji sprzężenia – *MC_GearInPos*.



Różnica nie jest duża w stosunku do funkcji *MC_GearIn*, aczkolwiek parametry, które występują jedynie w funkcji *MC_GearInPos* są bardzo istotne z punktu widzenia funkcjonalności jaką chcemy uzyskać.

Do parametrów wejściowych *Master* oraz *Slave* podłączamy (można to uczynić przez przeciągnięcie – *drag&drop* – bezpośrednio z drzewka projektu) oś wiodącą oraz nadążną (synchroniczną). Podanie zbocza narastającego na wejście *Execute* spowoduje wywołanie sprzężenia osi.

Interfejs funkcji *MC_GearInPos*, podobnie jak poprzednio, posiada opcjonalnie możliwość wprowadzenia przekładni w jakiej mają poruszać się osie sprzężone w ruchu synchronicznym – *RatioNumerator* oraz *RatioDenominator*. Czyli określamy ile obrotów ma wykonać oś nadążna na określoną ilość obrotów osi wiodącej. Przykładowo ustawiając przełożenie na wartość 1:3 – oś nadążna będzie pokonywać trzykrotność dystansu odczytanego z pozycji (zadanej lub aktualnej) osi wiodącej.

Kolejne dwa parametry (*MasterSyncPosition* oraz *SlaveSyncPosition*) są charakterystyczne dla synchronizacji absolutnej. Określają one punkt spotkania osi *master* i *slave*. Wyznaczamy więc pozycję osi wiodącej oraz nadążnej, w jakiej ma nastąpić zaspężlenie.

Pytanie, które się nasuwa w związku z wyznaczeniem powyższego punktu spotkania dla osi synchronicznych to – jak przebiegać ma dojście do sprzężenia od strony

parametrów dynamicznych osi nadążnej? Zagadnienie to realizowane jest przez wyznaczenie tzw. profilu synchronizacji, czyli właśnie określenia z jaką dynamiką lub na jakiej drodze synchronizacja ma zostać wykonana. Podstawowy parametr odpowiadający za wybór charakterystyki przejazdu osi nadążnej to *SyncProfileReference*. Mamy do wyboru dwie opcje – synchronizacja wyznaczająca profil synchronizacji na podstawie wskazanego dystansu pokonanego przez osi wiodącą lub synchronizacja oparta tylko o dynamikę obu osi. Na bazie parametrów dynamicznych i/lub przebytej drogi, obliczany jest profil, czyli trajektoria ruchu osi nadążnej w celu osiągnięcia sprężenia.

Wyjścia funkcji *MC_GearInPos* dostarczają informacji na temat statusu sprężenia (sprężony, zajęty, procedura przerwana) oraz ewentualnych błędów.

Synchronizacja „na drodze” (*SyncProfileReference* = 1)

Ustawienie parametru *SyncProfileReference* na wartość 1 oznacza wybór synchronizacji bazujący na drodze pokonanej przez osi wiodącą. W tym wariancie, dystans pokonany przez osi master (oraz jej dynamika) determinuje wyznaczenie trajektorii (profilu) synchronizacji dla osi synchronicznej.

Dystans ten wskazujemy przez podanie wartości parametru *MasterStartDistance*. Parametr ten istotny jest tylko w przypadku gdy *SyncProfileReference* = 1. Definiując zatem wartość drogi przebytej przez osi wiodącą określamy, w którym punkcie system rozpocznie realizację obliczonej trajektorii synchronizacyjnej dla osi nadążnej.

Profil synchronizacji jest stale przeliczany od momentu aktywacji funkcji do uzyskania statusu sprężenia osi. W związku z tym, wszelkie zmiany dynamiki osi zostają nałożone na obliczony profil.

Jeżeli w momencie aktywacji funkcji obie osie znajdują się w swoich pozycjach synchronicznych to status sprężenia (*MC_GearInPos.InSync*) zostanie ustawiony od razu na wartość 1. Jeśli w momencie aktywacji funkcji osi wiodąca znajduje się w swojej pozycji synchronicznej to musi ona najpierw przekroczyć punkt startowy synchronizacji (*MasterSyncPosition* minus *MasterStartDistance*), aby zainicjalizować synchronizację.

Przykładowo, przyjmijmy, że obie osie znajdują się na pozycji wyjściowej 0 mm. Uruchamiamy osi wiodącą i rozpędzamy do stałej prędkości 100mm/s. Ustawiamy wartość parametrów funkcji *MC_GearInPos* jak poniżej:

SyncProfileReference = 1

MasterSyncPosition = 100

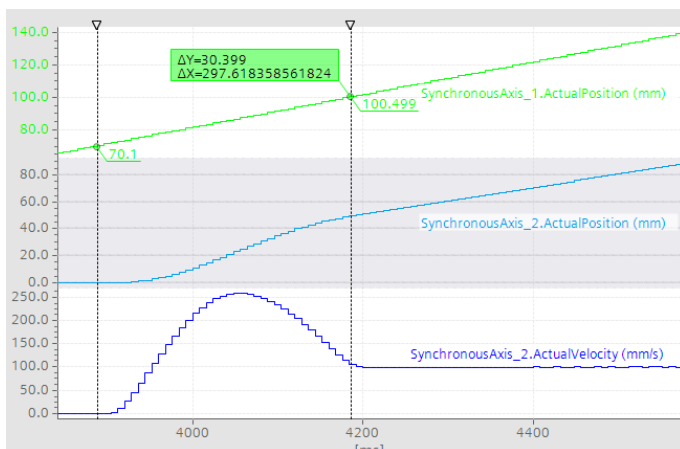
SlaveSyncPosition = 50

MasterStartDistance = 30

Zgodnie z powyższymi wartościami uruchamiamy funkcję sprężenia absolutnego, czyli wykonujemy synchronizację opartą o wartość drogi osi master. Punktem spotkania jest pozycja 100 mm (dla osi wiodącej) oraz pozycja 50 mm (dla osi nadążnej). Synchronizacja ma zostać wykonana „na drodze” 30 mm pokonanych przez osi wiodącą. W związku z powyższym, osi nadążna powinna rozpocząć synchronizację do osi wiodącej gdy ta będzie na pozycji 70 mm. W momencie zasprężenia (po 100 mm przebytych przez osi wiodącą od momentu jej uruchomienia) osi nadążna powinna znaleźć się na pozycji 50 mm (niezależnie od wartości jej pozycji wyjściowej). Jeśli dynamika układu nie pozwala na realizację tego zadania – obiekt technologiczny zgłosi błąd synchronizacji.

Poniższy przebieg zależności **pozycji osi wiodącej** (poruszającej się ze stałą prędkością 100 mm/s), **pozycji osi nadążnej** oraz **prędkości osi nadążnej** w czasie pozwala zaobserwować zmiany wartości parametrów obiektów technologicznych podczas procedury synchronizacji.

Opóźnienie w czasie, które widać na wykresie prędkości, osi nadążnej wynika z faktu, że zaprezentowany został przebieg wartości aktualnej, a nie zadanej.



Synchronizacja oparta o parametry dynamiczne (*SyncProfileReference = 0*)

Ustawienie parametru profilu synchronizacji na wartość 0 oznacza wybór synchronizacji oparty o dynamikę naszego układu. Ta opcja prowadzi do wyznaczenia trajektorii synchronizacji, która pozwoli na maksymalnie szybką synchronizację – ograniczoną przez dynamikę układu.

Wybierając ten profil sprężenia absolutnego, system jako podstawowe parametry wejściowe do obliczenia trajektorii synchronizacji osi nadążnej analizuje:

- wprowadzoną pozycję synchroniczną dla osi wiodącej oraz nadążnej,
- parametry dynamiczne instrukcji *Motion Control* (prędkość, przyspieszenie/opóźnienie oraz zryw),
- aktualną pozycję oraz dynamikę osi wiodącej, oraz nadążnej.

Profil synchronizacji jest stale przeliczany od momentu aktywacji funkcji do uzyskania statusu sprężenia osi. W związku z tym, wszelkie zmiany dynamiki osi zostają nałożone na obliczony profil.

Jeśli w momencie aktywacji funkcji, oś wiodąca znajduje się już na swojej pozycji synchronicznej – oś nadążna zostanie przesunięta na wskazaną dla niej pozycję synchronizacji. Jeżeli w momencie aktywacji funkcji obie osie znajdują się w swoich pozycjach synchronicznych – status sprężenia (*MC_GearInPos.InSync*) zostanie ustawiony od razu na wartość 1.

Przykładowo, przyjmijmy, że obie osie znajdują się na pozycji wyjściowej 0 mm. Uruchamiamy oś wiodącą i rozpędzamy do stałej prędkości 100mm/s. Ustawiamy wartość parametrów funkcji `MC_GearInPos` jak poniżej:

`SyncProfileReference = 0`

`MasterSyncPosition = 100`

`SlaveSyncPosition = 50`

`Velocity = 500`

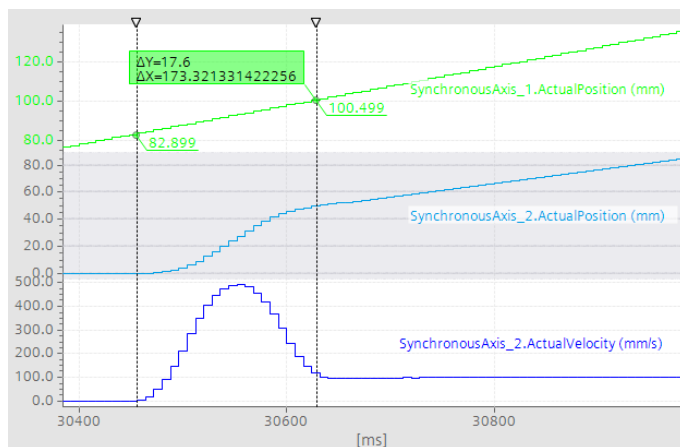
`Acceleration = 10000`

`Deceleration = 10000`

`Jerk = 200000`

Zgodnie z powyższymi wartościami, uruchamiamy funkcję sprzężenia absolutnego, czyli wykonujemy synchronizację opartą o parametry dynamiczne. Punktem spotkania, podobnie jak w poprzednim przykładzie, jest pozycja 100 mm (dla osi wiodącej) oraz pozycja 50 mm (dla osi nadążnej). Tym razem nie wiadomo na jakiej drodze osi wiodącej synchronizacja zostanie wykonana – zostanie to wyliczone przez wewnętrzny mechanizm systemu na podstawie aktualnych parametrów dynamicznych osi wiodącej oraz wskazanych parametrów synchronizacji dla osi nadążnej. W momencie zaszprzęglenia (po 100 mm przebytych przez oś wiodącą od momentu jej uruchomienia) oś nadążna powinna znaleźć się na pozycji 50 mm (niezależnie od wartości jej pozycji wyjściowej). Jeśli dynamika układu nie pozwala na realizację tego zadania – obiekt technologiczny zgłosi błąd synchronizacji.

Poniższy przebieg zależności **pozycji osi wiodącej** (poruszającej się ze stałą prędkością 100mm/s), **pozycji osi nadążnej** oraz **prędkości osi nadążnej** w czasie, pozwala zaobserwować zmiany wartości parametrów obiektów technologicznych podczas procedury synchronizacji.



W porównaniu do poprzedniego wyniku, możemy zaobserwować, że dynamika układu pozwoliła wykonać synchronizację szybciej – na drodze 17.6 mm przebytych przez oś wiodącą. Dobór odpowiedniej metody synchronizacji zależy jest wyłącznie od wymogów aplikacyjnych.

Odsprężenie osi nadążnej

Status sprężenia osi synchronicznej (*MC_GearInPos.InSync* oraz *MC_GearInPos.Busy*) pozostaje aktywny do momentu zakończenia pracy synchronicznej.

Odsprężenie osi nadążnej następuje w momencie wywołania dowolnej innej funkcji *Motion Control* na jej obiekcie technologicznym (*TO_SynchronousAxis*), np. przez zatrzymanie, zadanie prędkości bądź pozycjonowanie. Uwaga: w przeciwieństwie do synchronizacji relatywnej, nie ma możliwości odsprężenia osi nadążnej przez jej bazowanie. Błąd osi nadążnej również spowoduje zakończenie pracy synchronicznej.

17 Przesunięcie w ruchu synchronicznym

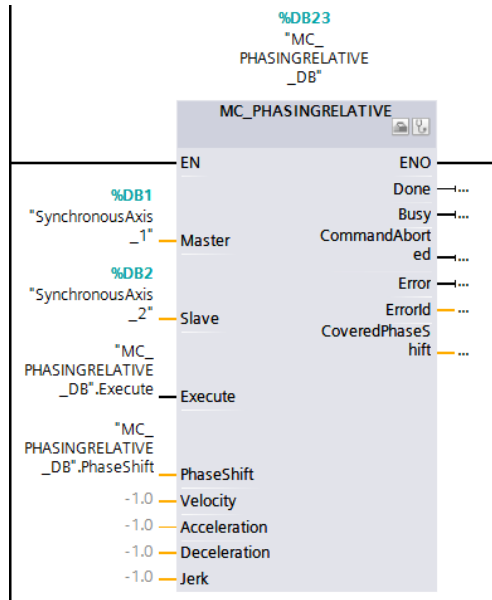
17.1 Przesunięcie fazowe

Obiekty technologiczne będące w sprzężeniu (relatywnym lub absolutnym), pozostają w stałej zależności pozycji aktualnej osi nadążnej względem osi wiodącej. Liniowa zależność pozycji obu osi może zostać przesunięta – bez konieczności dezaktywacji sprzężenia. Dla przykładu przyjmijmy, że dwie osie o przełożeniu 1:1 poruszają się synchronicznie – wartość ich pozycji jest identyczna. Różnica pomiędzy pozycją osi pierwszej i drugiej wynosi zero. Jeśli chcemy, aby oś nadążna wyprzedziła (lub przesunęła się wstecz) oś wiodącą, a co za tym idzie, aby została wygenerowana konkretna różnica pomiędzy ich pozycjami aktualnymi – możemy wykorzystać funkcję *MC_PhasingRelative* lub *MC_PhasingAbsolute*.

Przesunięciem fazowym nazywamy więc zmianę wartości pozycji zadanej osi nadążnej względem osi wiodącej w ruchu synchronicznym.

Parametrami funkcji realizujących te zadania są przede wszystkim obiekty technologiczne, czyli obiekty osi *master* oraz *slave*. Dodatkowo podajemy zbocze narastające na wejście *Execute* oraz wskazujemy wartość przesunięcia na wejście funkcji (*PhaseShift*). Parametry dynamiki przejazdu możemy ustalić ręcznie lub pozostawić jako wartości domyślne, które zostaną pobrane z ustawień obiektu technologicznego.

Wartość prędkości zadanej w parametrze funkcji zostaje zsumowana z prędkością aktualną osi nadążnej. Przykładowo dla osi poruszającej się z prędkością 20mm/s, wywołanie funkcji przesunięcia fazowego z parametrem *Velocity* = 100mm/s, spowoduje wykonanie zadania z prędkością maksymalną 120mm/s.



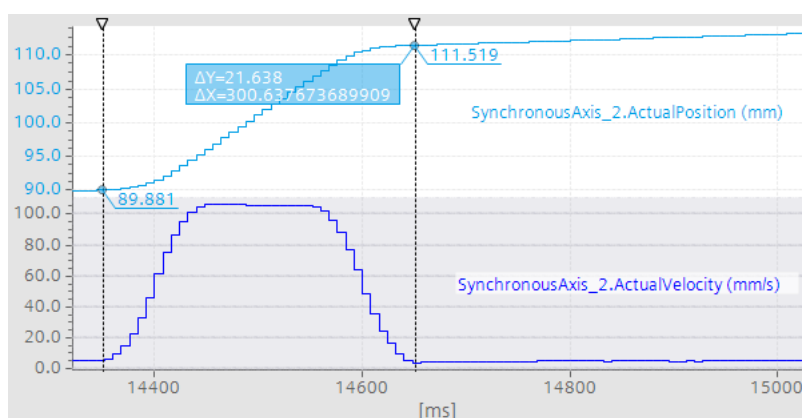
Pozostałe parametry dynamiczne (przyspieszenie, opóźnienie oraz zryw) nie sumują się z wartościami aktualnymi (nie jest to warunek konieczny do wykonania przesunięcia) – zostają przyjęte wartości domyślne z obiektu technologicznego lub te podane jako parametry funkcji w programie użytkownika.

Wartość pozycji osi wiodącej nie zostaje naruszona przez funkcje przesunięcia fazowego – przesunięcie realizowane jest jedynie przez zmianę parametrów dynamicznych osi nadążnej.

Warunki poprawnego wywołania funkcji dotyczą przede wszystkim poprawnej konfiguracji osi synchronicznej. Więcej informacji odszukać można w plikach pomocy systemowej TIA Portal dla w/w funkcji.

Przesunięcie fazowe relatywne

Pierwsza funkcja (*MC_PhasingRelative*) pozwoli przesunąć oś nadążną względem wiodącej relatywnie, czyli wskazujemy konkretną wartość, o jaką chcemy zwiększyć (lub zmniejszyć) odległość osi nadążnej względem wiodącej. Poniższy przebieg wartości **prędkości** oraz **pozycji** osi nadążnej w czasie prezentuje zachowanie obiektu technologicznego po wykonaniu funkcji.



W powyższym przykładzie oś nadążna została przesunięta podczas ruchu synchronicznego o 20 jednostek zgodnie z kierunkiem przyrostu pozycji. W związku z tym, że oś wiodąca była w ciągłym ruchu (5 mm/s) – przebieg pozycji przedstawia przyrost większy niż 20, gdyż systemowo został również dodany dystans, który oś wiodąca pokonała podczas wykonywania zadania przesunięcia fazowego osi nadążnej. Po wykonaniu funkcji, różnica pomiędzy pozycją osi *master*, a *slave* wzrosła o dokładnie 20 jednostek.

Jeśli osie poruszają się w synchronizmie z przekładnią elektroniczną, zostanie ona również uwzględniona podczas przesunięcia.

Przesunięcie fazowe absolutne

Wykorzystując funkcję *MC_PhasingAbsolute* nie podajemy już odległości, o jaką chcemy przesunąć oś nadążną względem wiodącej – w tym przypadku podajemy konkretną pozycję na jaką chcemy przesunąć oś nadążną. Punktem odniesienia wartości absolutnej przesunięcia jest tutaj różnica pozycji osi wiodącej oraz nadążnej w momencie dojścia do sprzężenia (np. przez wywołanie omówionych już funkcji *MC_GearIn* lub *MC_GearInPos*). Przesunięcie osi nadążnej następuje z uwzględnieniem zdefiniowanej przekładni (jeśli jest różna od 1:1). Czyli wykonując przesunięcie na pozycję 10 (w odniesieniu do punktu zerowego) w układzie sprzężonym w relacji przykładowo 1:2 – oś nadążna fizycznie przemieszczona zostanie o 20 jednostek.

Wartość absolutnej pozycji odniesienia pamiętana jest w systemie do momentu wyjścia ze sprzężenia, czyli wyzerowania statusu *MC_GearIn.InGear* lub *MC_GearInPos.InSync*.

17.2 Przesunięcie relatywne nałożone

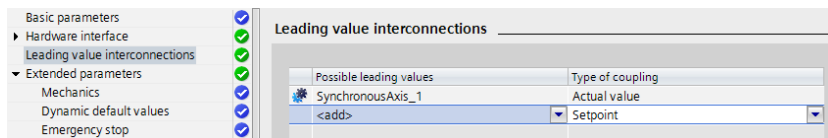
W zakresie przesunięcia, jakie możemy wygenerować na osi będącej w ruchu (również synchronicznym), mamy do dyspozycji jeszcze tzw. przesunięcie relatywne nałożone, czyli zadanie realizowane przez systemową funkcję *MC_MoveSuperimposed*.

Funkcja działa bardzo podobnie do omówionej w poprzedniej sekcji, *MC_PhasingRelative*. Mianowicie pozwala ona na relatywne przesunięcie, a konkretnie nałożenie parametrów na oś będącą w ruchu. Nałożenie oznacza, że wartość dodatkowego przesunięcia oraz prędkości, z jaką zostanie ono wykonane, zostają dodane do aktualnych wartości odpowiednich parametrów. Przyspieszenie, opóźnienie oraz zryw nie sumują się – przyjęte zostają wartości domyślne z obiektu technologicznego lub zadane przez parametry funkcji.

Różnica w stosunku to przesunięcia fazowego polega przede wszystkim na tym, że wywołujemy przesunięcie na dowolnej osi – niezależnej lub synchronicznej (wiodącej bądź nadążnej). Działanie funkcji *MC_MoveSuperimposed* jest niezależne od parametrów synchronizacji. Oznacza to, że przesunięcie można zaaplikować na oś wiodącą (wtedy automatycznie oś nadążna będzie „gonić” oś wiodącą) lub nadążną (w tym wypadku ruch osi wiodącej nie zostanie naruszony) – tym razem bez uwzględnienia zdefiniowanej przekładni, ale podobnie jak poprzednio – bez utraty statusu aktywnego sprzężenia osi *master/slave* (zarówno przy synchronizacji relatywnej jak i absolutnej).

18 Sprężenie do wartości aktualnej oraz zadanej

Podczas konfiguracji obiektu technologicznego typu *TO_SynchronousAxis* definiowaliśmy listę możliwych osi wiodących (parametr *Leading value interconnections*). Trzeba zwrócić tutaj jeszcze uwagę na wybór typu sprzężenia z daną osią wiodącą (*Type of coupling*). Do wyboru mamy wartość aktualną (*Actual value*) lub zadaną (*Setpoint*) pozycji osi wiodącej.



Wracając do początku niniejszej dokumentacji, warto przypomnieć sobie działanie mechanizmu pozycjonującego sterownika S7-1500. Wewnętrzny mechanizm oparty jest o interpolator i regulator pozycji (sterownik) oraz regulator prędkości (napęd). Podczas interpolacji, system wyznacza wartość zadaną pozycji na kolejny cykl pracy systemu. W sprzężeniu zwrotnym otrzymujemy wartość pozycji aktualnej. Kalkulacja wykonana przez mechanizm interpolacji na każdy cykl pozycjonowania wyznacza wartość zadaną pozycji.

W zależności od układu mechanicznego zostanie ona osiągnięta w odpowiednim czasie lub wcale (*following error*), natomiast zawsze będzie występować rozbieżność pozycji aktualnej w stosunku do pozycji zadanej. Tutaj właśnie pojawia się nasze zagadnienie – czy synchronizacja osi nadążnej powinna być wykonana do obliczonej przez interpolator pozycji zadanej (teoretycznej), czy też do aktualnej (fizycznie osiągniętej) pozycji osi wiodącej odczytanej z enkodera?

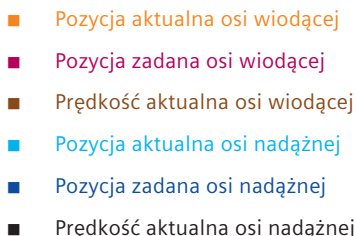
Synchronizacja do wartości zadanej

Wybór odpowiedniego typu sprzężenia synchronicznego zależy tylko i wyłącznie od wymagań realizowanego projektu. Rozważmy zatem aspekty czysto teoretyczne. Wartość zadana zostaje wyznaczona przez interpolator na każdy krok układu pozycjonującego. Jest to zatem wartość generowana przez sterownik i nie ma nic wspólnego z aktualną wartością, którą cały układ osiągnął w danym cyklu. Pytanie zatem, czy nasza oś nadążna powinna podążać zawsze trajektorią wyznaczoną dla osi wiodącej, czy też synchronizować się z pozycją aktualną tej osi. Ma to diametralne znaczenie z punktu widzenia aplikacji. Wybierając synchronizację do wartości zadanej – niezależnie od tego co się będzie działo z osią wiodącą – oś nadążna będzie pozycjonowana zgodnie z wartością wyznaczoną przez interpolator dla osi *master*.

W związku z powyższym – rozpatrzmy prosty przykład.

Dwie osie poruszają się w ruchu synchronicznym, gdzie oś nadążna synchronizuje się do wartości zadanej osi wiodącej. W pewnym momencie następuje chwilowe mechaniczne zatrzymanie osi wiodącej.

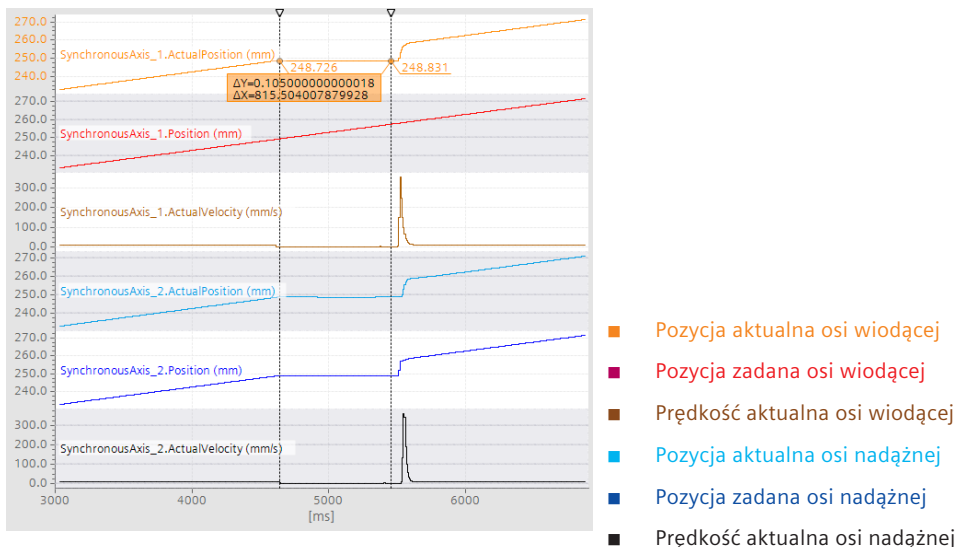
Poniższy przebieg parametrów dynamicznych osi nadążnej oraz wiodącej prezentuje zachowanie systemu w takiej sytuacji.



Podobne zjawisko obserwujemy dla prędkości osi wiodącej, która po ustąpieniu blokady mechanicznej, automatycznie gwałtownie przyspiesza, aby nadążyć za pozycją obliczoną przez interpolator – nie wpływa to jednak w żaden sposób na zachowanie osi nadążnej.

Drugi przypadek to synchronizacja osi nadążnej do wartości aktualnej osi wiodącej. W tym wariancie, wartością zadaną dla osi *slave* będzie wartość pozycji zmierzona przez układ pomiarowy osi wiodącej (enkoder). W związku z tym, że system analizować będzie realną (fizyczną) pozycję osi wiodącej – każda jej zmiana (również wynikająca z pracy układu, mechanicznego zatrzymania, zakłóceń, itp.) zostanie przeniesiona bezpośrednio na ruch osi nadążnej.

Poniższy przebieg parametrów dynamicznych osi nadążnej oraz wiodącej prezentuje zachowanie systemu w takiej sytuacji.

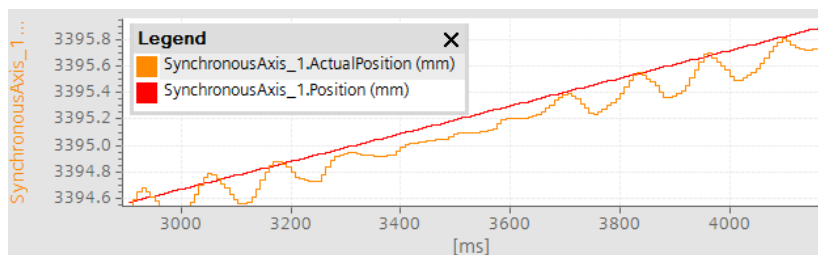


Ponownie obserwujemy, że przyrost pozycji aktualnej osi wiodącej zostaje zatrzymany, natomiast pozycja zadana kalkulowana przez interpolator stale obliczana jest zgodnie z zadaną dynamiką. W przeciwieństwie do synchronizacji z wartością zadaną – oś nadążna w tym układzie, zostaje wstrzymana zgodnie z pozycją aktualną osi wiodącej. Jest to bezpośredni efekt synchronizacji do wartości aktualnej pozycji osi wiodącej.

Podobne zjawisko obserwujemy dla prędkości osi wiodącej, która po ustąpieniu blokady mechanicznej automatycznie gwałtownie przyspiesza, aby nadążyć za pozycją obliczoną przez interpolator – ma to bezpośrednie przeniesienie na dynamikę osi nadążnej.

Taka metoda synchronizacji może okazać się przydatna w układach, gdzie każda zmiana pozycji osi wiodącej (również wynikająca z czynników zewnętrznych) powinna być automatycznie przeniesiona na oś synchroniczną.

Nie jest to jednak metoda zalecana, ze względu na jakość sygnału – wartość aktualna pozycji odczytana z układu pomiarowego osi wiodącej nie będzie tak dokładna jak wzorcowa obliczona przez interpolator. Wszelkiego rodzaju drgania oraz zakłócenia osi wiodącej, również zostaną przeniesione na oś nadążną. Porównanie wartości **zadanej** oraz **aktualnej** osi wiodącej (zakłócającej przez czynniki zewnętrzne – np. mechaniczne drgania) można zaobserwować na przykładowym przebiegu tych parametrów:



Kolejną wadą jest opóźnienie spowodowane koniecznością odczytu wartości pozycji osi *master*, jej przetworzeniem przez system nadrzędny i przesłaniem jako wartość zadaną dla osi *slave*, a także możliwościami dynamicznymi osi nadążnej. Na przykładowym wykresie można z łatwością zaobserwować kilkanaście milisekund opóźnienia pozycjonowania (oraz zmian prędkości) osi nadążnej w stosunku do pozycji aktualnej osi wiodącej.

W celu kompensacji tego efektu można wykonać tzw. ekstrapolację wartości aktualnej. Konfigurację filtrowania wykonujemy w ustawieniach obiektu technologicznego osi wiodącej (*Extended parameters* → *Actual value extrapolation*). Więcej informacji znajduje się w plikach pomocy systemu TIA Portal pod hasłem *Extrapolation of the leading values for actual value coupling*.

Warto również nadmienić, że w przypadku, gdy osią wiodącą jest obiekt technologiczny typu enkoder zewnętrzny (*TO_ExternalEncoder*) naturalnie możliwa jest tylko synchronizacja do wartości aktualnej.

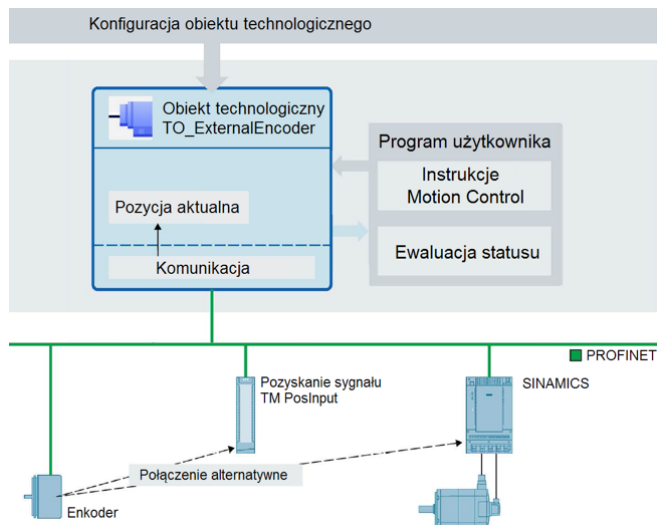
Obiekt technologiczny – enkoder zewnętrzny

Wspomniany już w poprzednich rozdziałach enkoder zewnętrzny jest istotnym elementem przy zadaniach koordynacji ruchu. Detekcja sygnału przez dodatkowy czujnik pozwoli odczytać aktualną pozycję układu w dowolnym punkcie pomiarowym. Zewnętrzne urządzenie pomiarowe może zostać podłączone do systemu sterowania na kilka sposobów. Możemy podłączyć enkoder bezpośrednio do napędu (jeśli ten posiada takie złącze) lub do sterownika.

W tym drugim przypadku, dla niewymagających aplikacji, mogą być to standardowe wejścia cyfrowe lub szybkie wejścia licznikowe. W bardziej zaawansowanych konfiguracjach, może być to połączenie przez dedykowany moduł technologiczny (wspomniane w rozdziale *Układ pomiarowy* *TM_Count* lub *TM_PosInput*) ewentualnie przez sieć komunikacyjną (np. PROFINET).

Poza fizycznym połączeniem detektora pozycji aktualnej, pozostaje kwestia jak zostanie on zaprogramowany w środowisku inżynierskim. Tutaj również można wybrać jedną z kilku dostępnych metod. W aplikacjach prostych można ewaluować stan wejścia cyfrowego przez program użytkownika lub przestrzeń adresową modułu technologicznego. Jeśli jednak przyjdzie nam oprogramować bardziej zaawansowane urządzenie, o krótkim cyklu pracy – zdecydowanie wygodniej będzie zastosować mechanizm systemowy, czyli dobrze nam już znany obiekt technologiczny. W zależności od konfiguracji sprzętowej będziemy mieli do wyboru obiekt technologiczny typu szybki licznik (*High Speed Counter*) lub *ExternalEncoder*. Ten ostatni jest najwygodniejszą oraz najbardziej funkcjonalną opcją, co więcej sterownik z serii S7-1500T pozwoli skonfigurować ten obiekt technologiczny jako oś wiodącą w ruchu synchronicznym. Procedura uruchomienia jest więc bardzo przystępna i analogiczna do wcześniej już poznanych obiektów technologicznych osi napędowych.

Poniższa ilustracja przedstawia zasadę działania obiektu technologicznego *TO_ExternalEncoder*



Budowa obiektu technologicznego pozwoli określić typ osi (obrotowa/liniowa), jej mechanikę (przekładnia, moduło), jednostkę pomiarową, parametry bazowania, filtrowanie sygnału (ekstrapolacja) oraz pozostałe atrybuty podłączonego układu pomiarowego. Kompletna konfiguracja pozwoli zastosować odczytaną pozycję, analogicznie jak w przypadku standardowej osi pozycjonującej bądź synchronicznej.

Oś wirtualna oraz symulowana

Dopełniając opis zagadnień synchronizacji układów napędowych, należy również wspomnieć, iż sterowniki serii S7-1500 umożliwiają sterowanie osią technologiczną w trybie wirtualnym lub symulacji. Oznacza to, że poza fizyczną osią (pozycjonującą, synchroniczną lub enkoderem zewnętrznym) – źródłem sygnału wiodącego dla osi nadążnej może być także programowy generator pozycji zadanej.

Docelowo uzyskamy podobny rezultat, jak w uprzednio opisanej synchronizacji do wartości zadanej osi fizycznej. Jest jednak kilka różnic w samej charakterystyce danego typu osi.

Oś rzeczywista

Standardowa konfiguracja sprzętowa osi rzeczywistej, posiada możliwość sterowania ruchem i wymaga podłączenia fizycznego zestawu urządzeń (napęd, silnik, enkoder).

Oś wirtualna

W ustawieniach obiektu technologicznego osi możemy zaznaczyć opcję ☐ Virtual axis. Standardowa oś napędowa zostaje przekonwertowana na oś wirtualną. Następstwem jest fakt, że oś jest niezmiennie kontrolerem ruchu, natomiast nie wymaga podłączenia napędu oraz enkodera. Oznacza to, że nie steruje rzeczywistym napędem.

Oś wirtualna jest często stosowana jako oś pomocnicza, która generuje wartości zadane pozycji dla wielu osi nadążnych pracujących synchronicznie. Nie wytwarza ona błędów, odchyień ani nie ulega zakłóceniom bądź awariom, jest zatem doskonałym wzorcem sygnału dla napędów skoordynowanych.

Zmiana konfiguracji osi wirtualnej może zostać zaimplementowana, jedynie przez wgranie przekonfigurowanego obiektu technologicznego do sterownika (w trybie STOP).

Symulacja osi

W podobny sposób, możemy w karcie właściwości obiektu technologicznego aktywować znacznik ☐ **Activate simulation**. Uruchomiona w ten sposób symulacja osi, jest również kontrolerem ruchu, w tym wypadku jednak w początkowej fazie nie wymaga podłączenia fizycznego napędu oraz enkodera.

Symulację stosuje się np. w celu oprogramowania oraz testowania układu bez fizycznych połączeń z napędami. Oś symulowana w końcowej fazie uruchomienia zostaje podłączona do rzeczywistych urządzeń.

Zmiana konfiguracji osi symulowanej może być zmieniona podczas pracy systemu – bez konieczności przeładowania projektu w sterowniku. Jednakże w momencie zakończenia trybu symulacji – wymagane jest poprawne połączenie napędu oraz enkodera.

Jeżeli aktywna jest zarówno symulacja jak i wirtualizacja osi – konfiguracja symulacji nie może zostać zmieniona w trybie pracy systemu.

Charakterystyka osi symulowanej/wirtualnej

Oś, z aktywną jedną z powyższych opcji, nie wysyła wartości zadanych do napędu oraz nie odczytuje wartości aktualnych. Sprzętowe krańcówki (*Limit switch*) oraz znaczniki bazowania nie mają tutaj zastosowania.

Obiekty technologiczne uwzględniające detekcję sygnału, np. przez moduł *TM_Timer_DIDQ* (*TO_MeasuringInput*, *TO_OutputCam*, *TO_CamTrack*) mogą zostać wykorzystane w połączeniu z osią wirtualną bądź symulowaną. Mechanizmy szybkiej detekcji oraz reakcji przez sygnały binarne omówimy w kolejnych rozdziałach.

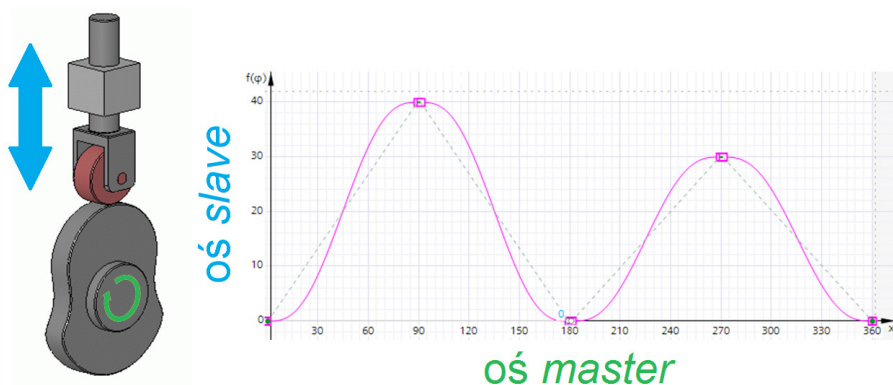
V. Sprzężenie krzywkowe

Niniejsza sekcja poświęcona została synchronizacji osi w wieloosiowym układzie napędowym. W odróżnieniu do omówionych uprzednio zagadnień, tym razem zajmiemy się synchronizmem nieliniowym. Synchronizacja osi przez swobodnie zdefiniowany profil nazywamy sprzężeniem krzywkowym.

19 Mechanizm krzywkowy

Znany już w XV wieku mechanizm krzywek jest układem mechanicznym pozwalającym na zamianę ruchu obrotowego (*oś master*) na nieliniowy ruch posuwisto-zwrotny (*oś slave*).

Krzywka jest elementem mimośrodowym konstrukcji mechanicznej o specjalistycznym kształcie przenoszącym ruch wału na tzw. popychacz.



Krzywka mechaniczna

Przez wiele dekad mechanizm krzywkowy był stricte układem mechanicznym dającym szerokie możliwości przekazania pozycji jednego elementu na inny. Zmiana położenia może być jednorazowa lub okresowa. Przeznaczenie takich rozwiązań jest bardzo szerokie, sztandarowym przykładem zastosowania układu krzywkowego jest mechanizm rozrządu w silnikach spalinowych, stosowany do synchronicznego sterowania pracą zaworów.

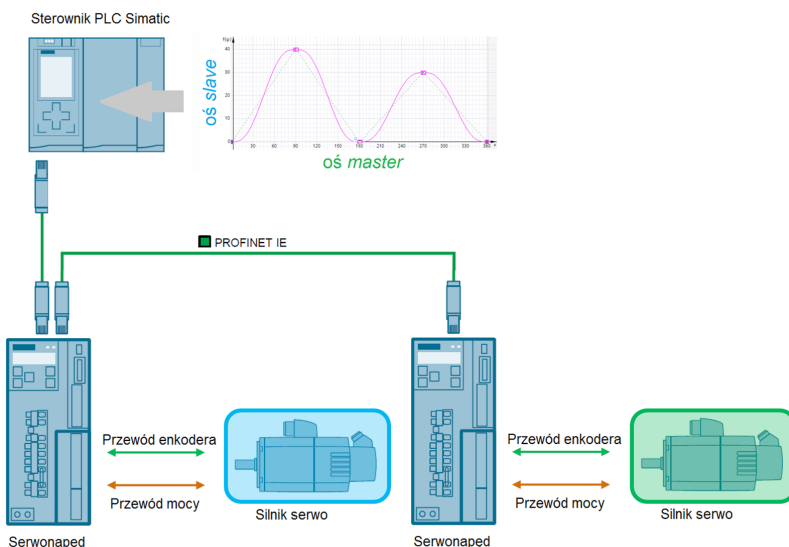
Z punktu widzenia projektu układu krzywkowego – zdefiniowanie wzajemnych przemieszczeń poszczególnych komponentów urządzenia to etap wstępny. Po wykonaniu konstrukcji mechanicznej należy wziąć pod uwagę obciążenie oraz eksploatację elementów wykonawczych, dopuszczalne przyspieszenia i potencjalne źródła błędów. Inżynier musi więc obliczyć precyzyjnie ścieżkę ruchu dla każdego z podzespołów aby zapewnić ciągłość pracy oraz zminimalizować zużycie mechanizmów. W wielu przypadkach poprawność wykonania projektu, doboru parametrów systemu czy optymalizacji układu napędowego może zostać stwierdzona dopiero po wykonaniu szeregu testów rozruchowych (np. o różnej częstotliwości), a także wytrzymałościowych na przestrzeni czasu.

Jeszcze do niedawna, sprężenia krzywkowe były intensywnie rozwijaną gałęzią inżynierii mechanicznej. W ostatnich latach tendencja ta uległa jednak diametralnej zmianie. Stało się tak przez bardzo dynamiczny rozwój technologii krzywek elektronicznych.

Krzywka elektroniczna

Wykorzystanie nadrzędnego układu napędowego oraz mechaniczne przeniesienie ruchu jednostki głównej na krzywkowy aktuator wtórny, pozwala na skoordynowany ruch synchroniczny o określonym profilu (funkcji). Rozwiązanie ściśle mechaniczne ma jednak sporo wad – dostarczalnych zarówno na etapie konstrukcyjnym jak i przy pracy oraz konserwacji maszyny. Jednorazowe zastosowanie projektu mechanicznego, zużycie elementów przeniesienia napędu czy kosztowne naprawy oraz rekonfiguracja – to tylko niektóre z utrudnień.

W związku z powyższymi niedogodnościami – sprzężenie mechaniczne zostało zastąpione krzywką elektroniczną. Zamiast pojedynczego mechanizmu napędowego, zastosowano dwa (lub więcej) niezależne napędy, które zostały zsynchronizowane odpowiednim profilem przez funkcję programową nadrzędnego systemu sterowania. Inaczej mówiąc – użyty został nadrzędny sterownik oraz napędy serwo pracujące w układzie synchronicznym. Jest to dokładnie taki sam sposób sprzężenia elektronicznego osi jak poznaliśmy już w poprzednim rozdziale, z tym, że funkcja sprzęgająca nie jest relacją liniową, a dowolną funkcją zależności osi *slave* (nadążnej) względem osi *master* (wiodącej).



Zastąpienie sprzężeń mechanicznych krzywkami elektronicznymi, znacznie poprawiło elastyczność rozwiązań, zwiększyło wydajność pracy maszyn oraz dało możliwość wprowadzania zmian do projektu na dowolnym jego etapie – bez konieczności przebudowy urządzenia. Skomplikowane mechanizmy mogły zostać zastąpione jednym lub wieloma silnikami, które wykonywały odpowiedni ruch skoordynowany według zdefiniowanej funkcji. Kluczowa z punktu widzenia operatora okazała się możliwość zmiany produktu lub jego wariantu bez zatrzymania, a nawet spowolnienia produkcji dzięki możliwości swobodnej parametryzacji krzywek w trybie pracy systemu.

Cechy te dają nieograniczone możliwości projektowe, prowadzą również bezpośrednio do redukcji kosztów oraz znacząco ułatwiają programowanie oraz usprawniają serwisowanie maszyny.

20

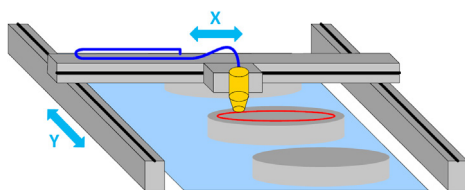
Krzywki w SIMATIC S7-1500T

Zgodnie z opisem z rozdziału poprzedniego – sterownik technologiczny SIMATIC S7-1500T pozwala na programową realizację trzech zasadniczych typów relacji synchronicznych: opisanych już wcześniej sprzężeń liniowych (relatywnego oraz absolutnego), a także krzywkowego. W zakresie rodziny sterowników S7-1500 – tylko jednostka technologiczna daje możliwość realizacji tego ostatniego zadania. Poza koniecznością posiadania jednostki z oznaczeniem „T” nie ma tutaj żadnych innych wymagań. Wystarczy środowisko inżynierskie TIA PORTAL w wersji V14 lub nowszej.

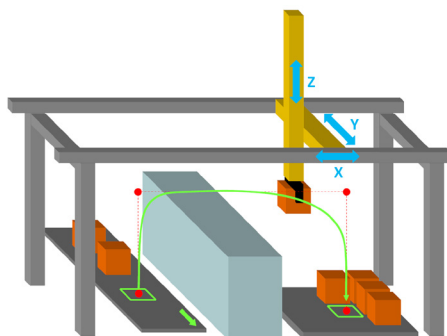
Nie są wymagane żadne dodatkowe pakiety programowe ani komponenty sprzętowe – komunikacja napęd-sterownik realizowana jest przez zintegrowany port PROFINET z obsługą komunikacji w trybie izochronicznym. Część logiczna również „zaszyta” jest w jednostce centralnej.

20.1 Zastosowanie krzywki elektronicznej

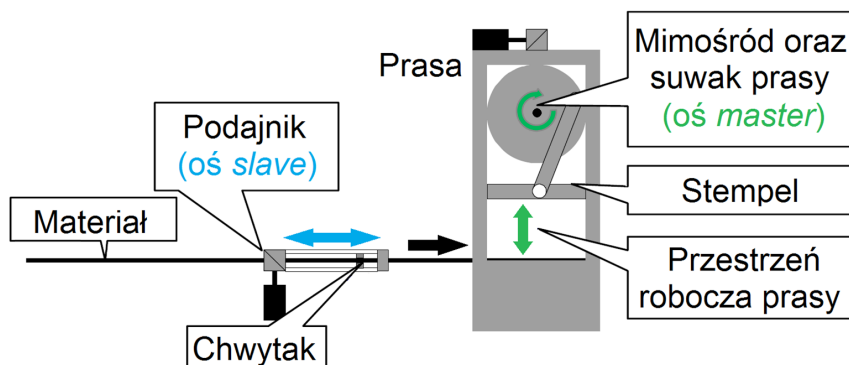
Mechanizm krzywkowy ma bardzo szerokie zastosowanie w różnego rodzaju aplikacjach. Wszędzie gdzie wymagana jest nieliniowa relacja synchroniczna dwóch lub większej ilości osi możemy zastosować sprzężenie krzywkowe. Przykładem może być – wspomniana już – aplikacja piły latającej lub noża obrotowego, które zrealizować można zarówno przez prostą synchronizację osi, ale również przez stworzenie odpowiedniej funkcji dla sprzężenia krzywkowego. Charakterystycznym przykładem mogą być również prasy lub urządzenia pakujące typu *flowpack*. Bardziej zaawansowane aplikacje, które również można zrealizować przez odpowiednią konfigurację relacji krzywkowych to urządzenia do kreślenia lub wycinania kształtów (np. frezarka czy wycinarka laserowa lub wodna) w układzie kartezjańskim XY



lub prosta interpolacja ścieżki w układzie kartezjańskim XYZ.



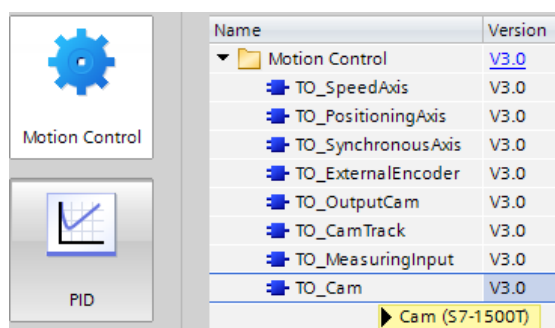
Wróćmy do przykładu z początku rozdziału, gdzie ruch obrotowy zostaje przemieniony w posuwisto-zwrotny. Klasyczne zastosowanie takiej konwersji można zaobserwować w prasach. Aplikacja prasy mimośrodowej (rysunek) to doskonała ilustracja takiego mechanizmu. Ruch obrotowy mimośrodowo przekłada się na przesunięcie stempla prasy, który dociskany jest do formy w momencie, gdy podajnik podsunie materiał w przestrzeń roboczą prasy. Mimośród stanowi tutaj klasyczne, mechaniczne przeniesienie ruchu obrotowego na liniowy, natomiast napęd mimośrodowy oraz napęd podajnika to dwie niezależne jednostki, które synchronizowane są przez profil krzywkowy zdefiniowany w nadrzędnym sterowniku PLC.



W takim urządzeniu, na jeden obrót napędu prasy (oś master) należy wygenerować przesunięcie podajnika od pozycji wyjściowej do pozycji końcowej (w przestrzeni roboczej prasy), a następnie jego powrót do pozycji bazowej. Cała maszyna pracuje zgodnie z cyklem pracy napędu master. W dalszej części dokumentu dowiemy się jak skonfigurować krzywkę do takiej aplikacji.

Obiekt technologiczny – sprzężenie krzywkowe

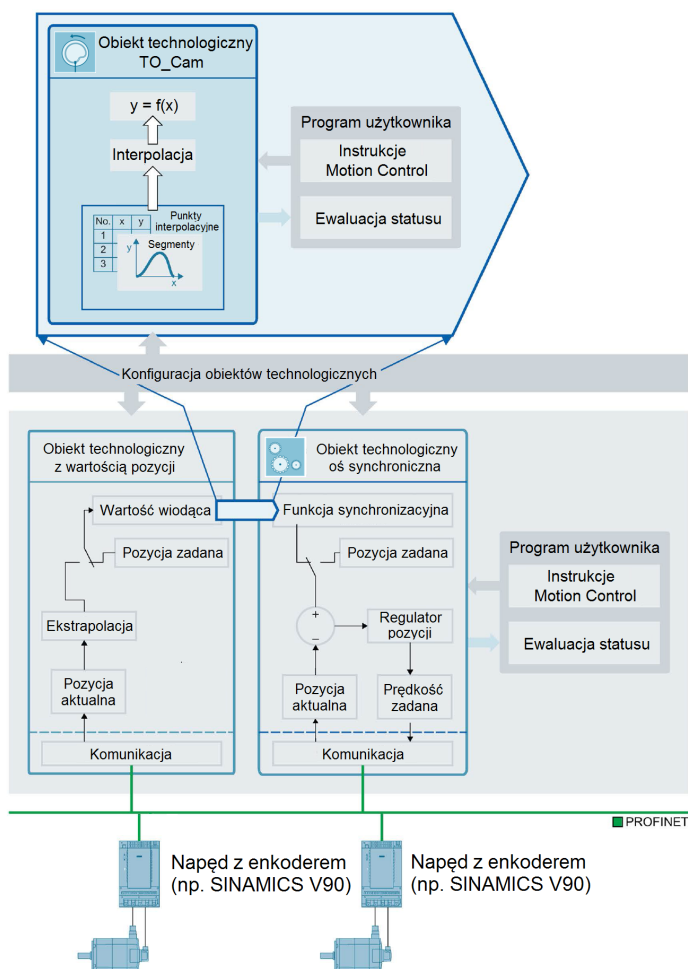
Podobnie jak w poprzednich przypadkach, tutaj również – pierwszym krokiem jest dodanie do projektu obiektu technologicznego. Nowy obiekt, który musimy wykorzystać to profil CAM, który de facto służy do sprzężenia dwóch uprzednio utworzonych obiektów technologicznych – osi master oraz osi slave. Oś wiodąca może być tutaj osią pozycjonującą, synchroniczną lub enkoderem zewnętrznym. Oś nadążna to – podobnie jak w przypadku każdego innego sprzężenia – oś synchroniczna.



Obiekt technologiczny CAM (krzywka) definiuje funkcję transferową $f(x)$. Zależność wartości wyjściowej względem wejściowej opisana jest przez funkcję o neutralnej jednostce. Funkcję przejścia (profil krzywki) definiujemy przez punkty i/lub segmenty interpolacyjne w dedykowanym edytorze graficznym. Zdefiniowana zależność musi zostać interpolowana w programie użytkownika przez funkcję *Motion Control*. Ustawienia mogą zostać zmodyfikowane w trybie pracy systemu. Te oraz inne kroki konfiguracyjne zostaną szczegółowo opisane w dalszej części dokumentu.

Poniższy schemat przedstawia działanie mechanizmu synchronizacji osi (obiektów technologicznych) w sprzężeniu krzywkowym realizowanym przez nadrzędny sterownik S7-1500T.

Możemy więc powiedzieć, że w pierwszej kolejności należy spełnić wszystkie wymagania dla pracy synchronicznej dwóch osi, a następnie zdefiniować nieliniową relację pozycji pomiędzy tymi obiektami technologicznymi.

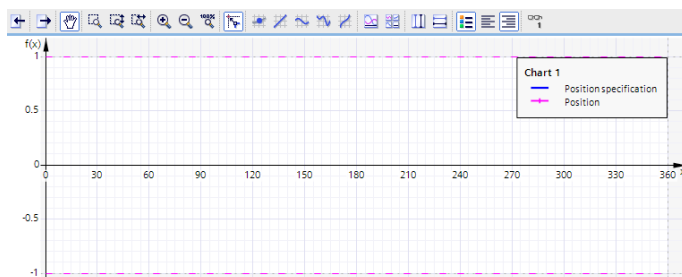


20.2 Edytor krzywek

Edytor zależności pozycji osi nadążnej względem osi wiodącej, pozwala na graficzną oraz tabelaryczną konfigurację punktów kluczowych, przez które ma przebiegać wykres funkcji, a także daje możliwość zaawansowanego określenia interpolacji jaka pomiędzy tymi punktami (lub segmentami) ma zostać przez system zastosowana. Użytkownik ma również możliwość wskazania konkretnej funkcji łączącej dane punkty profilu.

Dzięki elastycznej obsłudze interfejsu użytkownika, swobodzie wprowadzania charakterystyki wejściowej, natychmiastowej analizie parametrów dynamicznych układu w środowisku inżynierskim – zyskujemy wyższą jakość ruchu, a także pełną elastyczność w konfiguracji naszego systemu.

Po otwarciu konfiguratora widzimy pustą przestrzeń roboczą jak na poniższym obrazku.

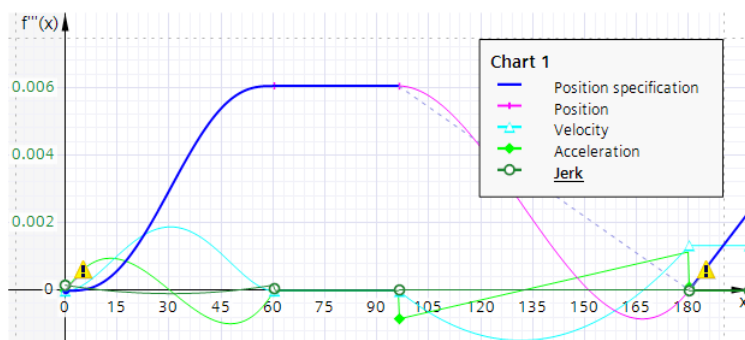


Podstawowymi parametrami, od których należy rozpocząć konfigurację jest zakres wartości jakie w obszarze sprężenia krzywkowego przyjmować będzie oś wiodąca oraz nadążna. Nasza przykładowa aplikacja składa się z wiodącej osi obrotowej modułu (0 – 360°) oraz osi liniowej. Takie też ustawienia możemy narzucić dla obszaru rysowania (zakładka *Profile* → *General*). Wartości pozycji interpretowane są zgodnie z konfiguracją obiektów technologicznych. Należy upewnić się, że zdefiniowane wartości osi zgadzają się z zakresem ich pracy w układzie mechanicznym aby zdefiniowana funkcja nie spowodowała problemów przy uruchomieniu.

Profil pozycji oraz jego pochodne

W celu zdefiniowania przebiegu funkcji pozycji możemy korzystać z punktu, linii, sinusoidy, wielomianu oraz odwróconej sinusoidy.

Po wykreśleniu profilu krzywki możemy wybrać opcję wyświetlenia pochodnych funkcji pozycji – na jednym lub czterech osobnych trendach. Z matematyki wiemy, iż pochodną funkcji jest zależność prędkości jej zmian względem jej argumentów. Argumentem naszej funkcji pozycji osi nadążnej jest pozycja osi wiodącej. W związku z powyższym pierwszą pochodną profilu pozycji $f(x)$ jest funkcja prędkości $f'(x)$ (velocity), której argumentem będzie niezmiennie pozycja osi wiodącej. Druga pochodna pozycji $f''(x)$ to relacja zmian prędkości względem pozycji czyli przyspieszenie (acceleration). Trzecia natomiast $f'''(x)$ to dynamika przyspieszenia czyli tzw. zryw (jerk).



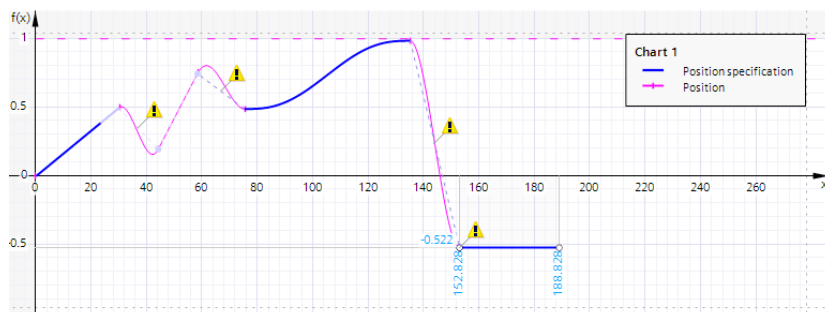
System oblicza i przedstawia powyższe przebiegi w formie graficznej w celu optymalizacji pracy układu. Funkcje te wykorzystywane są również celem interpolacji oraz mogą być wyznacznikiem ciągłości w celu wygładzenia profilu pozycji krzywki.

Interpolacja segmentów przejściowych

Rysując przebieg profilu krzywki przy pomocy edytora graficznego – możemy wykonać go w sposób ciągły (bez luk wartości w pełnym zdefiniowanym zakresie) lub wstawiając jedynie segmenty (lub punkty), które są kluczowe dla naszej aplikacji. Odcinki nieokreślone mogą finalnie zostać zdefiniowane przez wybór funkcji matematycznej w tabeli edytora.

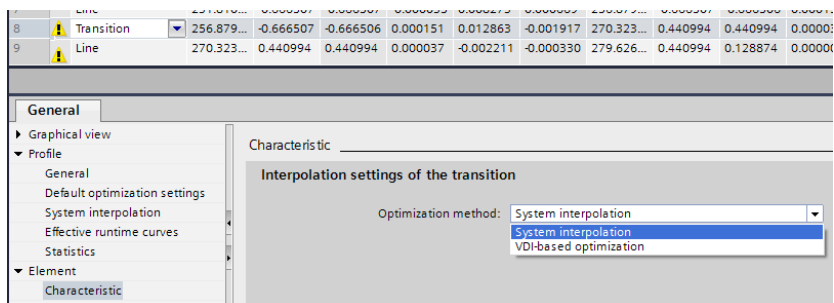
Line	ou
Transition	96
Point	18
Line	18
Sine	25
Transition	28
Inverse sine	31
Polynomial	

Ustawiając jednak funkcję segmentów przejściowych pomiędzy określonymi funkcjami na domyślnej opcji Transition – pozostawiamy ich interpolację mechanizmom systemowym.



Kolejne dwie zakładki w konfiguracji ogólnej profilu (*Default optimization settings* oraz *System interpolation*) określają parametry wybranego algorytmu optymalizacji (interpolacji) dla tych właśnie niezdefiniowanych segmentów przejściowych.

Dla każdego z nieokreślonych segmentów użytkownik może wybrać spośród dwóch metod interpolacji – VDI lub systemowej SIMATIC. Wyboru należy dokonać w konfiguracji danego segmentu czyli po jego zaznaczeniu w ostatniej zakładce edytora *Element*.



Jeżeli system w trakcie analizy tworzonego profilu ma jakieś ostrzeżenia dla poszczególnych elementów funkcji – informuje o tym bezpośrednio na wykresie przez wyświetlenie ikony ostrzegawczej ⚠. Skierowanie kursora myszy na symbol wykrzyknika spowoduje wyświetlenie komunikatu systemowego. Mogą być to informacje dotyczące nieciągłości wykresu (lub nakładania się punktów), problemów z kalkulacją efektywnych pochodnych, przekroczeń zakresów lub inne informacje dotyczące diagnostyki profilu. Nie ma konieczności usuwania wszystkich ostrzeżeń aczkolwiek jeśli jest to możliwe to warto to wykonać – z pewnością poprawi to jakość tworzonej funkcji.

Interpolacja systemowa

Wybierając opcję *System interpolation* jako metodę optymalizacji segmentu, należy określić algorytm jaki ma zostać zastosowany. Wyboru algorytmu dokonujemy w zakładce ustawień ogólnych profilu (*Profile* → *System interpolation*). Optymalizacja systemowa SIMATIC pozwala na interpolację liniową, funkcjami sklejającymi stopnia trzeciego oraz krzywymi Bézier-a.

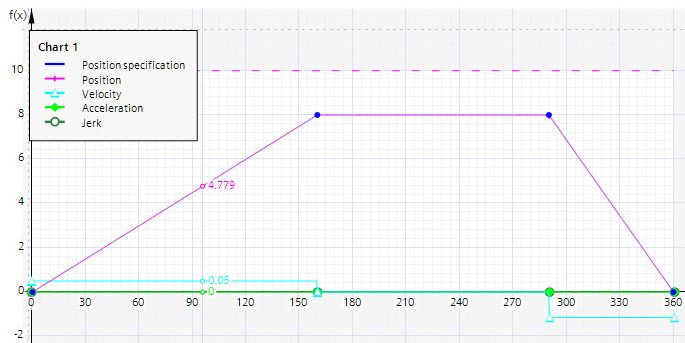
Korzystając z interpolacji systemowej możemy określić również charakterystykę segmentów skrajnych (*Behavior at boundary*) istotną przy rekurencyjnym wywołaniu profilu. Krzywka zostaje interpolowana w taki sposób aby jej pierwsza pochodna (prędkość) była równa na początku oraz na końcu profilu.

Rozważmy przykładowy profil krzywki, który został zdefiniowany przez 4 punkty w obszarze roboczym edytora graficznego. Wstawione punkty wyznaczają nam 3 segmenty, które wymagają interpolacji. Przykład ten może być bezpośrednim nawiązaniem do rozważanej aplikacji prasy, gdzie w pierwszej fazie cyklu – podajnik z chwytakiem przesuwają materiał (np. blachę) w obszar roboczy prasy, tam – faza druga – prasa wykonuje swoje zadanie (np. wytłoczenie, zgrzewanie lub wycięcie), a w etapie ostatnim – powrót chwytaka do pozycji wyjściowej. Wszystkie etapy pracy osi nadążnej skoordynowane są w trakcie jednego obrotu wiodącej osi rotacyjnej prasy mimośrodowej.

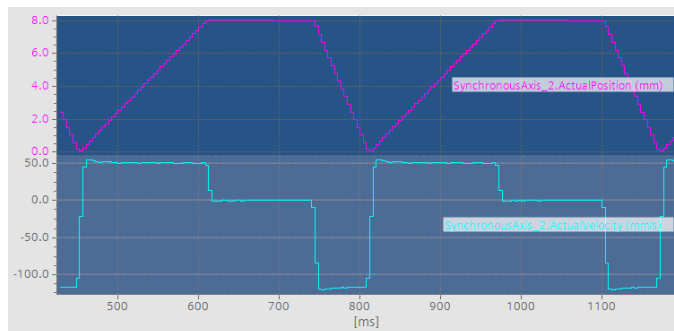
W celu przedstawienia działania algorytmów optymalizacyjnych – wstępnie nie będziemy określać odcinków dostępnymi funkcjami. Uwzględnimy również funkcję wygładzania pierwszej pochodnej na odcinkach skrajnych profilu – będzie miało to znaczenie przy optymalizacji nieliniowej (w cyklicznej pracy systemu).

Interpolacja liniowa (*Linear interpolation*)

Jest to szczególny przypadek aproksymacji za pomocą funkcji liniowej. W nawiązaniu do wcześniejszego opisu – w celach testowych – wygenerujemy przebieg profilu pozycji wraz z jego pochodnymi (pozycja, prędkość, przyspieszenie, zryw).

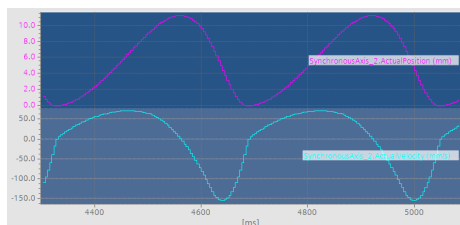
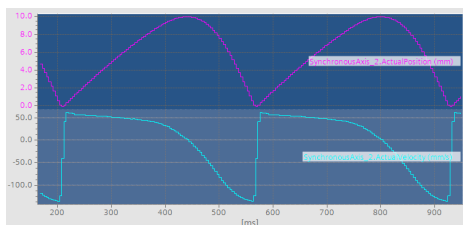
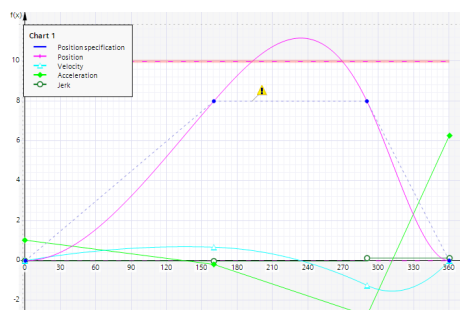
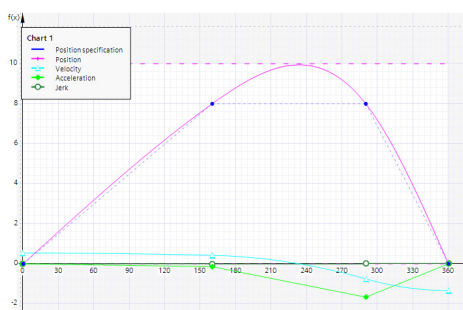


Metoda jest bardzo prosta i nie generuje przekroczeń limitów aczkolwiek w naszym przypadku powoduje nagłe zmiany dynamiki osi nadążnej, co będzie skutkowało szarpnięciem układu (zwłaszcza przy wyższej prędkości). Poniżej przykładowy rzeczywisty przebieg **pozycji** oraz **prędkości** osi nadążnej w czasie – dla powyższego rozwiązania wywoływanego cyklicznie. Możemy w łatwy sposób zaobserwować przeniesienie ustawień profilu na realny system.



Interpolacja funkcjami sklejanymi stopnia trzeciego (*Interpolation with cubic splines*)

Jest to metoda numeryczna polegająca na przybliżaniu nieznannej funkcji wielomianami niskiego stopnia. Algorytm ten jest często preferowany względem interpolacji wielomianowej ze względu na to, iż jego zastosowanie minimalizuje błędy aproksymacji.

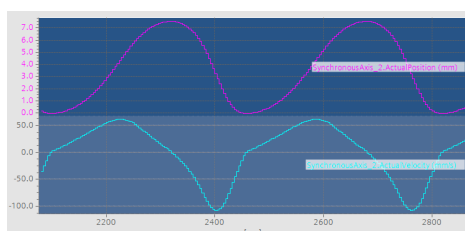
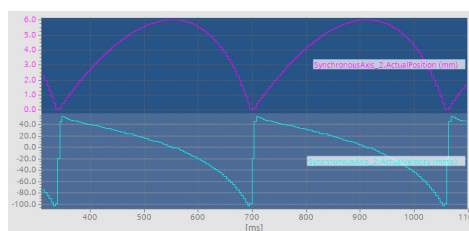
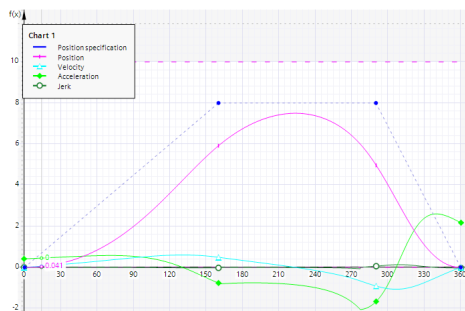
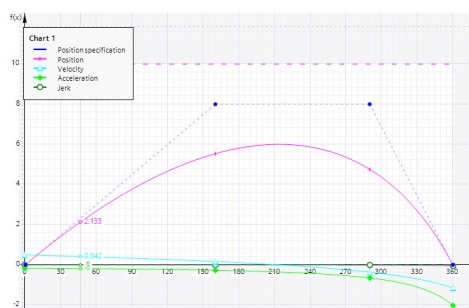


W naszym przykładzie algorytm sprawdzi się o wiele lepiej niż aproksymacja liniowa – zmiana pozycji będzie znacznie bardziej wygładzona, aczkolwiek możemy zaobserwować, iż w przypadku bezwarunkowej interpolacji (strona lewa) – podczas rekurencyjnego wywołania krzywki – nastąpi międzyfazowa nagle zmiana kierunku pracy osi (szarpnięcie). Z kolei w przypadku uwarunkowania ciągłości **pierwszej pochodnej** (strona prawa) – przejście do kolejnego cyklu jest gładkie, zwróćmy jednak uwagę, iż wyznaczona funkcja przekroczyła wyznaczony przez nas zakres wartości osi nadążnej.

Interpolacja krzywymi Bézier-a (Interpolation with Bézier splines)

Algorytm ten został opracowany w latach 60-tych przez francuskiego inżyniera firmy Renault. Krzywe Bézier-a są krzywymi parametrycznymi o charakterystyce wielomianowej. Znajdują szerokie zastosowanie w systemach optymalizacyjnych przetwarzania grafiki. Ze względu na cechy funkcji opisującej kształty, świetnie nadaje się także przy zagadnieniach optymalizacyjnych układów krzywkowych *Motion Control*.

Charakterystyczne dla optymalizacji krzywymi Bézier-a jest fakt, iż wyznaczona funkcja nie przebiega przez wszystkie punkty wskazane do interpolacji. W przypadku 4 punktów określonych w naszym przykładzie – algorytm wyznacza krzywą (3-ciego stopnia), która jest ograniczona przez jej styczne (w punkcie 1 oraz 4) wyznaczone przez pary punktów 1-2 oraz 3-4 (lewa strona – bez utrzymania ciągłości **pierwszej pochodnej** na skrajnych odcinkach profilu).



W naszym przykładzie ta metoda interpolacji wypada zdecydowanie najkorzystniej (zwłaszcza z uwzględnieniem utrzymania ciągłości prędkości na skrajnych odcinkach przebiegu – strona prawa) – nie powoduje przekroczeń zakresów wyznaczonych dla osi nadążnej, a także generuje optymalnie wygładzony kształt profilu. W związku z charakterystyką algorytmu wadą jednak może okazać się fakt, iż punkty 2 oraz 3 nie zostały ujęte w ścieżce osi nadążnej. Można to jednak zniwelować przez zmianę lokalizacji wyznaczonych wstępnie punktów lub przez połączenie różnych metod optymalizacji w konfiguracji przebiegu krzywki – o czym w dalszej części.

Wybierając interpolację systemową – wskazany algorytm zostaje zaordynowany dla wszystkich segmentów profilu, gdzie funkcja przejścia nie została określona. Aby zastosować inną metodą interpolacji dla różnych (nieopisanych funkcją) segmentów – możemy zastosować optymalizację VDI lub wybrać manualnie funkcję dla segmentu.

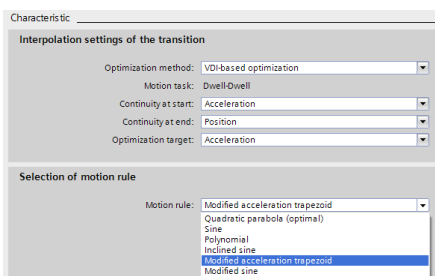
Optymalizacja VDI

Stowarzyszenie Inżynierów Niemieckich VDI (*Verein Deutsche Ingenieure*, www.vdi.eu) zdefiniowało wiele reguł w zakresie automatyzacji przemysłowej. Wśród nich znalazł się dokument (jeszcze z lat 80-tych) dotyczący standaryzacji zasad ruchu w układach ze sprzężeniem krzywkowym (*VDI Guideline 2143 – Motion rules for cam mechanism*). Wytyczne tej normy określają m.in. metodę optymalizacji profilu krzywki. Procedura znalazła miejsce jako alternatywa dla opcji systemowych proponowanych przez firmę Siemens.

Algorytm realizowany jest w oparciu o dwa wyróżniki:

- ciągłość wybranego parametru układu – pozycja, prędkość (bez wstrząsów), przyspieszenie (bez szarpnięć) lub zryw.
- cel optymalizacji (zgodnie z VDI) – brak określonego parametru, prędkość, przyspieszenie, zryw lub minimalna dynamika momentu.

Domyślne parametry optymalizacji VDI możemy skonfigurować globalnie (*Profile* → *Default optimization settings*). Dzięki temu wszystkie segmenty z wybraną interpolacją VDI będą estymowane w ten sam sposób. Z drugiej strony – korzystając z ustawień szczegółowych – każdy segment zdefiniowany dla tej metody interpolacji – sparametryzować możemy niezależnie (*Element* → *Characteristic*). Charakterystyka indywidualnych segmentów, daje również więcej kombinacji określających algorytm interpolujący, np. zróżnicowanie funkcji ciągłości na początku oraz końcu segmentu czy zastosowanie wybranej zasady określającej ruch (*Motion rule*) będącej częścią specyfikacji *VDI Guideline 2143*. Dostępne opcje filtrowane są w zależności od parametrów wybranych w panelu *Interpolation settings of the transition*.



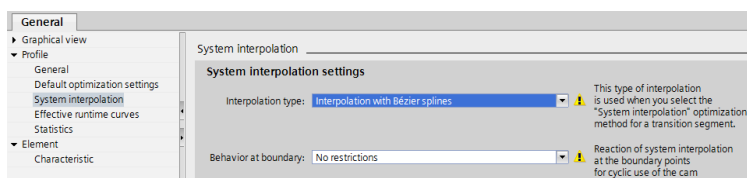
Więcej informacji odszukać można w tematach pomocy systemowej TIA Portal lub w samej dokumentacji organizacji VDI.

Z praktycznego punktu widzenia – korzystając z algorytmów VDI – uzyskamy efekty bardzo podobne do omówionej w poprzedniej sekcji interpolacji systemowej. W związku z powyższym w niniejszej dokumentacji nie będziemy zajmować się szczegółową analizą efektów optymalizacji VDI. Warto mieć jednak na względzie, iż pozwala ona (w przeciwieństwie do optymalizacji systemowej) na wybiórczą charakterystykę wskazanych segmentów, co można wykorzystać przy interpolacji mieszanej.

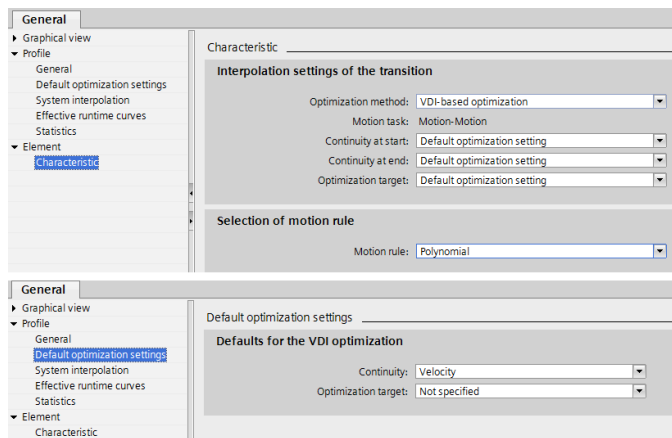
Interpolacja mieszana

Biorąc pod uwagę potrzeby naszej hipotetycznej aplikacji – możemy założyć, iż wskazane będzie chwilowe zatrzymanie obrabianego materiału po jego podsunięciu w obszar roboczy prasy. Założmy więc, że po wykonaniu 160° obrotu osi wiodącej (prasa), nastąpi wykonanie zadania (np. zgrzewanie), co będzie wymagało zatrzymania osi nadążnej (podajnik) na 130° obrotu osi *master*. Po tym etapie następuje powrót podajnika do pozycji wyjściowej (zostaje na tę fazę 70° obrotu osi wiodącej).

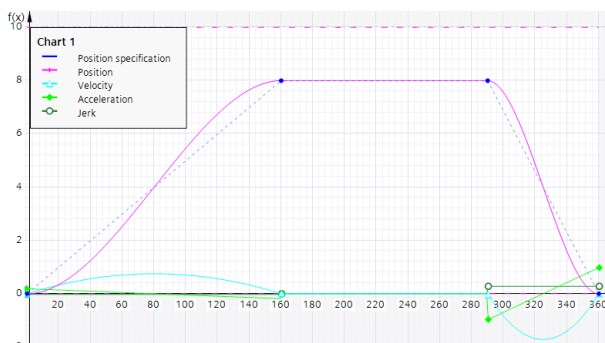
Zastosowanie połączenia dwóch omówionych metod interpolacji pozwoli uzyskać pożądany efekt. Konkretnie – dla segmentu pierwszego (pomiędzy punktami 1-2) oraz dla segmentu trzeciego (wyznaczonego przez punkty 3-4) wybierzmy optymalizację systemową krzywymi Bézier-a (bez wymogu utrzymania ciągłości pierwszej pochodnej na odcinkach skrajnych).



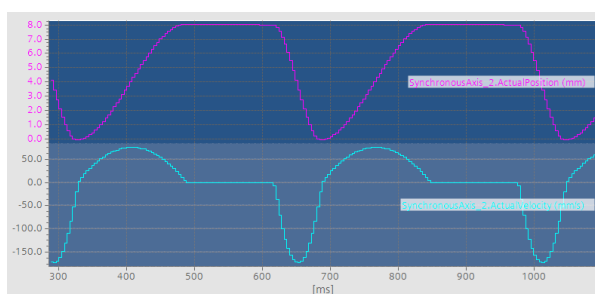
Z kolei dla segmentu środkowego (pomiędzy punktami 2-3) wybierzmy metodę VDI ze wskazaniem utrzymania ciągłości prędkości oraz bez określenia celu optymalizacji.



Powyższe ustawienia zaaplikowane dla naszego 4-punktowego przykładu spowodowały wygenerowanie następującego profilu krzywkowego.



Wynik interpolacji spełnia oczekiwania projektowe i wydaje się być optymalny z punktu widzenia dynamiki układu – nie ma przekroczeń zakresów, profil przechodzi przez wszystkie zdefiniowane punkty, a także prędkość ma charakterystykę ciągłą. Poniżej – podobnie jak w poprzednich przykładach – zaprezentowany został przebieg **pozycji** oraz **prędkości** osi nadążnej wygenerowany na rzeczywistym układzie dwóch serwonapędów.



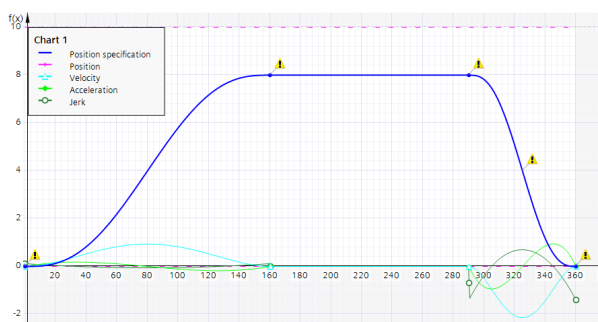
Czy rzeczywiście taki profil jest optymalny? Zależy to od wielu czynników. Dopiero wykonanie testów wytrzymałościowych (również długoterminowych) kompletnego systemu – pozwoli określić zużycie elementów, zbadać drgania układu, jego reakcje na zmiany częstotliwości oraz dynamiki pracy czy też oszacować jego żywotność.

Manualne określenie funkcji segmentów przejściowych

Na początku sekcji wspomnieliśmy o możliwości ręcznego zdefiniowania funkcji matematycznej dla segmentów przejściowych – bez konieczności ich interpolowania. Sprawdźmy więc jaki będzie efekt bezpośredniego wyboru dokonanego przez użytkownika.

Po wstawieniu punktów kluczowych dla naszego profilu – system automatycznie generuje (w tabeli poniżej obszaru rysowania) wiersze dla segmentów przejściowych. Pozostawiając tam wartość domyślną (*Transition*) wykonamy interpolację. Wskazując jednak bezpośrednio jedną z dostępnych funkcji, możemy określić matematyczną zależność pomiędzy dwoma punktami – dodatkowy punkt, linia, sinusoida bądź wielomian. Jeżeli jedna z tych opcji będzie dla naszego układu odpowiednia, nie będzie konieczności stosowania interpolacji.

W naszej aplikacji możemy spróbować wybrać wielomian dla obu segmentów skrajnych, natomiast dla środkowego linię. Uwaga – wybór dokonany w tabeli edytora można wycofać jedynie przez usunięcie segmentu z obszaru rysowania lub przez systemową funkcję *Undo*.



Łatwo zauważyć, iż otrzymany profil pozycji jest bardzo podobny do rezultatu interpolacji systemowej połączonej z VDI. Różnice dostrzegamy dopiero przy funkcjach przyspieszenia oraz zrywu. Ich istotność określają już testy uruchomieniowe.

Emulator parametrów dynamicznych

W zakładce Profile → *Effective runtime curves* możemy skonfigurować parametry krzywej trybu runtime dla osi wiodącej oraz nadążnej. Emulator programu sterownika analizuje przebieg sprężenia pozycji (oraz jego pochodnych) zgodnie z zadanymi parametrami (skalowanie, jednostki, dynamika) i prezentuje rezultat w obszarze roboczym edytora graficznego.

Wprowadzone modyfikacje nie są wgrywane do sterownika. Oznacza to, że interpolacja krzywki w CPU przebiega zgodnie z parametrami bazowymi. Wyznaczone współczynniki pozwalają jednak na testowanie oraz wizualizację dynamiki układu, np. symulacja reakcji na skalowanie w programie użytkownika przez parametry funkcji *MC_CamIn*.

Statystyka

Warto również zwrócić uwagę na liczbę wykorzystanych elementów konfiguracyjnych, która wskazana jest w ostatniej zakładce właściwości krzywki (*Profile* → *Statistics*). Maksymalnie możemy zdefiniować 1000 punktów oraz 50 segmentów graficznie opisujących profil. Będzie to istotne w przypadku gdy będziemy chcieli zastosować mechanizmy do modyfikacji profilu w trybie pracy, o czym w dalszej części dokumentu.

20.3 Interfejs użytkownika

Po zdefiniowaniu profilu krzywkowego naszego układu przechodzimy do programu użytkownika, gdzie sprzężenie zostanie uruchomione.

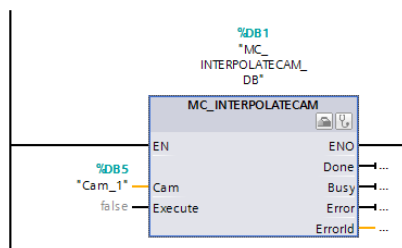
W poprzednim rozdziale (*Ruch skoordynowany*) odnajdziemy wiele analogii w konfiguracji obiektów technologicznych oraz funkcji interfejsu programistycznego użytkownika.

Przed wszystkim należy pamiętać, iż podstawowymi parametrami wejściowymi do naszej relacji krzywkowej są dwa standardowe obiekty technologiczne. Przynajmniej jeden z nich jest osią synchroniczną (oś nadążna), drugi natomiast może być osią pozycjonującą, synchroniczną lub enkoderem zewnętrznym. Konfiguracja obiektów technologicznych osi odbywa się identycznie jak w przypadku wcześniej omawianej synchronizacji liniowej – profil krzywkowy jest analogicznym mechanizmem sprzęgającym, z tym, że pozwala na definicję nieliniowej relacji pomiędzy osiami.

Różnica programistyczna jest również znikoma – konfiguracja programu w sterowniku sprowadza się do wywołania niemalże identycznej funkcji sprzęgającej oraz do uprzedniej interpolacji profilu krzywki w programie użytkownika.

Funkcja MC_InterpolateCam

Zgodnie z analizą przeprowadzoną w poprzedniej sekcji – efektem pracy graficznego interfejsu środowiska inżynierskiego jest opis trajektorii sprzężenia krzywkowego. Wyznaczyliśmy metodę interpolacji profilu (lub jego statyczny opis funkcjami matematycznymi) – teraz musimy wykonać praktyczną interpolację w programie użytkownika. Wynika to z faktu, iż wgranie określonego graficznie profilu do sterownika S7-1500T, powoduje jego ulokowanie w tymczasowej pamięci, do momentu jego interpolacji przez funkcję użytkownika. Dopiero wywołanie funkcji *MC_InterpolateCam* spowoduje aktualizację profilu zgodnie z konfiguracją obiektu technologicznego *TO_Cam*.



Podczas interpolacji opisane zostają wszystkie luki pomiędzy punktami oraz segmentami zdefiniowanymi w konfiguracji. Procedura przypisuje stałą wartość pozycji osi nadążnej dla każdej wartości zdefiniowanego zakresu pracy osi wiodącej.

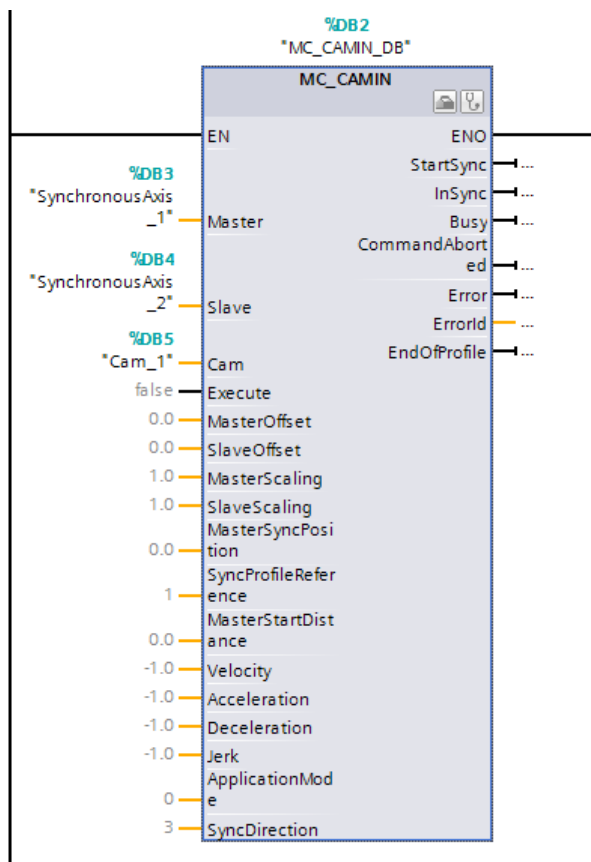
Teoretycznie – wgranie profilu, który w środowisku inżynierskim został opisany kompletnie, czyli zdefiniowane zostały wszystkie wartości osi nadążnej w pełnym zakresie pracy osi wiodącej (bez interpolacji) – nie wymaga zastosowania funkcji *MC_InterpolateCam*, a co za tym idzie spowoduje automatyczną reinicjalizację profilu w sterowniku zaraz po wgraniu nowej wersji projektu. W praktyce jednak, pominięcie chociażby jednego punktu, wymusi wywołanie funkcji interpolacyjnej. W związku z powyższym dobrą praktyką jest założenie, iż zawsze przed wywołaniem sprzężenia krzywkowego – powinna zostać wykonana interpolacja programowa. Nie jest to procedura, która w jakikolwiek sposób zaburzy pracę systemu lub utrudni konfigurację.

Pod parametr *Cam* podłączamy nasz obiekt technologiczny (można przeciągnąć metodą *drag&drop* z drzewka projektu), oraz podajemy zboczne narastające na wejście *Execute*.

Wyjście *MC_InterpolateCam.Done* dostarczy nam informacji o zakończeniu procedury. Dobrze jest zatem aby sygnał ten wykorzystać bezpośrednio jako wejście *MC_CamIn.Execute* aktywujące sprzężenie krzywkowe.

Funkcja MC_CamIn

W poprzednim rozdziale omówiliśmy szczegółowo zasady działania funkcji *GearInPos* służącej do aktywacji liniowej synchronizacji absolutnej osi. Analogicznym sprzężeniem (z tym, że nieliniowym) jest relacja krzywkowa. W związku z powyższym, funkcja realizująca dojsię do zaspzęglenia oraz jego utrzymanie w trybie pracy, ma sporo podobieństw.



Parametry podstawowe

Bazowymi wejściami bloku są w tym przypadku oczywiście nasze osie (*Master*, *Slave*), a także nieliniowa relacja pomiędzy nimi czyli nasz profil krzywkowy (*Cam*). Standardowo wywołanie funkcji odbywa się przez podanie zbocza narastającego na wejście *Execute*.

Skalowanie

Kolejne cztery parametry funkcji *MC_CamIn* (*MasterOffset*, *SlaveOffset*, *MasterScaling* oraz *SlaveScaling*) stanowią swego rodzaju odniesienie do przekładni, którą definiowaliśmy w przypadku liniowego sprzężenia osi. Pozwolą zatem zmodyfikować zakres operacji naszego profilu lub wykonać jego skalowanie. Więcej informacji na temat modyfikacji krzywek w trybie pracy aplikacji przedstawimy w dalszej części książki.

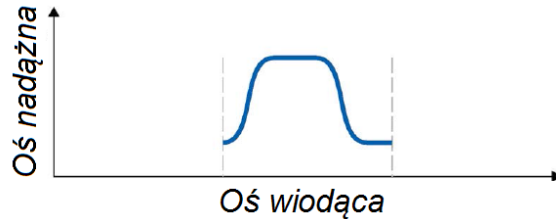
Dojście do zaszprężenia

Podobnie jak w przypadku funkcji *MC_GearInPos* – funkcja *MC_CamIn* pozwala określić parametry dojścia do zaszprężenia. Analogicznie możemy wybrać tutaj synchronizację „na drodze” (*SyncProfileReference* = 1) lub przez zdefiniowanie maksymalnych wartości parametrów dynamicznych (*SyncProfileReference* = 0). Opis tych dwóch trybów przedstawiony w rozdziale *Sprężenie absolutne osi* jest tutaj w pełni adekwatny.

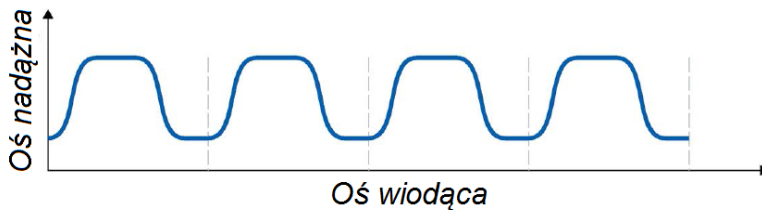
Tryb pracy

Jedyny całkowicie nowy parametr to *ApplicationMode*. Wartość tego parametru określa tryb wywołania krzywki w programie sterownika. Dostępne są trzy warianty:

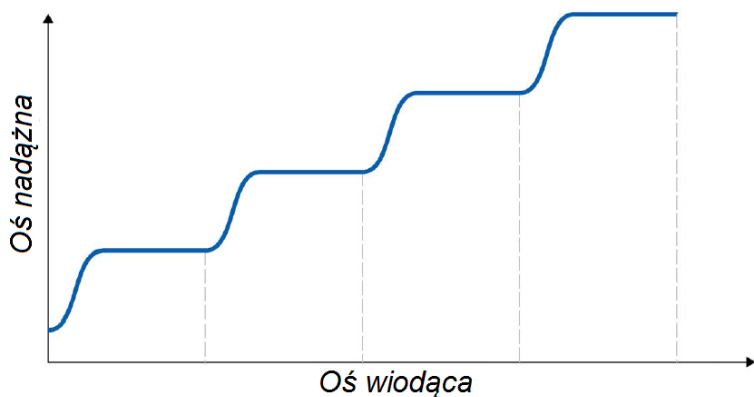
- wywołanie jednorazowe (*ApplicationMode* = 0) – ruch synchroniczny zakończony jest po osiągnięciu punktu końcowego profilu (lub początkowego jeśli wywołanie jest w kierunku przeciwnym). W celu zabezpieczenia układu przed skokowymi zmianami dynamiki – prędkość osi nadążnej musi być zerowa w skrajnych punktach profilu.



- wywołanie cykliczne (*ApplicationMode* = 1) – profil wykonywany jest rekurencyjnie, w momencie osiągnięcia punktu końcowego profilu, zostaje automatycznie powtórzony od punktu początkowego. Podobnie jak poprzednio – w celu zabezpieczenia układu przed skokowymi zmianami dynamiki – prędkość osi nadążnej musi być zerowa w skrajnych punktach profilu.



- wywołanie cykliczne dodawane (*ApplicationMode* = 2) – profil wykonywany jest rekurencyjnie, w momencie osiągnięcia punktu końcowego profilu, staje się on nowym punktem początkowym w kolejnej iteracji. W celu zabezpieczenia układu przed skokowymi zmianami dynamiki – prędkość osi nadążnej musi być ciągła w skrajnych punktach profilu.



Kierunek synchronizacji

Dla osi wiodącej o charakterystyce moduło, istotnym będzie również kierunek synchronizacji osi nadążnej. Ustawienie odpowiedniej wartości parametru *SyncDirection* pozwoli na wybór kierunku – zgodnie lub przeciwnie z przyrostem wartości, ewentualnie automatycznie w dowolnym kierunku najkrótszą ścieżką.

Parametry wyjściowe

Wyjścia funkcji *MC_CamIn* dostarczają informacji na temat statusu sprzężenia (sprężony, zajęty, procedura przerwana), rozpoczęcia oraz osiągnięcia ostatniego punktu profilu (status ustawiany przynajmniej na jeden cykl programu) oraz ewentualnych błędów.

Odsprężenie osi slave

Status sprzężenia osi synchronicznej profilu krzywkowego (*MC_CamIn.InSync* oraz *MC_CamIn.Busy*) pozostaje aktywny do momentu zakończenia pracy synchronicznej. Odsprężenie osi *slave* następuje w momencie wywołania dowolnej innej funkcji *Motion Control* na jej obiekcie technologicznym (*TO_SynchronousAxis*), np. przez zatrzymanie, zadanie prędkości bądź pozycjonowanie. Błąd osi nadążnej również spowoduje zakończenie pracy synchronicznej.

20.4 Edycja krzywek w programie PLC

Na początku rozdziału wspomnieliśmy o możliwości edycji zdefiniowanych profili krzywkowych, również w trybie pracy systemu. Istotnie, w zależności od wymagań aplikacji możemy zastosować różne metody pozwalające na elastyczne dostosowanie parametrów sprężenia, do potrzeb produkcyjnych bądź uruchomieniowych.

Podczas budowania oraz testowania układu bardzo przydatna jest łatwość edycji parametrów krzywki – zarówno z poziomu środowiska inżynierskiego, jak i programu sterownika. Sterownik S7-1500T pozwoli nam na elastyczną modyfikację parametrów sprężenia w celu usprawnienia fazy inżynierskiej projektu.

Modyfikacja projektu to jednak nie wszystko – jeszcze więcej zalet dostrzeże w otwartości systemu użytkownik końcowy, który z łatwością będzie mógł samodzielnie dostosować tryb pracy urządzenia, nawet bez dostępu do środowiska narzędziowego oraz projektu. Modyfikacja funkcji sprzęgającej w trybie pracy systemu sterowania, daje możliwość przezbrownienia linii na inny produkt, modyfikacji parametrów wytwarzanego produktu, dostosowania do zmian podzespołów lub półproduktów, kalibracji układu, zmiany punktu odniesienia osi lub ich skalowania oraz wiele innych zależnych od aplikacji funkcji, które mogą zostać wdrożone nawet przy wykorzystaniu jedynie urządzenia HMI.

Skoncentrujemy się teraz na edycji profilu w trybie pracy systemu. Rozważmy kilka wariantów, które dostępne są w sterownikach serii S7-1500T.

Statyczna definicja obiektów technologicznych

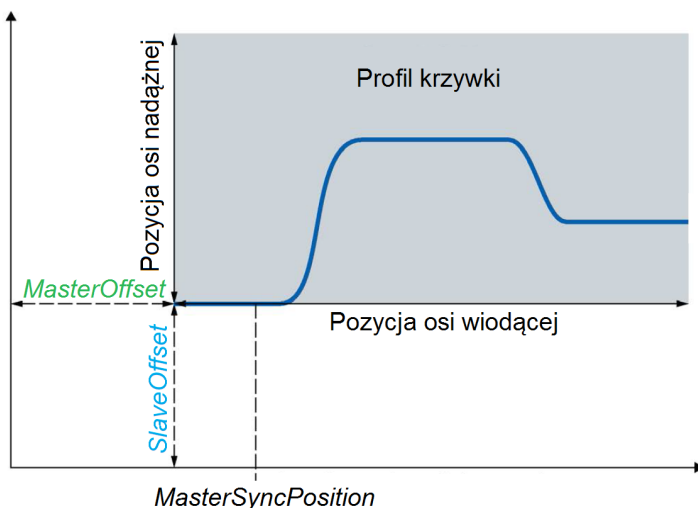
Przyjmijmy, że konstruowana maszyna ma zostać przystosowana do wytwarzania wielu produktów lub jego wariantów, co wiązać się będzie bezpośrednio z koniecznością zmiany profilu sprężenia osi skoordynowanych. Podstawową cechą systemu jest możliwość zdefiniowania wielu profili krzywkowych (tak jak wykonaliśmy to w poprzedniej sekcji), a następnie swobodna ich implementacja przez funkcje programu użytkownika. Taka metoda sprawdzi się w układach, gdzie będziemy w stanie określić zapotrzebowanie na konkretne funkcje krzywkowe, już na etapie inżynierskim. W programie sterownika trzeba jedynie wywołać odpowiednią ilość razy funkcję *MC_InterpolateCam* oraz *MC_CamIn*.

Metoda ta, przy każdej zmianie profilu, wymagać będzie interpolacji oraz aktywacji krzywki, aczkolwiek definicja wielu obiektów technologicznych *TO_Cam* nie zajmuje zasobów *Motion Control*, których liczba dla danego typu CPU jest ograniczona (*Tools→Resources→Motion Control resources*).

Przesunięcie profilu

Idąc krok dalej – założmy, że mamy zdefiniowane statycznie profile krzywek podobnie jak w powyższym przykładzie, z tym, że chcemy mieć również możliwość edycji tychże funkcji (przez program sterownika). Adaptacja profilu może odbywać się przez parametry funkcji sprzęgającej (*MC_CamIn*) lub przez modyfikację technologicznego bloku danych. Zaczniemy od podstawowych, oraz łatwych w obsłudze, opcji dostępnych z poziomu funkcji interfejsu użytkownika.

Wspomniane już parametry *MasterOffset* oraz *SlaveOffset* pozwalają na zdefiniowanie wektora przesunięcia całego profilu na płaszczyźnie zależności osi wiodącej względem nadążnej. Innymi słowy możemy przesunąć punkt startowy profilu, a co za tym idzie – każdy kolejny, który został jednoznacznie wyznaczony podczas interpolacji. Sytuację ilustruje poniższa grafika.



Zwróćmy również uwagę, iż przesuwając całość profilu – przemieszczamy także pozostałe parametry zdefiniowane dla sprzężenia, czyli np. punkt dojścia do zasprężenia określony pozycją osi wiodącej (*MasterSyncPosition* – patrz *Synchronizacja absolutna*) czy zakres wartości osi nadążnej.

W przypadku przesunięcia wertykalnego (*SlaveOffset*) zmieniamy zakres wartości, w którym pracować będzie oś nadążna. Przykładowo przyjmijmy domyślny zakres osi 0-10 mm. Jeśli przesuniemy go o 15 mm, zakres zostanie zmieniony na 15-25mm. Trzeba tutaj zwrócić uwagę na sytuację gdzie dojście do zasprężenia realizowane będzie „na drodze”. Zależnie od pozycji wyjściowej osi nadążnej – dystans przeznaczony na synchronizację nie ulegnie zmianie, natomiast odległość jaką będzie musiała pokonać oś nadążna, już tak.

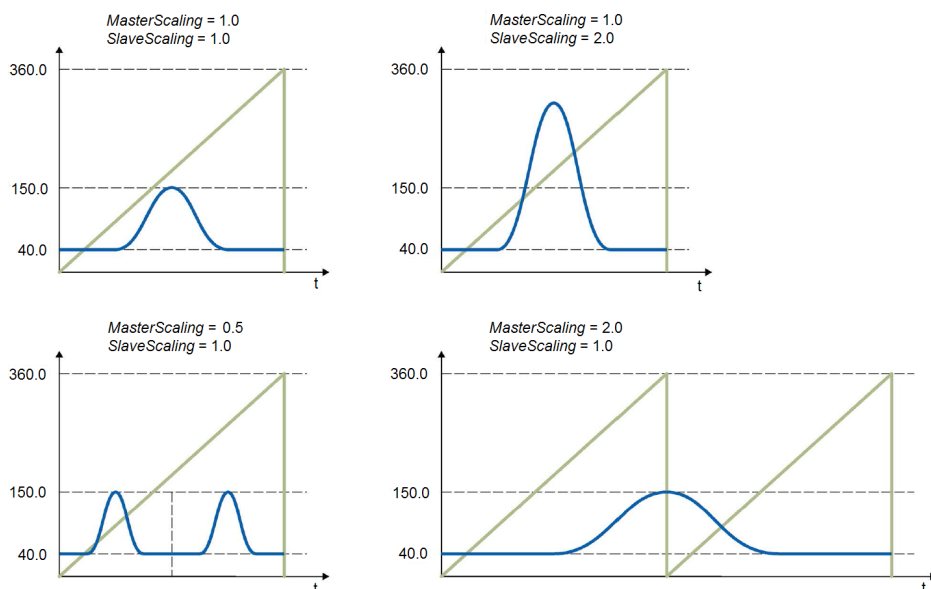
Wykorzystując przesunięcie horyzontalne profilu (*MasterOffset*) nie zmieniamy programowego zakresu wartości osi wiodącej (np. przy osi obrotowej modułu), a jedynie przesuujemy jej charakterystykę względem osi nadążnej. Od strony dynamiki, dojście do zasprężenia będzie realizowane więc analogicznie jak w sytuacji bez przesunięcia, z tym, że w innym punkcie osi wiodącej.

Przesunięcie profilu nie wymaga ponownej interpolacji – wystarczy podanie zbocza narastającego na wejście *Execute* funkcji *MC_CamIn*. Zmiana powyższych parametrów nie wprowadza zmian do krzywki zdefiniowanej w środowisku inżynierskim.

Skalowanie profilu

Poza możliwością przesunięcia całości zdefiniowanej funkcji, wraz z zakresem jej wartości oraz argumentów, funkcja systemowa umożliwia także zmianę „zagęszczenia” wartości względem osi układu współrzędnych. Innymi słowy możemy rozciągnąć lub skurczyć zdefiniowany profil, czyli wykonać jego skalowanie. W celu realizacji tego zadania przygotowano zostały dwa parametry funkcji *MC_CamIn* – *MasterScaling* oraz *SlaveScaling*. Odpowiadają one za skalowanie wartości osi master oraz osi slave.

Poniższe przebiegi czasowe, ilustrują reakcję systemu na wybrane wartości parametrów skalujących zakres wartości osi **wiodącej** oraz **nadążnej**.



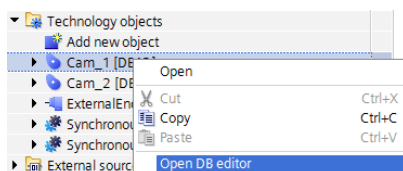
Podobnie jak w przypadku przesunięcia profilu – skalowanie powoduje zmianę zakresu pracy osi nadążnej lub rozkład jej wartości względem osi wiodącej. Nie ulegają zmianie parametry dojścia do zaszprzęglenia (np. *MasterSyncPosition*). Skalowanie profilu nie wymaga ponownej interpolacji – wystarczy podanie zbocza narastającego na wejście *Execute* funkcji *MC_CamIn*. Zmiana powyższych parametrów nie wprowadza zmian do krzywki zdefiniowanej w środowisku inżynierskim.

Tworzenie krzywki w trybie RT

Powyższe metody z pewnością znajdą zastosowanie w systemach, gdzie liczba profili będzie ograniczona lub różne warianty produkcyjne będą wiązać się jedynie z wektorową lub proporcjonalną modyfikacją zdefiniowanych sprzężeń. Co w takim razie w przypadku gdy zajdzie potrzeba stworzenia całkowicie nowego profilu krzywki?

Przejdźmy do najbardziej zaawansowanej oraz otwartej metody jaką możemy tutaj wykorzystać, a mianowicie do swobodnej edycji profilu przez modyfikację obiektu technologicznego z poziomu programu sterownika.

Zacznijmy jednak od kilku zdań wstępu. Z poprzednich rozdziałów wiemy już, że każdy obiekt technologiczny, który wstawiamy do projektu jest w rzeczywistości tzw. technologicznym blokiem danych, czyli blokiem danych, który posiada zintegrowany interfejs usprawniający konfigurację, uruchomienie oraz diagnostykę osi. Pomijając jednak edytory graficzne, mamy do czynienia po prostu z blokiem danych. Łatwo zaobserwować to przez opis wstawionego do projektu obiektu technologicznego, który zawsze zawiera numer przydzielonego przez system DB. Istnieje również możliwość otwarcia widoku źródłowego technologicznego bloku danych przez wybór z menu podręcznego opcji *Open in DB editor*.



Widok tabelaryczny odzwierciedla wszelkie parametry konfiguracyjne, które ustawimy przez jego graficzny odpowiednik. W przypadku *TO_Cam* znajdziemy tutaj zarówno szczegółowe informacje na temat statusu, błędów lub ostrzeżeń generowanych przez obiekt technologiczny, jak i samą konfigurację profilu – od ustawień osi wykresu, przez parametry interpolacji, a kończąc na zdefiniowanych punktach oraz segmentach funkcji sprzęgającej.

Zgodnie z wcześniejszym opisem – definicja profilu pozwala na wyznaczenie maksymalnie 1000 punktów oraz 50 segmentów opisujących relację krzywkową. Strukturę tych informacji możemy odczytać bezpośrednio z technologicznego bloku danych.

	Name	Data type	Start value
9	ValidPoint	Array[1..1000] of Bool	
10	Segment	Array[1..50] of TO_Cam_Str...	
11	Segment[1]	TO_Cam_Struct_SegmentDa	
12	xmin	LReal	0.0
13	xmax	LReal	160.0
14	a0	LReal	0.0
15	a1	LReal	0.0625372361093122
16	a2	LReal	-7.83577256832016E-05

W przykładowym profilu – segment pierwszy zawiera się w przedziale 0.0 – 160.0 wartości osi wiodącej. Pozycja wyjściowa osi nadążnej w tym zakresie ma wartość 0.0 ($a0$). Możemy też odczytać początkową wartość pochodnych w tym punkcie ($a1$, $a2$, $a3$), co można skorelować z wartościami przedstawionymi w edytorze graficznym.

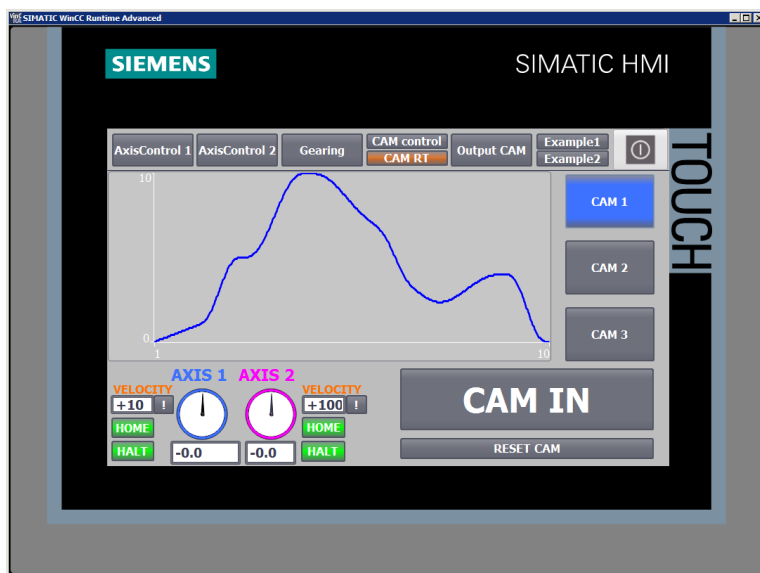
Wartości zawarte w bloku danych mogą nam posłużyć do graficznej reprezentacji krzywki, np. na ekranie panelu operatorskiego. Kontynuując jednak drogę do edycji profilu – możemy tutaj również wykonać modyfikację parametrów sprzężenia. Zmiana wartości rekordów bloku danych w środowisku inżynierskim, nie przyniesie jednak oczekiwanego efektu, gdyż nadrzędnym źródłem danych jest edytor graficzny. W związku z tym – wszelkie zmiany zostaną przywrócone (do ustawień początkowych) podczas kompilacji projektu. Jesteśmy w stanie jednak trwale wpływać na zawartość technologicznego bloku danych w trybie pracy aplikacji. Możemy rozważyć tutaj wariant gdzie profil mamy wstępnie zdefiniowany w środowisku inżynierskim lub utworzony został jedynie pusty obiekt technologiczny *TO_Cam*, który zostanie uzupełniony danymi podczas pracy systemu.

Modyfikacja krzywki od strony technologicznego bloku danych, nie jest zadaniem prostym. Z pewnością wymaga szerokiej wiedzy – zarówno programistycznej jak i znajomości technologii uruchamianego układu. Jest to oczywiście zadanie wykonalne aczkolwiek wychodząc naprzeciw integratorom oraz użytkownikom końcowym firma Siemens przygotowała przykład aplikacji, który w łatwy sposób można wykorzystać do zrealizowania opisywanego zadania.

Jest to gotowa biblioteka, która będzie doskonałą bazą do bezpośredniego wdrożenia w aplikacji ze sprzężeniem krzywkowym.

Szczegółowy opis zagadnienia, dokumentacja, a także przykładowy projekt można pobrać bez ograniczeń w następującej lokalizacji sieciowej:
<https://support.industry.siemens.com/cs/pl/en/view/105644659>

Opracowany przykład pozwala nie tylko swobodnie przekształcić profil krzywkowy w trybie pracy aplikacji, ale również daje możliwość wyświetlenia (np. na panelu operatorskim) aktualnie zaimplementowanego profilu.



VI. Wejścia pomiarowe wyjścia krzywkowe

Aby sprostać najbardziej wymagającym aplikacjom przemysłowym, nowoczesne układy serwonapędowe muszą zapewnić, nie tylko wysoką dokładność pozycjonowania, ale również ściśle określoną reakcję na odczytane sygnały oraz zdarzenia.

Charakterystyka układu pomiarowego oraz jego parametry zostały już omówione w poprzednich rozdziałach. W dalszej części zajmiemy się możliwościami systemów SIMATIC w zakresie dokładnej detekcji zmian sygnałów w czasie, a także precyzyjnej odpowiedzi układu na wykonane pomiary.

W celu realizacji zadań o wysokiej precyzji, nadrzędny system sterownia powinien zostać wyposażony w odpowiednią funkcjonalność. Od strony sprzętowej potrzebny będzie tzw. moduł technologiczny. Programowo z kolei, po raz kolejny zajmiemy się implementacją odpowiedniego algorytmu m.in. przy pomocy obiektów technologicznych.

21

Moduły technologiczne

Dla rozwiązań wymagających bardzo wysokiej wydajności, systemy SIMATIC przewidują zastosowanie dedykowanych modułów funkcyjnych. Ważną kwestią jest tutaj również odciążenie jednostki centralnej – w związku z tym zadania technologiczne wykonywane są po stronie samego modułu, a nie w jednostce centralnej sterownika. W naszych rozważaniach istotne będą funkcje pomiarowe, zliczające, pozycjonujące oraz zadania oparte o ściśle określony czas reakcji.

W rozdziale Zagadnienia podstawowe omówiliśmy szczegółowo zasadę działania trybu izochronicznego, którego zastosowanie jest kluczowe w konfiguracji modułów dedykowanych funkcjom technologicznym. W związku z powyższym każdy z przedstawionych modułów musi obsługiwać, a w konfiguracji odpowiednich funkcji technologicznych – wykorzystywać funkcjonalność trybu IRT.

Zgodnie z informacjami z rozdziału *Zagadnienia podstawowe* – obsługa trybu izochronicznego możliwa jest jedynie przez połączenie rozproszonego systemu wejść/wyjść bezpośrednio do portu komunikacyjnego (np. PROFINET) jednostki centralnej S7-1500T. Dlatego też rozważać będziemy moduły dedykowane dla zdecentralizowanych systemów I/O z rodziny ET 200SP/MP. Poniżej przedstawimy aktualnie dostępne moduły technologiczne.

Moduł licznikowy TM Count



W rozdziale *Układ pomiarowy* przedstawiliśmy konstrukcję oraz funkcjonalność urządzeń pomiarowych pozycji. Jako moduł technologiczny posiadający wejście cyfrowe przeznaczone dla enkoderów inkrementalnych – wyszczególniony został moduł technologiczny *TM Count*. Karta ta w naszym przypadku znajduje głównie takie właśnie zastosowanie. Jest to jednak urządzenie uniwersalne – o szerszym zakresie funkcji. *TM Count* pozwala również na realizację zaawansowanych zadań zliczających oraz pomiarowych. Funkcje takie znajdują częste zastosowanie w przemyśle spożywczym, papirniczym bądź przy urządzeniach pakujących.

Moduł licznikowy *TM Count* pozwala na realizację następujących zadań technologicznych:

- podłączenie enkoderów inkrementalnych oraz impulsowych 24V,
- zliczanie impulsów (dowolny przedział w zakresie $\pm 2^{31}$) z częstotliwością do 200kHz,
- funkcje pomiarowe – obliczanie częstotliwości, prędkości oraz długość impulsu,
- sterowanie procesem zliczania przez zintegrowane wejścia cyfrowe (ustawianie lub przechwytywanie aktualnej wartości licznika),
- szybką reakcję na zliczoną lub zmierzoną wartość przez zintegrowane wyjścia binarne.

Powyższe funkcje realizowane są przez autonomiczny procesor modułu technologicznego. Dzięki temu, możemy uzyskać precyzyjne pomiary oraz krótkie czasy odpowiedzi, równocześnie odciążając pamięć operacyjną jednostki centralnej.

Przykładem zastosowania modułu *TM Count* może być odczytanie pozycji kątowej osi obrotowej modułu, a w określonym zakresie kąta obrotu ustawienie wyjścia cyfrowego – np. nadruk etykiety na opakowaniu w urządzeniach typu flowpack.

Moduł licznikowy i pozycjonujący TM PosInput



Nieco bardziej zaawansowanym modułem technologicznym jest *TM PosInput*.

Poza funkcjonalnością zgodną z poprzednikiem (zliczanie, mierzenie) umożliwia dodatkowo:

- odczyt wejść z częstotliwością do 1MHz,
- podłączenie enkoderów z wyjściami TTL (5V),
- podłączenie enkoderów absolutnych w standardzie SSI.

Typowe zastosowanie modułu *TM PosInput* pozwala na realizację zadań takich jak:

- detekcja pozycji,
- obliczanie prędkości,
- pomiar długości produktów,
- dozowanie, mierzenie, etc.

TM Pulse – sterowanie zaworami proporcjonalnymi oraz silnikami prądu stałego

Dla systemu ET 200SP dostępny jest moduł technologiczny z wyjściami impulsowymi – dedykowany do sterowania zaworami, silnikami DC, silnikami krokowymi oraz innymi urządzeniami wykonawczymi (również napędami serwo – impulsowo). Karta *TM Pulse* posiada m.in. następujące właściwości:

- jeden kanał wyjściowy 4A lub 2 kanały po 2A,
- maksymalna częstotliwość wyjść 100kHz,
- tryby pracy PWM/PTO lub stała częstotliwość,
- sterowanie silnikami DC z uwzględnieniem kierunku obrotów,
- opóźnienie wyłączenia (dokładność do μ s).

Moduł wejść/wyjść czasowych TM Timer DIDQ



Dla aplikacji krytycznych czasowo przewidziany został moduł wejść/wyjść binarnych typu time-based. Oznaczenie to, sugeruje że zintegrowane kanały pozwalają na odczytywanie oraz wystawianie sygnałów z najwyższą dokładnością w czasie. Podobnie jak w przypadku poprzednich modułów technologicznych – *TM Timer DIDQ* posiada autonomiczny procesor, a więc sygnały przetwarzane są w cyklu niezależnym od parametrów magistrali komunikacyjnej oraz wydajności jednostki centralnej. Dzięki temu możemy osiągnąć rozdzielczość na poziomie 1 mikrosekundy!

Funkcjonalność modułu *TM Timer DIDQ* pozwala na odczyt zmian sygnałów wejściowych oraz bardzo precyzyjną rejestrację stempli czasowych tych zdarzeń. Wyjścia zaś mogą zostać ustawione w określonym punkcie w czasie.

Zintegrowane wejścia cyfrowe posiadają następujące funkcje:

- rejestrację do 2 stempli czasowych na cykl aplikacji,
- 32-krotne programowe zwiększenie rozdzielczości odczytu stanów wejść, tzw. nadpróbkowanie czyli oversampling,
- funkcja licznika o częstotliwości do 50 kHz,
- możliwość podłączenia enkodera inkrementalnego,
- opóźnienie wyłączenia (dokładność do μ s).

Wyjścia cyfrowe charakteryzują się:

- rejestrację do 2 stempli czasowych na cykl aplikacji,
- 32-krotnym programowym zwiększeniem rozdzielczości zapisu stanów wyjść, tzw. nadpróbkowanie czyli oversampling,
- funkcją modulacji szerokości impulsów (wyjście PWM).

Typowe aplikacje jakie można zrealizować przy pomocy modułu *TM Timer*:

- precyzyjne określenie czasu reakcji układu,
- wyjścia krzywkowe,
- pomiar długości obiektów,
- dozowanie płynów,
- pomiar czasu.

Najbardziej zaawansowanym modulem – zarówno w zakresie technicznych możliwości sprzętowych, jak i podejścia programowego do implementowanych algorytmów – jest moduł *TM Timer*. W związku z powyższym w dalszej części dokumentu zajmiemy się charakterystyką tego właśnie komponentu oraz jego zastosowań w zakresie współpracy ze sterownikiem S7-1500T oraz obiektami technologicznymi.

22 Wejścia/wyjścia typu time-based

Zajmijmy się teraz nieco bardziej szczegółowo modulem wejść/wyjść czasowych (*time-based*) *TM Timer DIDQ*. W niniejszej sekcji przedstawimy możliwości techniczne oraz funkcyjne jakie daje ten moduł technologiczny.

Karta występuje w dwóch wersjach – dla systemu wejść/wyjść ET 200SP oraz w standardzie S7-1500/ET 200MP. W tej pierwszej wersji moduł posiada zintegrowane 4 wejścia oraz 6 wyjść cyfrowych. W drugiej zaś 16 swobodnie konfigurowalnych kanałów (maksymalnie 8DI lub 16DO).

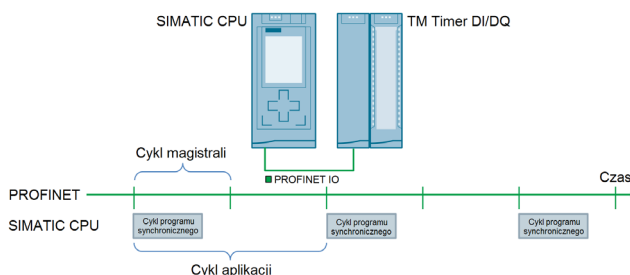
Aby dobrze zrozumieć funkcjonalność modułu w szybkich aplikacjach synchronicznych, zaczniemy od omówienia kilku zagadnień podstawowych. Zostały już one wstępnie przedstawione, aczkolwiek obrazowe wyjaśnienie kilku zasadniczych kwestii, uczyni konfigurację łatwą oraz transparentną.

Tryb izochroniczny

Moduły technologiczne *TM Timer* zapewniają obsługę trybu izochronicznego, którego charakterystykę przedstawiliśmy w rozdziale *Zagadnienia podstawowe*. Systemowa funkcja IRT obsługiwana jest w strukturze rozproszonych wejść/wyjść, m.in. dla sieci PROFINET IO. Tryb pracy synchronicznej wymagany jest w celu realizacji zadań technologicznych, których opis przedstawimy w niniejszej sekcji: funkcje czasowe, pomiarowe czy oversampling. W trybie izochronicznym cykl programu użytkownika, magistrali komunikacyjnej, odczytu sygnałów wejściowych oraz ich przetwarzanie przez procesor modułu technologicznego są zsynchronizowane. Stempel czasowy, wartości liczników, ciąg bitów funkcji nadpróbkowania oraz bity statusowe przetwarzane są w stałych odstępach czasu determinowanych przez cykl aplikacji.

Cykl magistrali oraz cykl aplikacji

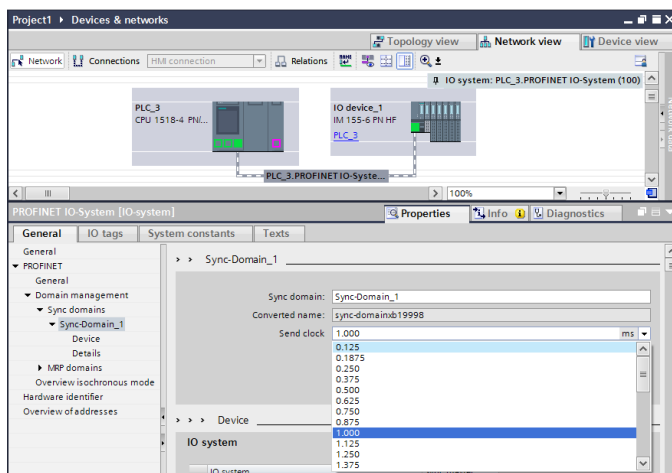
W konfiguracji naszego układu scentralizowanego definiowaliśmy cykl wykonywania programu przez określenie częstotliwości wykonywania bloku synchronicznego MC-Servo (OB91). W zależności od wybranej struktury systemu oraz programowego podejścia do kwestii synchronicznego wywołania programu – podczas parametryzacji w TIA Portal spotkamy się z pojęciami cyklu aplikacji (*Application cycle*) oraz cyklu magistrali (*Send clock*). Te generatory będą wywoływać cyklicznie synchroniczny program użytkownika. Pracę układu w trybie izochronicznym prezentuje poniższa grafika.



Cykl magistrali

W celu zapewnienia wysokiej dokładności oraz determinizmu przetwarzania stempli czasowych w CPU SIMATIC, moduł technologiczny *TM Timer* powinien być podłączony do jednostki centralnej przez PROFINET izochroniczny.

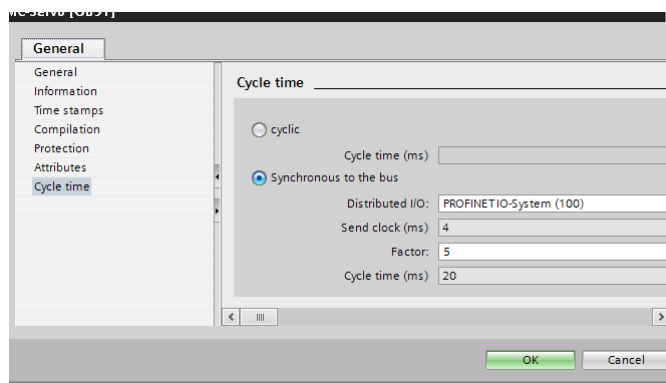
Praca w trybie IRT oznacza, iż dane pomiędzy modułem technologicznym, a jednostką centralną, wymieniane będą w ściśle określonych odstępach czasu. Ten stały odcinek czasu nazywany jest zegarem magistrali (*send clock*) czyli jest jej cyklem izochronicznym. Cykl zegara można zdefiniować maksymalnie do 4ms. Jego wartość minimalna zależna jest od możliwości mastera synchronizacji (CPU) – w przypadku najmocniejszych jednostek SIMATIC (CPU 1518) można ustawić jego wartość nawet na 125µs.



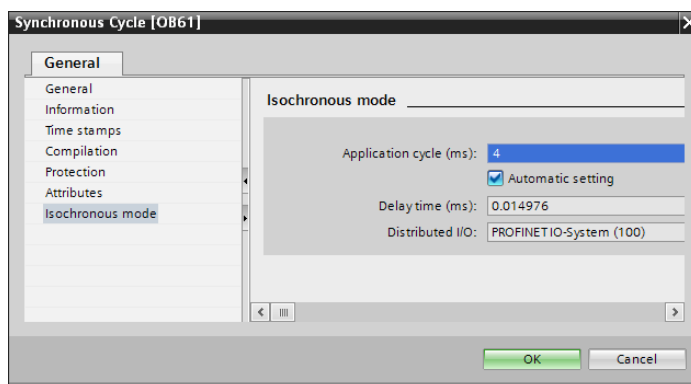
Konfiguracja wykonywana jest w ustawieniach sieci PROFINET (*Devices & networks*) po uprzednim wyznaczeniu mastera domeny synchronizacyjnej (*Sync-Domain* → *IO Device*).

Cykl aplikacji

Generator zegara magistrali komunikacyjnej jest podstawą do wyznaczenia cyklu pracy aplikacji, który jest kluczowy dla naszego projektu. Cykl aplikacji może pokrywać się z cyklem magistrali jeśli wszystkie urządzenia domeny synchronizacyjnej obsługują określony cykl. W przypadku napędów SINAMICS V90 ograniczeniem są 2ms i to w naszej konfiguracji będzie również ograniczeniem całego układu. Cykl aplikacji określamy jako wielokrotność zegara sieci PROFINET IRT. Parametr ten będziemy konfigurować w ustawieniach synchronicznego bloku organizacyjnego jednostki centralnej. Zgodnie z informacjami z poprzednich sekcji w przypadku naszej konfiguracji będzie to blok organizacyjny generowany automatycznie przez system w momencie dodania obiektu technologicznego czyli *MC-Servo*. Poniżej zakładka ustawień bloku OB91:



Można również wygenerować blok synchroniczny magistrali ręcznie przez dodanie bloku OB61 (*Synchronous Cycle*), w tym wypadku ustawienia wyglądają następująco:



Konfiguracja manualna zostanie opisana w dalszej części dokumentu jako alternatywa dla rozwiązania z użyciem obiektów technologicznych jednostki centralnej S7-1500T.

Bloki synchroniczne zawierają instrukcje naszego programu synchronicznego – czyli np. zadania realizowane przez moduł *TM Timer*. W przypadku bloku OB61 będziemy konfigurować logikę programu, funkcje pomiarowe, wyjścia krzywkowe, przetwarzanie stempli czasowych, etc. – wewnątrz bloku. Wykorzystując obiekty technologiczne typu *Measuring Input* czy *Output Cam* program będziemy realizować standardowo w bloku organizacyjnym (np. OB1), a system sam zapewni wywołanie odpowiednich funkcji w cyklicznym bloku OB91. Jest to ułatwienie dla programisty oraz czyni projekt spójnym i czytelnym. Blok *MC-Servo* nie może być edytowany przez użytkownika.

Należy również pamiętać, iż w celu zapewnienia przewidywalnej pracy systemu – synchroniczny blok organizacyjny powinien mieć możliwie wysoki priorytet, gdyż jest to blok generujący cykliczne przerwanie programowe wszystkich bloków o priorytecie niższym. Jeśli w projekcie znajdzie się zarówno blok OB61 jak i OB91, któryś z nich będzie dla sterownika ważniejszy. W celu uzyskania oczekiwanej dynamiki układu oraz zapewnienia bezbłędnej pracy sterownika nadrzędnego – zaleca się stosowanie jednego bloku synchronicznego na projekt.

Cykl programu synchronicznego

Wywołanie bloku synchronicznego reprezentuje cykl pracy naszej aplikacji. Trzeba dobrać jego długość w sposób optymalny. Jeśli wydłużymy go w stosunku do cyklu magistrali (wielokrotność cyklu zegara), odciążymy zasoby obliczeniowe CPU, aczkolwiek możemy równocześnie doprowadzić do zmniejszenia dokładności naszej aplikacji. Trzeba więc stwierdzić jakiego rzędu dokładność będzie wymagana. Z drugiej jednak strony cykl aplikacji nie może być zbyt krótki gdyż musi trwać wystarczająco długo, aby zdążył wykonać się cykl programu synchronicznego, czyli innymi słowy odczyt wejść, ustawienie wyjść oraz zaprogramowana przez użytkownika logika programu.

System automatycznie, już na etapie konfiguracji w TIA Portal, oblicza czas wymagany na wykonanie zadań odczytu/zapisu (T_i/T_o) dla każdego urządzenia typu *slave* w sieci PROFINET IO IRT. Kalkulacja ta znajduje się w ustawieniach portu komunikacyjnego (*Advanced options* → *Isochronous mode*). Przykładowo, możemy zaobserwować pełną konfigurację powyższych parametrów, dla wyspy sygnałowej ET 200SP z modułem *TM Timer* oraz *TM Count* pracującymi w trybie izochronicznym.

Isochronous mode	
☒ Isochronous mode	
Send clock:	2.0000 ms
Application cycle:	4.000 ms
TiTo values:	Automatic minimum
Time T_i (read in process values):	0.17775 ms
Intervals:	0.000125 ms
Time T_o (output process values):	0.209875 ms
Intervals:	0.000125 ms

Jeśli czas wykonania cyklicznego programu synchronicznego przekroczy cykl aplikacji, zapętlony zostanie stos przerwań, co spowoduje błąd sterownika.

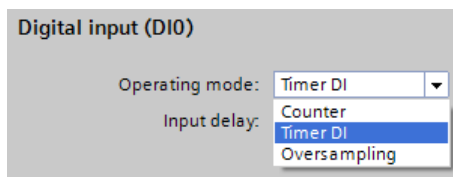
Interfejs modułu technologicznego

W dalszej części będziemy odwoływać się do właściwości modułu *TM Timer*. Często pojawi się nawiązanie do interfejsu sterowania (*control interface*) oraz interfejsu statusowego (*feedback interface*). Jest to obszar pamięci modułu dostępny dla użytkownika. Szczegółowy opis parametrów tej przestrzeni pamięci można odszukać w dokumentacji technicznej dostępnej na stronach wsparcia technicznego:

- ET 200SP, TM TIMER DIDQ 10X24V:
<https://support.industry.siemens.com/cs/us/en/view/95153951>
- S7-1500/ET 200MP, TM TIMER DIDQ 16X24V:
<https://support.industry.siemens.com/cs/us/en/view/95153313>

22.1 Detekcja sygnałów

Zgodnie z wcześniejszym opisem – wejścia cyfrowe posiadają dedykowane funkcje technologiczne, takie jak – możliwość dokładnego określenia stempla czasowego zdarzenia, zwiększenie rozdzielczości odczytu stanów w zakresie cyklu aplikacji czy też funkcje licznikowe. Dla każdego z kanałów definiujemy wybraną funkcję na etapie konfiguracji sprzętowej zgodnie z poniższym zrzutem ekranu.



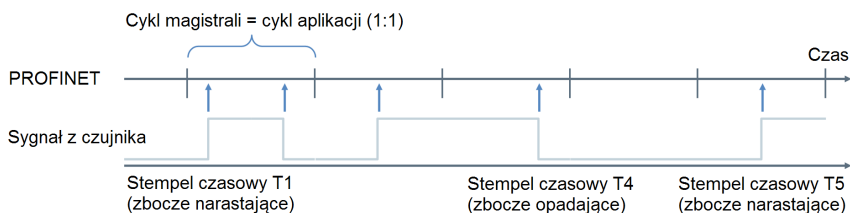
Rejestracja stempla czasowego (Timer DI)

Wybierając opcję *Timer DI* – przypisujemy wejściu cyfrowemu funkcję rejestracji zmiany jego stanu wraz z jednoczesnym zapamiętaniem stempla czasowego takiego zdarzenia. Stempel czasowy określony jest na podstawie tzw. bazy czasu. Jest to wspólny czas synchroniczny dla wszystkich zdarzeń czasowych, które mogą również pochodzić z wielu modułów sprzętowych. Czas ten jest parametrem relatywnym, którego zliczanie rozpoczyna się w momencie przejścia CPU w tryb RUN. Przechowywanie tego wyróżnika w zmiennej typu *LTime* pozwala na wygenerowanie ciągłego przyrostu wartości (wyrażanej w nanosekundach) do ponad 290 lat.

Procesor modułu pozwala na detekcję zarówno zbocza narastającego jak i opadającego wejścia cyfrowego. Ograniczeniem jest fakt, iż na jeden cykl aplikacji, układ może zapisać parametry maksymalnie dwóch zdarzeń dla każdego z kanałów. Stemple czasowe przypisane zostają zgodnie z wartością czasu bazowego. Ich wartość może posłużyć, np. w celu obliczenia odstępu czasu pomiędzy zdarzeniami lub w celu wygenerowania reakcji po określonym czasie.

Funkcjonalność taka dostępna jest jedynie w module typu *time-based*, wymaga również pracy w trybie izochronicznym.

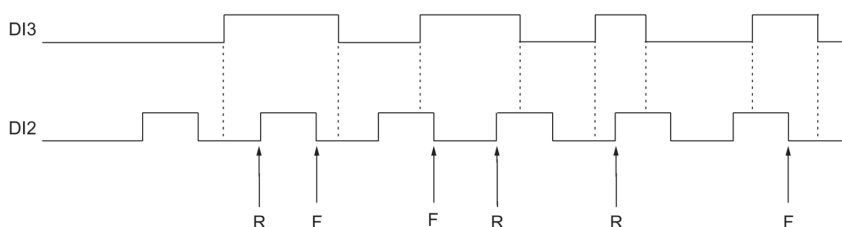
Poniższy przebieg czasowy przedstawia rejestrację stempli czasowych zdarzeń wejściowych w ramach pojedynczego cyklu aplikacji (w tym przypadku równemu cyklowi magistrali). Maksymalnie system jest w stanie zarejestrować dwa zdarzenia na cykl aplikacji (czyli np. w ciągu 1 ms).



Wejście aktywujące (enable input)

Sprzętowe sygnały wejściowe można skonfigurować w takiej zależności, aby jeden z nich (przykładowo DI3) stanowił sprzętowe okno czasowe aktywujące funkcję pomiarową wejścia DI2. Istnieje możliwość nadpisania zezwolenia sprzętowego – za pośrednictwem interfejsu sterowania modułu technologicznego.

Poniższy wykres prezentuje przykładowy przebieg rejestracji stempli czasowych zdarzeń wejścia DI2, w obszarze okna pomiarowego aktywowanego poziomem wysokim sygnału DI3.

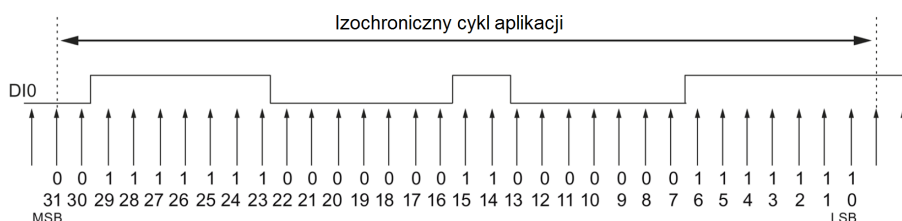


- R Stempel czasowy zbocza narastającego sygnału DI2
 F Stempel czasowy zbocza opadającego sygnału DI2

Zwiększenie rozdzielczości odczytu stanów wejść (oversampling)

Funkcja nadpróbkowania (*oversampling*) służy do realizacji zadania zarejestrowania 32 stanów wejścia cyfrowego podczas jednego cyklu aplikacji (cykl synchroniczny, np. OB61 lub OB91). 32 próbki – odczytane w równych odstępach czasu – zapisywane są na koniec każdego cyklu w 4 bajtach interfejsu statusowego (*feedback interface*) modułu technologicznego.

Nadpróbkowanie wymaga pracy w trybie izochronicznym. Poniższa grafika prezentuje oversampling wejścia DI0.



Zastosowanie powyższego mechanizmu pozwala na zwiększenie dokładności odczytu oraz reakcji na zdarzenia w aplikacjach wymagających bardzo krótkich cykli pracy.

Funkcje licznikowe (Counter)

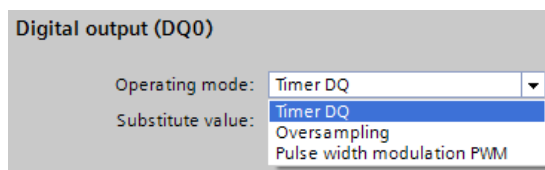
Zintegrowane liczniki rejestrują oraz dodają wykryte zdarzenia. Moduł technologiczny pozwala na skonfigurowanie trzech wejść licznikowych, do których możemy podłączyć np. enkoder inkrementalny lub impulsowy. Nawiązując do rozdziału *Układ pomiarowy* – mamy tutaj możliwość ewaluacji kwadraturowej dla enkoderów z przesuniętymi fazowo sygnałami A oraz B. W przypadku enkodera impulsowego analizowane są tylko zbocza narastające lub tylko opadające.

Liczniki rozpoczynają naliczanie od wartości 0, do $2^{32}-1$, po osiągnięciu wartości skrajnej następuje przepełnienie (*overflow*) oraz powrót do wartości 0. W przypadku połączenia enkodera inkrementalnego, mamy również możliwość zliczania w dół. Wartość licznika zapisywana jest w interfejsie statusowym (*feedback interface*) modułu technologicznego jako wartość 32-bitowa. Opis interfejsu statusowego oraz sterującego można odszukać w plikach pomocy środowiska TIA Portal lub w dokumentacji modułu *TM Timer*.

22.2 Wysterowanie wyjść

Podobnie jak w przypadku kanałów wejściowych, również i zintegrowane wyjścia binarne posiadają dedykowane funkcje technologiczne. Podstawową funkcją jest możliwość precyzyjnego wysterowania wyjścia w czasie. Analogicznie do modułu wejściowego dostępna jest również funkcja oversampling. Dodatkowo zintegrowana została funkcja modulacji długości impulsu PWM.

Wybraną funkcję określamy dla wybranego kanału w ustawieniach sprzętowych modułu technologicznego.



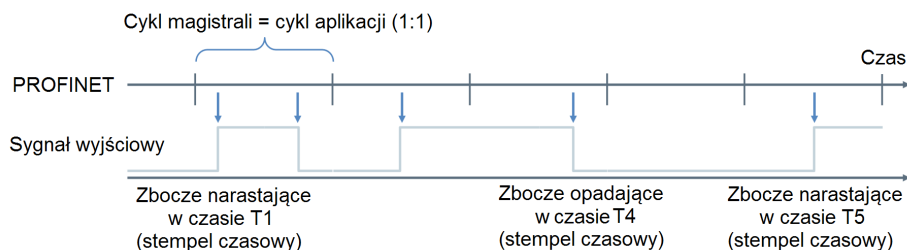
Sterowanie czasowe (Timer DQ)

Zastosowanie stempli czasowych pozwala na zbudowanie systemu, wykonującego powtarzalne operacje w ustalonym czasie. Wyjścia czasowe modułu technologicznego mogą zatem zostać wyzwolone bądź zgaszone, w bardzo precyzyjnie określonym punkcie w czasie. Przykładowo, możemy zdefiniować dokładnie czas reakcji wyjścia na pozyskany sygnał wejściowy.

Funkcje czasowe odnoszą się do wcześniej opisanej bazy czasu. Reakcje czasowe wymagają pracy układu w trybie izochronicznym.

Podobnie jak w przypadku odczytu sygnałów wejściowych – wysterowanie wyjść, jest również ograniczone systemowo do dwukrotnej zmiany stanu poszczególnych kanałów wyjściowych na jeden cykl aplikacji.

Poniższy przebieg czasowy przedstawia generowanie zdarzeń wyjściowych w ramach pojedynczego cyklu aplikacji (w tym przypadku równemu cyklowi magistrali), czyli np. w przeciągu 1 ms.

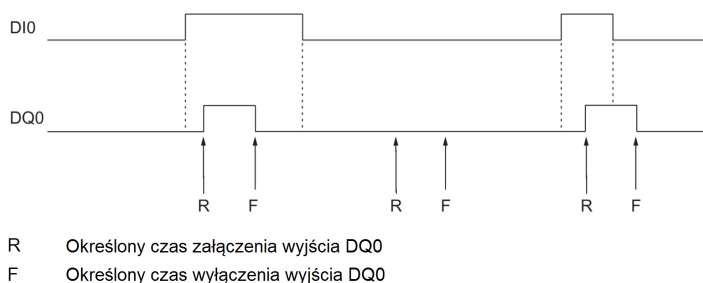


Wejście aktywujące (enable input)

Tak jak w przypadku wejść binarnych – sygnały wyjściowe mogą również zostać powiązane z bramą aktywującą sterowanie. Sprzętowy sygnał wejściowy można skonfigurować w takiej zależności, aby stanowił sprzętowe okno czasowe aktywujące funkcję ustawiania wyjścia. Kasowanie wyjścia jest zawsze niezależne od sygnału aktywującego.

Istnieje możliwość nadpisania zezwolenia sprzętowego – za pośrednictwem interfejsu sterowania modułu technologicznego.

Poniższy wykres prezentuje przykładowy przebieg sterowania wyjściem DQ0 w obszarze okna pomiarowego aktywowanego poziomem wysokim sygnału DI0.



Modulacja szerokości impulsu (pulse width modulation)

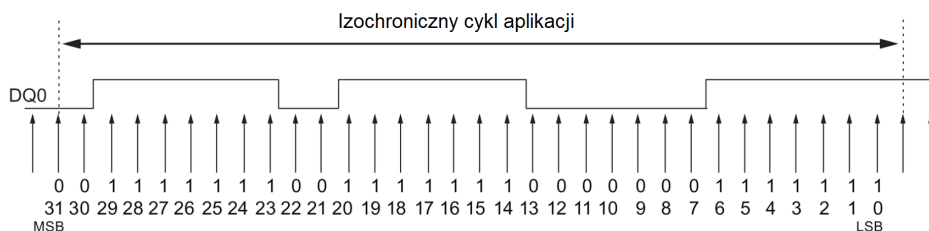
Wyjścia cyfrowe modułu *TM Timer* posiadają również zintegrowaną funkcję pozwalającą na sprzętową konfigurację stosunku impuls-przerwa w cyklu aplikacji czyli funkcji PWM. Konfiguracja wykonywana jest przez interfejs sterowania (*control interface*) wybranego kanału wyjściowego. Wartość zadaną podajemy jako parametr procentowy, który ewaluowany jest z dokładnością zbliżoną do 3%.

Zwiększenie dokładności sterowania (oversampling)

Funkcja nadpróbkowania wyjścia cyfrowego działa w sposób odwrotny do uprzednio opisanego jej zastosowania dla kanałów wejściowych. W tym wypadku, obraz stanów sygnału wyjściowego (na następny cykl) przesyłany jest do modułu technologicznego przed jego rozpoczęciem. Dzięki temu możemy zaprogramować nawet 32 zmiany stanu wyjścia cyfrowego w przeciągu jednego cyklu aplikacji (np. OB61 lub OB91).

Ciąg bitów dla danego kanału wyjściowego przesyłany jest do modułu technologicznego przez interfejs sterowania.

Nadpróbkowanie wyjść wymaga pracy w trybie izochronicznym. Poniższa grafika prezentuje *oversampling* wyjścia DQ0.



UWAGA!

W przypadku wyjść cyfrowych należy upewnić się, iż kombinacja cyklu aplikacji oraz 32-bitowej sekwencji nadpróbkowania, nie przekroczy maksymalnej częstotliwości przełączania kanału wyjściowego, która w przypadku modułu *TM Timer* wynosi 10kHz.

22.3 Przykładowe aplikacje

Poznaliśmy już teoretyczny zakres funkcji jakie realizuje moduł typu *time-based*. Zastanówmy się teraz w przypadku jakiego typu aplikacji opisana technologia może okazać się przydatna.

Zdefiniowany czas odpowiedzi układu

Dzięki funkcjom pomiarowym o charakterystyce czasowej, możemy bardzo precyzyjnie określić czas odpowiedzi systemu, na zdarzenie, które zostało zarejestrowane razem ze stemplem czasowym jego wystąpienia.

Moduł *TM Timer* przetwarza otrzymane instrukcje autonomicznie, a co za tym idzie możemy wysterować wyjście z określonym opóźnieniem – niezależnym od cyklu aplikacji (poza opóźnieniem jednego cyklu wynikającym z konieczności przesłania zadania logicznego do systemu nadrzędnego). W bardzo łatwy sposób możemy więc dodać wymagany czas do stempla czasowego zarejestrowanego przy zmianie stanu sygnału wejściowego. Wykorzystanie tak precyzyjnego pomiaru czasu znajduje zastosowanie, np. w aplikacjach typu *print mark correction*, nóż obrotowy czy odrzucanie elementów.

Pomiar długości

W aplikacjach, gdzie mamy do czynienia z szybko przemieszczającym się elementem – dokładne wskazanie stempla czasowego dwóch zmian stanu wejścia cyfrowego pozwalają wykonać różnicę tych pomiarów. Dzięki temu (oraz znajomości prędkości lub pozycji przemieszczanego obiektu) możemy bardzo dokładnie określić długość detalu.

Wyjścia krzywkowe

Przebieg ruchu osi jest prognozowany na podstawie jej aktualnej pozycji (np. informacja z modułu licznikowego lub obiektu technologicznego). Bazując na wyniku aproksymacji, system potrafi dokładnie określić czas oraz pozycję występowania wyjścia (tzw. wyjście krzywkowe) i przekazać ją do modułu *time-based*. Dzięki temu, załączenie kanału wyjściowego odbywa się dokładnie na wyznaczonej pozycji.

Dozowanie

Wiemy już, że dzięki technologii modułu *TM Timer* możemy precyzyjnie zdefiniować czas załączenia oraz wyłączenia wyjścia cyfrowego. Funkcjonalność ta, może okazać się niezbędna w aplikacjach sterowania zaworem dozującym. Załączenie wyjścia na ściśle określony czas, pozwoli na precyzyjne odmierzenie ilości cieczy lub innej substancji.

Graficzną prezentację wybranych aplikacji można prześledzić na stronie internetowej www.siemens.pl/S7-1500T.

23 Programowanie w TIA Portal

Przejdźmy teraz do bardziej praktycznego podejścia do technologii, którą omówiliśmy we wstępie teoretycznym. Mamy już wiedzę, jakiego typu funkcjonalność posiadają moduły technologiczne oraz w jakich aplikacjach, te zaawansowane rozwiązania, będą przydatne. Sprawdźmy teraz w jaki sposób realizuje się wybrane mechanizmy od strony programistycznej.

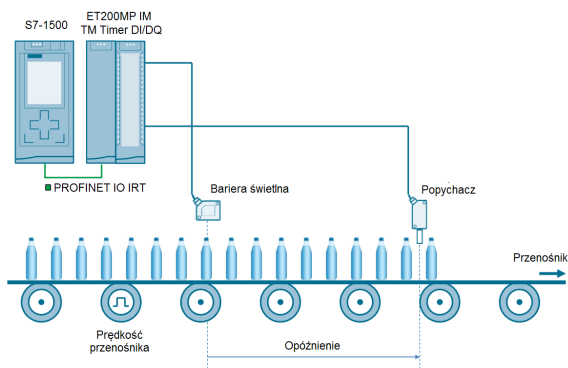
Ogólnie ujmując – tematy odczytu sygnałów wejściowych orazysterowania wyjść w szybkich i precyzyjnych aplikacjach układów serwonapędowych, kryją się pod pojęciami wejść pomiarowych (*Measuring input*) oraz wyjść krzywkowych (*Output cam*). Zagadnienia te odnoszą się do przetwarzania sygnałów binarnych. Krzywka, w tym znaczeniu, ma podobną funkcjonalność do omawianego w poprzednim rozdziale sprzężenia krzywkowego. W tym przypadku odnosi się jednak, do ściśle określonej reakcji wyjścia cyfrowego względem odczytanej pozycji osi technologicznej.

23.1 Konfiguracja manualna

Idąc za wstępem teoretycznym – wiemy już, że powinniśmy zrealizować zadania modułu technologicznego w synchronicznym bloku organizacyjnym jednostki centralnej. Pierwsze z podejść, np. do zagadnienia czasowej reakcji układu, możemy zrealizować przez własny program użytkownika zawarty w synchronicznym bloku organizacyjnym *Synchronous cycle* (np. OB61). Takie podejście do programowej realizacji zadań technologicznych, możemy określić mianem rozwiązania użytkownika, gdyż sami musimy napisać program, który obsłuży w odpowiedni sposób sygnały wejściowe i wyjściowe. Zostanie wykonane to przez dedykowane w tym celu systemowe funkcje programowe.

Funkcje czasowe

W celu zobrazowania funkcjonalności modułu typu *time-based* – rozważmy przykładową aplikację. Wyobraźmy sobie transporter z przemieszczającymi się produktami (np. butelki), czujnikiem (np. świetlnym) wadliwych produktów oraz popychaczem wykonującym zadanie zrzucenia z linii zbędnego artykułu. Przykładowy układ zilustrowano poniższą grafiką.



Zadanie sprowadza się więc do precyzyjnego odczytu sygnału z czujnika optycznego oraz wystawienia wyjścia binarnego po zdefiniowanym czasie. Znając prędkość przenośnika możemy dynamicznie dostosować parametry układu podczas jego pracy. Możemy również odnosić się bezpośrednio do pozycji przenośnika (enkoder).

Moduł technologiczny zgodnie z wymaganiami systemowymi został połączony do jednostki centralnej S7-1500 przez izochroniczną sieć PROFINET IO IRT. Rozwiązanie nie uwzględniające wykorzystania obiektów technologicznych można zastosować przy wykorzystaniu standardowego sterownika z rodziny S7-1500.

Rozwiązanie problemu w tym przypadku sprowadza się więc do kilku czynności:

- wykorzystanie synchronicznego bloku organizacyjnego (*Synchronous cycle*),
- synchronizacji czasu dla wybranych modułów – funkcja *TIO_Sync*,
- obsługa wejść/wyjść czasowych przez funkcje *TIO_DI* oraz *TIO_DQ*,
- wygenerowanie programowego opóźnienia wysterowania wyjścia (popychacz) względem odczytu wejścia (czujnik).

W niniejszym dokumencie nie będziemy zajmować się opisem konfiguracji takiego podejścia do opisanego zagadnienia, ze względu na to, że szczegółowy opis poszczególnych kroków konfiguracyjnych oraz przykładowy projekt utworzony w środowisku TIA Portal znajduje się w następującej lokalizacji sieciowej:

<https://support.industry.siemens.com/cs/ww/en/view/109738186>

Wracając do opisu cyklu synchronicznego programu, magistrali oraz aplikacji – trzeba zwrócić uwagę, iż zawsze będziemy mieć drobne opóźnienie w reakcji układu, wynikające z cyklu aplikacji oraz komunikacji z jednostką centralną. Nie ma możliwości aby w zakresie jednego cyklu aplikacji zaprogramować reakcję na wykryte zdarzenie – wymaga to przetworzenia programu w jednostce centralnej. Jakiego rzędu są to opóźnienia, przeanalizujemy podczas konfiguracji wejścia pomiarowego.

Ekstrapolacja pozycji

Powyższy przykład prezentuje przykładowe rozwiązanie dla zagadnienia obsługi wejść/wyjść czasowych. Jesteśmy w stanie tutaj uzyskać bardzo wysoką dokładność czasu reakcji. W wielu szybkich aplikacjach nie będzie nas jednak interesować czas, a pozycja osi w danym momencie. Niezależnie skąd wartość pozycji trafi do systemu (oś technologiczna, szybki licznik, enkoder zewnętrzny, etc), skonfrontowanie jej z pozyskanym stemplem czasowym będzie możliwe dopiero w jednostce centralnej.

Założmy, że odczytamy stempel czasowy zdarzenia w środku cyklu aplikacji. Na koniec jego trwania informacja zostanie wysłana do jednostki centralnej. Tam – przez program użytkownika – należy, na podstawie przebiegu pozycji osi w poprzednich cyklach, oszacować jaka będzie pozycja w danym punkcie zakończonego cyklu. Ekstrapolację taką (czyli po prostu aproksymację) należy realizować przez algorytm cyklicznego programu synchronicznego użytkownika. Po jego wykonaniu, będziemy w posiadaniu wartości pozycji, co pozwoli na reakcję w kolejnym cyklu synchronicznym.

Uzupełniając przekrój możliwości systemowych – w dalszej części zajmiemy się wnikliwą analizą podejścia zautomatyzowanego przy wykorzystaniu obiektów technologicznych.

23.2 Wejście pomiarowe – obiekt technologiczny

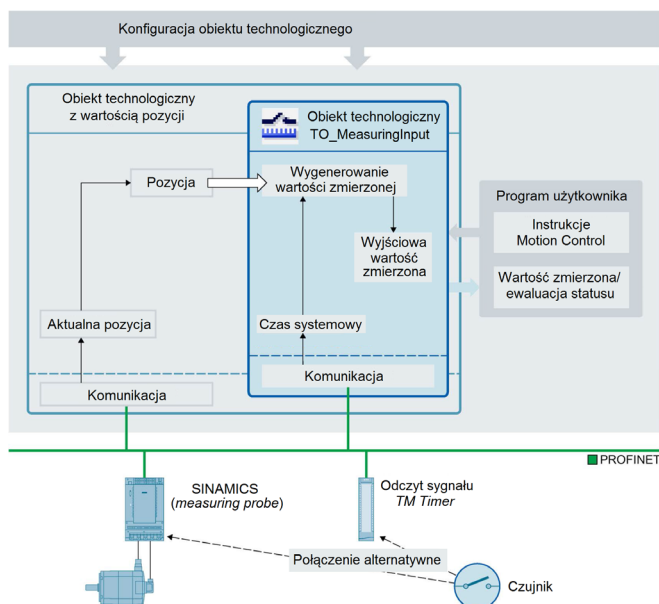
Alternatywnym podejściem do obsługi wejść/wyjść czasowych jest wykorzystanie obiektów technologicznych. W dalszej części poznamy dwa nowe obiekty, służące do obsługi wejść pomiarowych oraz wyjść krzywkowych. Rozwiązanie problemu w taki sposób możliwe jest jedynie w sterownikach S7-1500 z firmware v2.0 lub nowszym.

Obiekty technologiczne niebywale usprawniają konfigurację w/w funkcji. Niemalże całość zagadnienia opisanego w powyższej sekcji realizowana jest systemowo, praktycznie bez konieczności pisania programu użytkownika. Analizę zagadnienia zaczniemy od omówienia pierwszego obiektu technologicznego.

Obiekt technologiczny – wejście pomiarowe

Obiekt technologiczny *TO_MeasuringInput* pozyskuje pozycję aktualną osi lub enkodera zewnętrznego w momencie wykrycia zmiany stanu wejścia cyfrowego. Jest to bardzo precyzyjny mechanizm odczytu wejścia czasowego, który wymaga zastosowania dedykowanego sprzętu (np. modułu technologicznego *TM Timer*).

Poniższa grafika ilustruje zasadę działania wejścia pomiarowego obsługiwanego przez obiekt technologiczny.



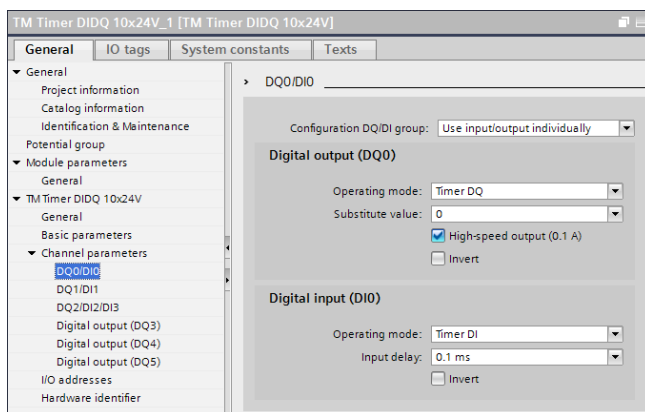
Zgodnie z przedstawionym opisem – moduł *TM Timer*, służy m.in. do precyzyjnego przechwytywania stempla czasowego zdarzeń wejściowych. Nie rejestruje on jednak bezpośrednio pozycji osi. W związku z powyższym, stempel czasowy odczytany przez moduł technologiczny (lub wejście cyfrowe napędu) zostaje przesłany do jednostki centralnej, a dopiero tam wygenerowana zostaje wartość pozycji osi. Dzieje się to, po odczytaniu jej z obiektu technologicznego, który pozycję aktualną rejestruje (oś pozycjonująca, synchroniczna lub enkoder zewnętrzny). Zgodnie ze schematem graficznym, łatwo zauważyć, iż system (poprzez obligatoryjnie zintegrowany obiekt

technologiczny) automatycznie zestawia odczytany stempel czasowy z aktualną pozycją osi. Wartość pozycji zostaje więc ekstrapolowana systemowo, a wyjściowym parametrem obiektu technologicznego jest gotowa pozycja z danego punktu w czasie.

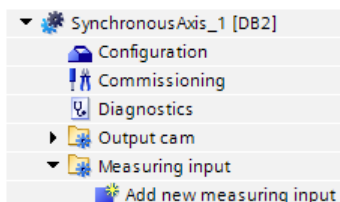
Już na tym etapie łatwo zauważyć, że konfiguracja jest dalece zautomatyzowana. Nie ma tutaj konieczności generowania bloku synchronicznego, gdyż jak wiemy obsługa obiektów technologicznych odbywa się w systemowym bloku *MC-Servo*. W związku z tym nie ma również potrzeby stosowania funkcji do synchronizacji czasu oraz obsługi wejść/wyjść czasowych. A co najważniejsze, nie musimy tworzyć żadnego programu użytkownika w celu obliczenia pozycji osi w momencie zarejestrowania zdarzenia wejściowego, gdyż wartość ta zwracana jest przez obiekt technologiczny *TO_MeasuringInput*.

Konfiguracja w TIA Portal

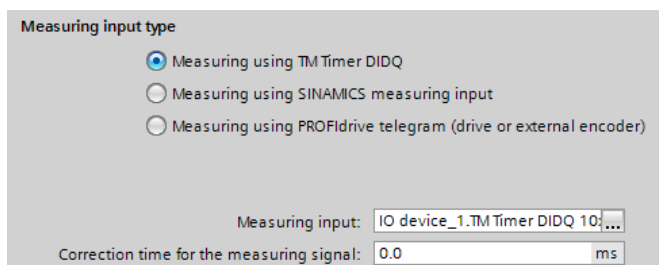
Pierwszym krokiem w konfiguracji jest parametryzacja kanału wejściowego modułu *TM Timer*. Obiekt technologiczny *TO_MeasuringInput* działa na zasadzie wejść czasowych, zgodnie z wymaganiami programowymi wybrany kanał należy ustawić na tę właśnie opcję. W zakładce konfiguracyjnej wybieramy opcję jak na poniższym rzucie ekranu – przykładowo dla kanału DI0.



Następnie dodajemy do projektu obiekt technologiczny *Measuring input*. Można to wykonać podobnie jak w poprzednich przykładach (wtedy w oknie wyboru funkcji technologicznej trzeba będzie bezwarunkowo wskazać oś, do której wejście pomiarowe ma zostać przypisane) lub w zakresie wybranej osi (obiektu technologicznego) wybrać zakładkę *Measuring input* i wybrać opcję *Add new measuring input*.



W ustawieniach wstawionego obiektu technologicznego podstawowym elementem konfiguracyjnym jest wybór wejścia obsługującego czasowe wejścia pomiarowe. Mamy tutaj do wyboru trzy opcje.



Sprzętowym źródłem kanału pomiarowego może być :

- wejście modułu technologicznego *TM Timer (time-based)* – stempel czasowy pozyskany jest po stronie systemu rozproszonych wejść/wyjść, natomiast ewaluacja pozycji wykonywana jest w obiekcie technologicznym jednostki centralnej,
- kanał pomiarowy znajdujący się bezpośrednio na przekształtniku częstotliwości (*time-based*), np. wejścia *measuring probe* napędów SINAMICS S120 (przykład opisano tutaj: <https://support.industry.siemens.com/cs/pl/en/view/99086336>), w tym przypadku podobnie – napęd przesyła jedynie stempel czasowy zdarzenia do obiektu technologicznego,
- pomiar wykonany po stronie napędu lub enkodera zewnętrznego, a następnie przesłany przez ramkę *PROFIdrive* do sterownika. W tym rozwiązaniu pozycja obliczona jest już po stronie napędu lub enkodera. W ramce *PROFIdrive* przesyłana jest do obiektu technologicznego już wartość wyjściowa pozycji aktualnej.

Korekta czasu

W związku z tym, że technologia pomiarów związana jest ściśle z czasem detekcji zdarzeń – w konfiguracji sprzętowej system umożliwia wprowadzenie ręcznej korekty stempla czasowego (*Correction time for the measuring signal*). Oznacza to, że jeśli znamy czynniki, które mogą powodować opóźnienie rejestracji czasu zdarzenia, możemy manualnie wprowadzić w parametrze wejścia pomiarowego tę wartość czasu. Przyczyną może być, np. odchylenie wynikające z konstrukcji mechanicznej, czynników środowiskowych lub opóźnienie doprowadzenia wartości mierzonej przez czujnik do wejścia modułu pomiarowego.

Uwagi

Wejście pomiarowe może być wyzwolone jednorazowo lub cyklicznie. Ta druga opcja dostępna jest tylko przy wykorzystaniu modułu *TM Timer*. W tej wersji wymagana jest praca w trybie izochronicznym. W jednym cyklu aplikacji wykonane mogą być maksymalnie dwa pomiary.

Każde wejście pomiarowe musi być przypisane do jednego obiektu technologicznego zawierającego pozycję aktualną.

Do jednego obiektu technologicznego z pozycją aktualną, może być przypisanych wiele wejść pomiarowych.

Jednostka pomiaru pozycji pobierana jest z ustawień nadrzędnego obiektu technologicznego.

Zwróćmy jeszcze uwagę na kartę ustawień zaawansowanych obiektu technologicznego *Measuring Input (Configuration → Extended parameters)*.

Taking into consideration the current system settings, the result is:

Time from output of a MC_MeasuringInput job until measuring event detection becomes effective:	8.0	ms
Minimum time after measuring event until measured value is available in the controller (measuring of one edge):	10.0	ms
Minimum time after measuring event until measured value is available in the controller (measuring of two edges):	10.0	ms

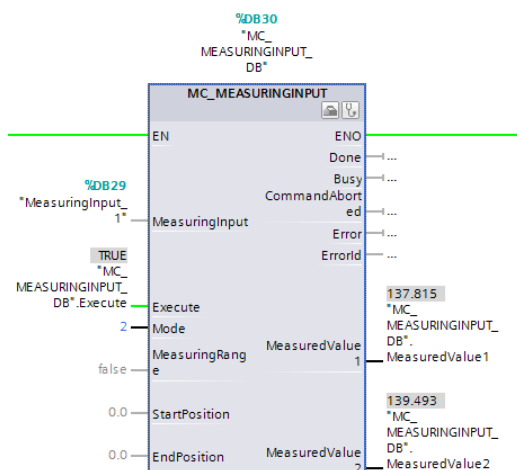
W związku z wcześniejszym opisem, każde zadanie pomiaru oraz reakcja na ekstrapolowaną wartość pozycji poprzedzone jest opóźnieniem wynikającym z cyklu aplikacji oraz komunikacji modułu pomiarowego z jednostką centralną S7-1500. System, na podstawie zdefiniowanego cyklu aplikacji oraz zegara magistrali komunikacyjnej oblicza czas od aktywacji funkcji pomiarowej do wykonania pomiaru oraz minimalny czas potrzebny na przesłanie stempla czasowego (lub dwóch) zarejestrowanego zdarzenia z powrotem do sterownika.

Powyższy przykład wygenerowany jest dla przykładu gdzie cykl aplikacji = 4ms, cykl zegara magistrali = 2ms. Na wykonanie pomiaru system potrzebuje dwóch cykli izochronicznych aplikacji. Przesłanie pomiaru oraz jego wpisanie do obiektu technologicznego wymaga czasu jednego cyklu magistrali oraz dwóch cykli aplikacji czyli w sumie 10ms.

Interfejs użytkownika – funkcja MC_MeasuringInput

Ostatnim etapem konfiguracji jest wstawienie do programu użytkownika funkcji *Motion Control* do aktywacji wejścia pomiarowego. Blok funkcyjny znajduje się na liście instrukcji w zakładce *Technology → Motion Control → Measuring input, output cam, cam track*. Do wyboru mamy dwie funkcje: *MC_MeasuringInput* (aktywacja jednorazowego pomiaru) oraz *MC_MeasuringInputCyclic* (pomiar cykliczny). Obie funkcje parametryzujemy w analogiczny sposób.

Pod parametr *MeasuringInput* podłączamy nasz obiekt technologiczny. Aktywacja pomiaru odbywa się przez podanie zbocza narastającego na wejście *Execute*.



Kolejnym parametrem jest wybór trybu pracy wejścia pomiarowego (*Mode*). Wracając do danych technicznych modułu *TM Timer* oraz możliwości programowych systemu – maksymalnie możemy zarejestrować dwa zdarzenia (stemple czasowe) w ciągu jednego cyklu aplikacji. Poprzez parametr *Mode* będziemy właśnie określać jakiego typu zdarzenia to będą. Naturalnie, zdarzenia te mogą być od siebie oddalone o dowolną wartość czasu, nie muszą być zawarte w jednym cyklu aplikacji

Po aktywacji funkcji w określonym trybie wykonany zostanie jeden z pomiarów widocznych w tabeli poniżej.

Tryb	Charakterystyka pomiaru
0	Następne zbocze narastające.
1	Następne zbocze opadające.
2	Dwa kolejne zbocza.
3	Dwa kolejne zbocza, z których pierwsze musi być narastające: - Zbocze narastające zapisane jest do parametru <i>MeasuredValue1</i> - Zbocze opadające zapisane jest do parametru <i>MeasuredValue2</i>
4	Dwa kolejne zbocza, z których pierwsze musi być opadające: - Zbocze opadające zapisane jest do parametru <i>MeasuredValue1</i> - Zbocze narastające zapisane jest do parametru <i>MeasuredValue2</i>

Kolejne parametry funkcji *MC_MeasuringInput* – *MeasuringRange*, *StartPosition* oraz *EndPosition* pozwalają opcjonalnie określić zakres pozycji osi, w której wejście pomiarowe ma wykonywać rejestrację zdarzeń. Ustawienie wejścia *MeasuringRange* na wartość 0 powoduje wykonywanie pomiaru niezależnie od aktualnej pozycji.

Interfejs wyjściowy funkcji użytkownika, dostarcza informacji na temat statusu wykonywanego zadania oraz ewentualnych błędów. Wartości zmierzone przez cały układ zapisywane są do parametrów *MeasuredValue1* (pierwsza zmierzona wartość) oraz *MeasuredValue2* (druga zmierzona wartość – w trybie 2, 3, lub 4). Zwróćmy uwagę, iż do parametrów *MeasuredValue1* oraz *MeasuredValue2* trafia już ekstrapolowana systemowo pozycja osi dla stempla czasowego zarejestrowanego przez moduł typu *time-based*.

Analizując nieco bardziej szczegółowo technologiczny blok danych wejścia pomiarowego, zauważymy, że system trwale rejestruje zmierzone wartości. Trwale czyli do momentu wykonania kolejnego pomiaru – w przeciwieństwie do interfejsu wyjściowego funkcji, gdzie zerowane są one po wykryciu zbocza opadającego *Execute*. Blok danych zawiera również informacje na temat ilości wykonanych pomiarów, a także o ilości zdarzeń, których nie był w stanie zarejestrować (więcej niż dwa zdarzenia w cyklu aplikacji).

Pomiar cykliczny – funkcja *MC_MeasuringInputCyclic*

Jeśli zdefiniowane wejście pomiarowe wykonywać ma rejestrację zdarzeń w każdym cyklu aplikacji, przewidziana została do tego zadania funkcja systemowa. Funkcja *MC_MeasuringInputCyclic* działa analogicznie do funkcji pomiaru jednorazowego, z tym, że możliwa jest praca jedynie w trzech trybach – pomiar zboczy narastających, opadających lub obu tych typów. Interfejs funkcji dostarcza bezpośrednio informacji na temat obliczonej wartości pozycji, a także zliczonej oraz utraconej liczby zdarzeń.

Podsumowanie

Na tym etapie możemy wykonać krótkie porównanie podejścia do realizacji zadań pomiarów czasowych przez obiekt technologiczny, względem rozwiązania manualnego. Zauważmy, iż zastosowanie mechanizmu systemowego praktycznie rozwiązuje całość zagadnienia od strony programowej. W przeciwieństwie do ręcznej obsługi zdarzeń czasowych – nie musimy stosować własnego synchronicznego bloku organizacyjnego, nie ma konieczności zapewnienia synchronizacji czasu, ani wywołania programu obsługującego odczyt kanałów wejściowych. Dodatkowo, biorąc pod uwagę, że nie pracujemy jedynie na parametrach czasowych, a potrzebujemy w programie sterownika również wartość pozycji aktualnej w momencie odczytu zdarzenia – automatyczna ekstrapolacja pozycji wynikająca z zarejestrowanego stempla czasowego jest znacznym ułatwieniem całej procedury.

Sztandarowym zastosowaniem wejścia pomiarowego z uwzględnieniem wartości pozycji jest aplikacja tzw. *print mark correction*. Nazwa pochodzi od technologii stosowanej do detekcji punktu wykonania zadania (np. odcięcie kawałka elementu). Dzięki nadrukowi (*print mark*) na materiale, system w każdym cyklu wykonuje korektę pozycji cięcia. Powoduje to, że nacięcie jest w każdym cyklu możliwie precyzyjne. Rozwiązanie takie, wykorzystywane jest zwłaszcza w przypadku materiałów, które mogą ulec rozciągnięciu, przesunięciu

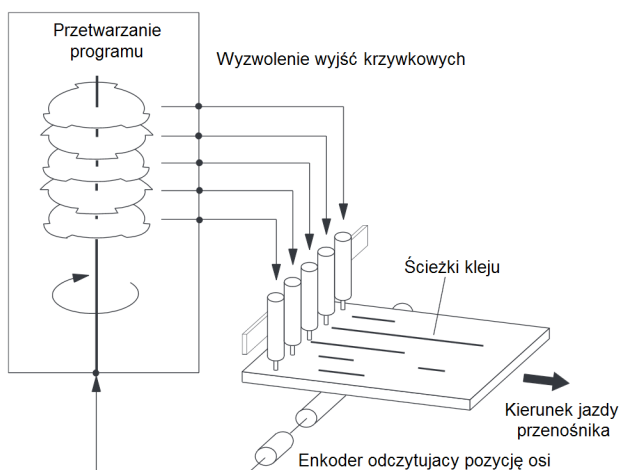
bądź deformacji podczas cyklu produkcyjnego. Przykładem może być zawijanie produktów spożywczych w opakowania foliowe (*flowpack*).

Krótką animację prezentującą zagadnienie można obejrzeć w następującej lokalizacji sieciowej: <http://www.siemens.pl/S7-1500T>

23.3 Wyjście krzywkowe – obiekt technologiczny

Można powiedzieć, że mechanizmem odwrotnym do zagadnienia wejścia pomiarowego – jest wyjście krzywkowe. W tym wypadku nie chodzi nam o ustalenie pozycji w momencie zdarzenia, a o wysterowanie sygnału binarnego przy określonej pozycji osi lub enkodera zewnętrznego. Sygnał krzywkowy może być ewaluowany w programie użytkownika lub wystawiony bezpośrednio na sprzętowe wyjście cyfrowe.

W celu przedstawienia funkcjonalności oraz zastosowania wyjść krzywkowych rozważmy przykładową aplikację – precyzyjne nakładanie ścieżek kleju na sklepany materiał. Wyobraźmy sobie przenośnik, na którym umieszczony zostaje materiał, na którego powierzchnię mamy bardzo dokładnie nakładać klej. Przenośnik będzie w tym wypadku naszą osią, której pozycję będziemy w programie analizować. Ścieżki kleju umieszczane będą przez obsługę szybkimi wyjściami zaworów dozujących. Poniższa ilustracja obrazuje rozważany przykład.



Aplikacje tego typu zazwyczaj wymagają wysokiej dokładności sterowania wyjściami cyfrowymi względem odczytanej pozycji przenośnika.

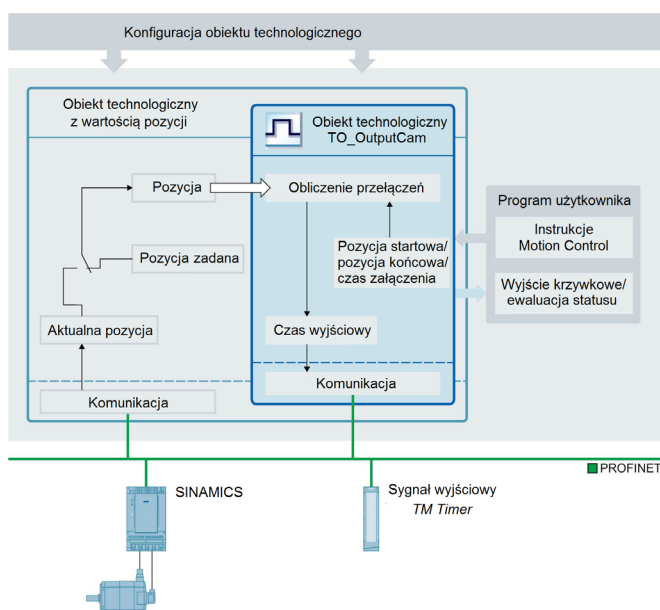
W zależności od potrzeb aplikacji, oczekiwanej dokładności oraz prędkości poszczególnych zadań – możemy zastosować odpowiednią konfigurację sprzętową takiego układu. Wykorzystane mogą zostać zarówno standardowe wyjścia sprzętowe jak i omówione w niniejszym rozdziale precyzyjne wyjścia czasowe.

Od strony programowej możemy wyobrazić sobie – analogicznie jak w przypadku wejścia pomiarowego – manualną obsługę takiego mechanizmu lub wykorzystanie obiektów technologicznych sterownika S7-1500. Jako parametr referencyjny przyjąć można zarówno pozycję jak i czasysterowania wyjścia. Wiemy już, jak poszczególne podejścia mogą pomóc lub utrudnić tworzenie algorytmu sterowania.

Na potrzeby poznawcze, założmy, że aplikacja jest bardzo wymagająca – szybka oraz precyzyjna. W związku z powyższym będziemy chcieli zastosować wyjścia czasowe modułu *TM Timer*. W celu wskazania możliwie prostej drogi konfiguracyjnej – zastosujemy dedykowane dla takich rozwiązań obiekty technologiczne.

Obiekt technologiczny – wyjście krzywkowe

Wyjście krzywkowe jest pewnego rodzaju odniesieniem od sprzężeń krzywkowych, o których dywagowaliśmy w rozdziale poprzednim. W odniesieniu do wyjść krzywkowych, poruszamy się jednak w obszarze sterowań binarnych. Dla każdej zdefiniowanej pozycji osi wiodącej, poprzez konfigurację obiektu technologicznego *TO_OutputCam*, będziemy określać stan wyjścia cyfrowego. Algorytm sterowania zaprezentowany został na poniższym schemacie.



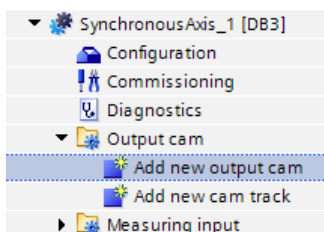
System odczytuje pozycję osi lub enkodera zewnętrznego. Alternatywnie możemy jako wartość źródłową pozycji przyjąć pozycję zadaną osi (obliczoną przez interpolator). Pozycja przekazywana jest do obiektu technologicznego wyjścia krzywkowego, gdzie obliczone zostaną czasy przełączenia wyjścia binarnego. Stan wyjścia można zdefiniować przez pozycję załączenia oraz wyłączenia lub przez pozycję załączenia z uwzględnieniem czasu jego trwania. Stan wyjścia krzywkowego, wyliczony przez obiekt technologiczny, może zostać przekazany do programu użytkownika lub na wyjście sprzętowe. W tym drugim przypadku może być to standardowy kanał wyjścia cyfrowego lub wyjście czasowe modułu technologicznego *TM Timer*.

Standardowe wyjścia cyfrowe uzależnione są cyklem pracy aplikacji oraz możliwościami sprzętowymi systemu wejść/wyjść. W przypadku szybkich aplikacji czasowych (*TM Timer*) pozycja załączenia wyjścia cyfrowego jest ekstrapolowana systemowo, podobnie jak w przypadku omówionego wejścia pomiarowego. Oznacza to, że system na podstawie aktualnej pracy układu (prędkość/dynamika) estymuje dokładny czas załączenia wyjścia niezależny od cyklu pracy aplikacji. Moment przełączenia stanu wyjścia, wyliczany jest przez mechanizm systemowy na dwa cykle aplikacji przed jego wykonaniem. Uzyskujemy tutaj powtarzalność sterowania na poziomie mikrosekund.

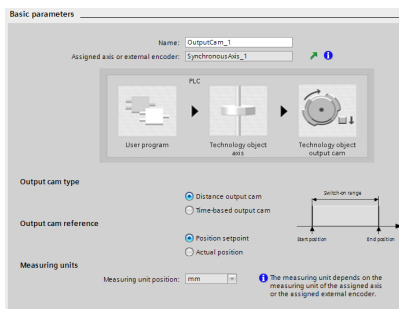
Konfiguracja w TIA Portal

Przy założeniu wykorzystania systemu I/O typu *time-based* jako sprzętowego wyjścia krzywkowego, konfigurację zaczniemy analogicznie jak w przypadku wejścia pomiarowego – od ustawień modułu technologicznego. Wybrany kanał powinien zostać ustawiony jako wyjście czasowe (*Timer DQ*) – jest to wartość domyślna.

Następnie dodajemy do projektu obiekt technologiczny *TO_OutputCam*. Charakterystyka jest podobna do wejścia pomiarowego także dodajemy wyjście krzywkowe jako nowy obiekt (i wskazujemy nadrzędną oś z wartością pozycji) lub aktywujemy wyjście w obszarze istniejącej osi lub enkodera zewnętrznego.



Karta ustawień podstawowych przedstawiona została na poniższym zrzucie ekranu.



Ten panel konfiguracyjny jest bardzo istotny, gdyż to tutaj określimy zasadę działania naszego wyjścia krzywkowego. Przede wszystkim określamy typ wyjścia (*Output cam type*) – może być to wyjście w pełni zależne od pozycji (*Distance output cam* – załączenie oraz wyłączenie wyjścia w konkretnym punkcie przypisanej osi) lub zależne od pozycji załączenie połączone z czasem jego trwania (*Time-based output cam*).

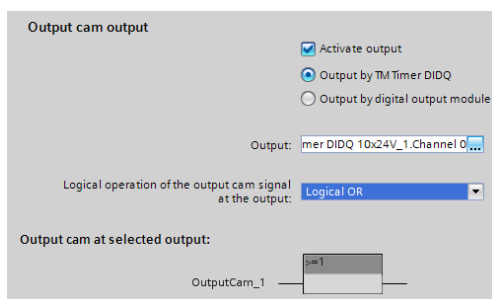
Drugi ważny parametr to źródło wartości pozycji, na podstawie której obliczane będą czasy przełączeń wyjścia krzywkowego. Znamy już doskonale różnicę pomiędzy pozycją zadaną (*Position setpoint*) oraz aktualną (*Actual position*) – wybór możemy wykonać właśnie pomiędzy tymi dwoma wartościami.

Jednostka pozycji pobierana jest z ustawień nadrzędnego obiektu technologicznego.

Kolejnym krokiem jest parametryzacja sprzętowego wyjścia sterownika.

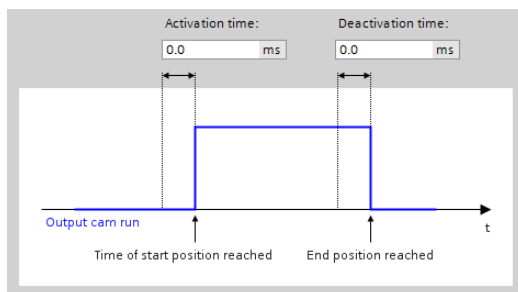
Przed wszystkim – jeśli wyjście ma zostać ustawione elektrycznie – należy aktywować kanał wyjściowy (*Activate output*). W przeciwnym razie wartość krzywki będzie ewaluowana ale tylko w obszarze programowym obiektu technologicznego. Co nie zmienia faktu, że można wartość wyjściową wykorzystać w programie użytkownika.

Aktywacja krzywki pozwoli na wybór kanału wyjściowego. Zgodnie z wcześniejszymi informacjami – fizycznym wyjściem krzywkowym może być standardowa karta I/O (*Output by digital output module*) lub moduł czasowy (*Output by TM Timer DIDQ*).



Zwróćmy uwagę, iż możemy również wykonać operację logiczną (OR/AND) na aktualnej wartości konfigurowanej krzywki. Funkcja ta potrzebna jest w sytuacji gdy do jednego kanału sprzętowego przypisanych jest więcej niż jedno wyjście krzywkowe (jest to dopuszczalne). W takim przypadku, wynik skonfigurowanej logiki uzależniać będzie wystawienie wyjścia fizycznego.

Kończąc konfigurację obiektu technologicznego, możemy określić bardziej szczegółowe zasady pracy wyjścia krzywkowego (*Extended parameters*). Mamy sposobność dostosowania momentu załączenia oraz wyłączenia wyjścia, przed osiągnięciem pozycji startowej/końcowej:



oraz zdefiniowania histerezy dla wahań pozycji nadrzędnego obiektu technologicznego, które nie powinny być interpretowane jako zmiana kierunku pracy osi (pozycja aktualna).

Uwagi

Praca w trybie czasowym (*TM Timer*) wymaga trybu izochronicznego. Tryb IRT obsługiwany jest jedynie przez systemy wejść/wyjść PROFINET – ET 200SP/MP.

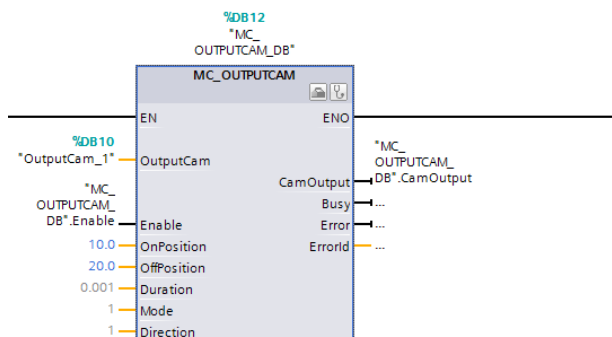
Wyjście krzywkowe musi być zawsze powiązane z nadrzędnym obiektem technologicznym, którego pozycja będzie ewaluowana. Źródłem pozycji może być oś pozycjonująca, synchroniczna lub enkoder zewnętrzny.

Jedno wyjście krzywkowe może mieć tylko jedno źródło pozycji.

Oś lub enkoder zewnętrzny może mieć przypisanych wiele różnych wyjść krzywkowych.

Interfejs użytkownika – funkcja MC_OutputCam

W celu aktywacji i parametryzacji skonfigurowanego wyjścia krzywkowego – pozostaje nam wywołanie odpowiedniej funkcji programowej. Dedykowaną temu zadaniu funkcję *MC_OutputCam*, znajdziemy na liście instrukcji w zakładce *Technology* → *Motion Control* → *Measuring input, output cam, cam track*.



Podstawowym parametrem jest oczywiście nasz obiekt technologiczny (podłączamy pod parametr *OutputCam*). Wyjście krzywkowe aktywowane jest stanem wysokim, stale załączonym na wejściu *Enable*. *OnPosition* oraz *OffPosition* to pozycja załączenia i wyłączenia wyjścia krzywkowego (dla konfiguracji *Distance output cam*) w jednostce wynikającej z konfiguracji nadrzędnego obiektu technologicznego. Konfiguracja czasowa (*Time-based output cam*) wymaga podania pozycji załączenia oraz czasu trwania (w milisekundach) stanu wysokiego wyjścia (*Duration*).

Funkcja interfejsu użytkownika pozwala również na wybór trybu pracy wyjścia krzywkowego (*Mode*). Parametr ten pozwoli wywołać przebieg pracy wyjścia w trybie standardowym (*Mode = 1*) lub zanegowanym (*Mode = 2*), ewentualnie stale aktywnym (*Mode = 3*).

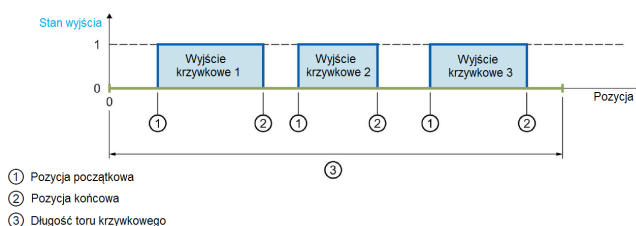
Kierunek (*Direction*) determinuje czy wyjście krzywkowe ma być aktywowane jedynie podczas przyrostu (lub spadku) wartości pozycji czy też niezależnie od kierunku pracy osi.

Parametr *CamOutput* odzwierciedla aktualny stan zdefiniowanego wyjścia krzywkowego. Pozostałe wyjścia funkcji *MC_OutputCam* dostarczają informacji na temat aktualnego stanu pracy funkcji systemowej oraz ewentualnych błędów.

23.4 Tor krzywkowy – Cam Track

Ostatnie zagadnienie, o którym warto wspomnieć w tej części dokumentacji – jest jeszcze jeden nowy obiekt technologiczny – *TO_CamTrack*. Ten element stanowi połączenie wielu krzywek, o których rozmawialiśmy powyżej. Tor krzywkowy służy do wygenerowania sekwencji przełączeń wyjścia binarnego – zależnie od pozycji osi lub enkodera zewnętrznego. W odróżnieniu do standardowego wyjścia krzywkowego – w tym przypadku generować będziemy nie jeden, a wiele przedziałów, w których pojedyncze wyjście krzywkowe będzie załączane lub wyłączane.

Tor krzywkowy może zawierać do 32 pojedynczych wyjść krzywkowych, które mogą być aktywowane w obszarze programu użytkownika lub na fizycznym kanale sprzętowym – dokładnie w ten sam sposób jak opisaliśmy w poprzedniej sekcji. Przykład działania toru krzywkowego opartego o pozycję załączenia widnieje na poniższej ilustracji.

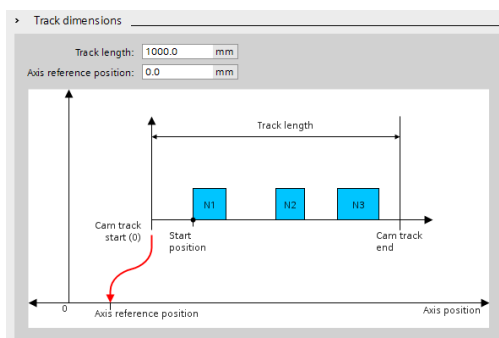


Konfiguracja w TIA Portal

Ścieżka krzywkowa jest niezależnym obiektem technologicznym (*TO_CamTrack*), który dodajemy do projektu analogicznie jak uprzednio opisane wyjście krzywkowe. Konfiguracja również sprowadza się do zdefiniowania podobnych parametrów, takich jak typ wyjścia (pozycja/czas), źródło pozycji referencyjnej czy fizyczny kanał wyjściowy.

Różnice w budowie obiektu technologicznego sprowadzają się do dwóch zagadnień – długości ścieżki oraz jej przebiegu.

Długość toru krzywkowego (*Extended parameters* → *Track dimensions* → *Track length*) określa całkowity zakres pozycji osi (lub enkodera zewnętrznego), w którym zdefiniowany przebieg stanu wyjścia krzywkowego będzie realizowany. Wartości wykraczające poza skonfigurowany przedział nie zostaną przez system uwzględnione.



Zwróćmy uwagę, iż istnieje również możliwość zdefiniowania offsetu dla osi dostarczającej wartość pozycji (*Axis reference position*). Wprowadzona wartość będzie traktowana jako początkowy punkt odniesienia do uruchomienia toru krzywkowego.

Przebieg wyjścia krzywkowego (N1, N2, N3, ...) określamy w ostatniej zakładce panelu konfiguracyjnego obiektu technologicznego (*Extended parameters* → *Output cam data*).

Output cam data

Set output cam type: distance output cam

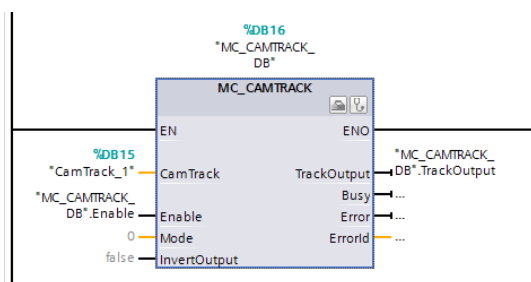
Output cam	Valid	Start position	End position
1	☒	0.0 mm	20.0 mm
2	☒	39.0 mm	88.0 mm
3	☒	200.0 mm	500.0 mm
4	☐	0.0 mm	0.0 mm
5	☐	0.0 mm	0.0 mm

Punkty załączeń oraz wyłączeń (maksymalnie 32 krzywki) definiujemy przez wprowadzenie pozycji startowej oraz końcowej każdego z przedziałów. W związku z tym, że definicja punktów kluczowych umieszczona została na poziomie obiektu technologicznego, interfejs użytkownika wymagać będzie jedynie aktywacji gotowego toru krzywkowego.

Aby wprowadzać zmiany do charakterystyki toru krzywkowego z poziomu programu użytkownika – wymagana jest modyfikacja odpowiednich rekordów technologicznego bloku danych. Nie stanowi to żadnego problemu z technicznego punktu widzenia.

Interfejs użytkownika – funkcja MC_CamTrack

Funkcja programowa aktywująca skonfigurowany tor krzywkowy znajduje się w zakładce funkcji *Motion Control* (*Technology* → *Motion Control* → *Measuring input, output cam, cam track*).



Pod parametr *CamTrack* podłączamy skonfigurowany obiekt technologiczny. Dopóki wejście *Enable* będzie aktywne (stan wysoki), dopóty tor krzywkowy będzie pozostawał aktywny.

Funkcja *MC_CamTrack* pozwala na aktywację ścieżki krzywkowej w trzech trybach (*Mode*):

- Aktywacja/deaktywacja natychmiastowa (*Mode* = 0) – zmiana statusu wejścia *Enable* powoduje natychmiastowe uruchomienie bądź zatrzymanie przetwarzania skonfigurowanego toru krzywkowego (zgodnie z aktualną pozycją osi lub enkodera zewnętrznego).
- Aktywacja/deaktywacja od następnego cyklu (*Mode* = 1) – zmiana stanu wejścia *Enable* na wartość TRUE spowoduje uruchomienie skonfigurowanego toru krzywkowego od następnego cyklu (od pozycji początkowej). Skasowanie wejścia *Enable* spowoduje zakończenie przetwarzania toru krzywkowego po zakończeniu pełnego cyklu (osiągnięta pozycja końcowa).
- Załączenie/wyłączenie wyjścia (*Mode* = 2) – ustawienie wejścia *Enable* w tym trybie spowoduje automatyczne ustawienie wyjścia krzywkowego na stan wysoki. Skasowanie wejścia *Enable* spowoduje natychmiastowe ustawienie wyjścia krzywkowego w stan niski.

Podanie wartości TRUE na wejście *InvertOutput* spowoduje zanegowanie sygnału wyjściowego ścieżki.

Aktualny stan wyjścia toru krzywkowego odczytać można przez parametr *TrackOutput*. Pozostałe wyjścia dostarczają informacji na temat statusu pracy funkcji oraz ewentualnych błędów.

VII. Bezpieczeństwo w układach *Motion Control*

Niezmierzalnie istotną kwestią w obszarze instalacji przemysłowych jest bezpieczeństwo. Nie chodzi tu jedynie o nadrzędne bezpieczeństwo ludzi, ale również o zabezpieczenie maszyn oraz mienia przed uszkodzeniem, czy też przed wywołaniem szkód w środowisku naturalnym – ogólnie przed wystąpieniem zdarzeń niebezpiecznych.

W wielu zakładach przemysłowych, eliminacja ryzyka może zostać wykonana w łatwy sposób, bez konieczności nakładania rygorów na instalację. W przemyśle maszynowym, gdzie występują różnego rodzaju zagrożenia związane z ruchem urządzeń (np. prasy, roboty, piły, przenośniki), nie jest to już takie proste i wymaga zastosowania odpowiedniej struktury projektu.

Budowa systemu bezpiecznego z punktu widzenia prawa, a także wymagań użytkownika, powinna rozpocząć się już na etapie konceptu maszyny. Daje to sposobność zaprojektowania jej konstrukcji spójnie z zasadami bezpieczeństwa i pozwala na jej wytworzenie, instalację, obsługę, konserwację, a po zakończeniu jej cyklu życia na utylizację – zgodnie z wytycznymi odpowiednich norm.

24

Czym jest system *fail-safe*?

W obszarze krytycznych aplikacji, w których awarie mogą doprowadzić do zagrożenia życia ludzkiego, uszkodzenia instalacji technologicznej lub szkód związanych z zanieczyszczeniem środowiska, należy stosować systemy bezpieczne (tzw. *fail-safe*). Systemy takie redukują ryzyko związane z wypadkami dzięki temu, że w razie nieprawidłowości doprowadzają system do stanu bezpiecznego.

Układy typu *fail-safe* wykrywają sytuacje niebezpieczne oraz awarie urządzeń, a także własne, wewnętrzne błędy i usterki. Są one na ogół atestowane przez niezależne instytucje (w Niemczech TÜV).

Obowiązujące normy

Podstawą prawną w budowie maszyn przeznaczonych na rynek europejski (oznakowanie CE) jest tzw. *Dyrektywa Maszynowa* Unii Europejskiej (www.uokik.gov.pl/download.php?plik=1455). Bezpośrednio z tej dyrektywy (tzw. *Nowego podejścia*) wynikają normy zharmonizowane wyznaczające cechy konstrukcyjne oraz programowe urządzeń stwarzających potencjalne niebezpieczeństwo.

Dwie wiodące normy europejskie wyznaczone zostały przez komisję IEC (*International Electrotechnical Commission*) oraz organizację ISO (*International Organization for Standardization*).

Zgodność z Dyrektywą Maszynową może zostać zapewniona różnymi drogami: w ramach akceptacji maszyny przez autoryzowaną jednostkę badawczą,

- poprzez spełnienie wymagań norm zharmonizowanych (opis w dalszej części),
- przez dostarczenie dowodów bezpieczeństwa, co wiąże się ze zwiększonymi nakładami na badania i dokumentację.

EN/IEC 62061

Standard ten jest uszczegółowioną częścią bardziej rozległej normy (EN/IEC 61508) dotyczącej tzw. bezpieczeństwa funkcjonalnego związanego z bezpieczeństwem systemów elektrycznych, elektronicznych oraz programowalnych systemów elektronicznych. EN/IEC 62061 stanowi zbiór wytycznych przewidzianych dla w/w systemów w odniesieniu do bezpieczeństwa maszyn w automatyce przemysłowej. Norma obejmuje pełen łańcuch bezpieczeństwa – od czujnika do elementu wykonawczego.

Wymagania dotyczące działania nieelektrycznych elementów sterowania maszyn (np. hydraulicznych, pneumatycznych lub elektromechanicznych) związanych z bezpieczeństwem, nie są przedmiotem normy.

Bezpieczeństwo funkcjonalne

Norma IEC sprowadza się do realizacji koncepcji tzw. bezpieczeństwa funkcjonalnego. Z definicji – jest to podejście do wszystkich działań w cyklu życia systemów zawierających elektryczne i/lub elektroniczne i/lub programowalne elektroniczne elementy składowe nierozdzielnie związane z bezpieczeństwem technicznym.

SIL (Safety Integrity Level)

Nałożenie reguł bezpieczeństwa na system sterowania oraz konstrukcję maszyny w przypadku normy EN/IEC 62061 powoduje osiągnięcie odpowiedniego poziomu bezpieczeństwa, który określany jest jako tzw. poziom nienaruszalności bezpieczeństwa (*Safety Integrity Level*). W obszarze automatyki przemysłowej poziom SIL przyjmuje wartość od 1 do 3, gdzie SIL 3 to poziom najwyższy.

EN/ISO 13849-1

W przypadku normy ISO odnosimy się do bezpieczeństwa maszyn w rozumieniu elementów systemu sterowania związanych z bezpieczeństwem. Norma EN ISO 13849-1 ma szerokie zastosowanie i uwzględnia wszystkie technologie, w tym elektryczne, hydrauliczne, pneumatyczne i mechaniczne.

PL (Performance Level)

Stosując odpowiednie reguły bezpieczeństwa w stosunku do projektowanego układu zgodnie z wytycznymi EN/ISO 13849-1 osiągniemy tzw. poziom zapewnienia bezpieczeństwa (*Performance Level*). Przyjmuje on wartość od „a” do „e”, gdzie PL_e to poziom najwyższy.

Którą normę stosować?

W celu dostosowania urządzenia do wymogów Dyrektywy Maszynowej, możemy zastosować dowolną normę zharmonizowaną, czyli np. EN/ISO 13849-1 lub EN/IEC 62061. Obie te normy prowadzą do tych samych rezultatów, aczkolwiek przy zastosowaniu nieco odmiennych metod.

Normę EN/IEC 62061 łatwiej będzie narzucić na systemy wykorzystujące zaawansowane układy sterowania, natomiast norma EN/ISO 13849-1 jest wygodniejsza w układach podzespołów elektryczno-elektronicznych o niskim stopniu złożoności oraz pozwala na zastosowanie w konfiguracji z różnymi technologiami.

Dostosowanie maszyny do wymogów bezpieczeństwa

Procedura dostosowania ma na celu określenie poziomu wymogów, jakie powinna spełniać maszyna oraz ewentualne wskazanie środków, jakie należy przedsięwziąć, aby spełniała ona wymogi bezpiecznego użytkowania. Powinna składać się ona z kilku etapów, które umożliwią pełny opis maszyny: przegląd, ocena, plan, realizacja, nadzorowanie. Szczegółowy opis proponowanych kroków dostępny jest w następującej lokalizacji sieciowej:

https://www.automatyka.siemens.pl/docs/BM_ProcDostMaszyn.pdf.

Analiza ryzyka

W celu wyznaczenia wymaganego poziomu bezpieczeństwa – zgodnie z EN/IEC 62061 (SIL) lub EN/ISO 13849-1 (PL) – przewidziane zostały odpowiednie algorytmy pozwalające obliczyć poziom bezpieczeństwa jaki powinien w danej aplikacji zostać spełniony. Wyznaczając poziom SIL/PL dla budowanej maszyny, będziemy analizować potencjalne zagrożenia – rozmiar szkody, częstotliwość i/lub czas przebywania człowieka w strefie zagrożenia, prawdopodobieństwo jego wystąpienia oraz możliwość uniknięcia urazu. Precyzyjny opis takiej analizy dostępny jest w następującej dokumentacji technicznej:

https://www.automatyka.siemens.pl/docs/docs_ia/Bezpieczenstwo_funkcjonalne.pdf

25

Wskaźniki bezpieczeństwa

Niezależnie od wybranej metody analizy zagrożeń oraz normy, z którą zgodność będziemy chcieli uzyskać – w celu realizacji naszego celu, dobrać należy odpowiednie urządzenia o określonych parametrach bezpieczeństwa. Będziemy zatem musieli przede wszystkim wykorzystać odpowiednie urządzenia *fail-safe* ale również będzie trzeba połączyć je w odpowiedniej strukturze.

Dla opisywanych norm, wskaźniki jakościowe komponentów zapewniających bezpieczeństwo mogą być różne, dlatego dalsze rozważania poprzedzmy wprowadzeniem w nomenklaturę techniczną.

EN/IEC 62061

W celu określenia poziomu SIL naszego układu oraz wyboru odpowiednich komponentów spełniających wymagany poziom, będziemy korzystać ze wskaźników podanych przez producenta danego podzespołu lub systemu. W obszarze normy EN/IEC 62061 spotkamy się z następującymi pojęciami, które należy wziąć pod uwagę:

- **SRECS** (*Safety Related Electrical Control System*) – elektryczny system sterowania maszyny, którego uszkodzenie może skutkować bezpośrednim wzrostem ryzyka. Pojęcie to obejmuje wszystkie elementy wymagane do realizacji funkcji bezpieczeństwa – np. czujnik, sterownik oraz element wykonawczy.
- **SRCF** (*Safety Related Control Function*) – funkcja sterowania związana z bezpieczeństwem. Funkcja sterowania o określonym poziomie nienaruszalności bezpieczeństwa realizowana przez SRECS, przeznaczona do utrzymania warunków bezpieczeństwa maszyny lub zapobiegania bezpośredniemu wzrostowi ryzyka.
- **SFF** (*Safe Failure Fraction*) – wskaźnik uszkodzeń bezpiecznych. Frakcja całkowitej liczby uszkodzeń podsystemu, które nie spowodują uszkodzeń niebezpiecznych. Parametr podawany jest przez producenta urządzenia.
- **DC** (*Diagnostic Coverage*) – pokrycie diagnostyczne. Zmniejszenie prawdopodobieństwa niebezpiecznego uszkodzenia sprzętu w wyniku działania automatycznych testów diagnostycznych. Zależność pomiędzy oczekiwanym poziomem bezpieczeństwa a wartością współczynnika DC została opisana w poniższym dokumencie:
<https://support.industry.siemens.com/cs/ww/en/view/35444114>
- **PFH** (*Probability of dangerous Failure per Hour*) – prawdopodobieństwo niebezpiecznego uszkodzenia na godzinę. Przykładowe wartości tego współczynnika w odniesieniu do sterowań SIMATIC, a także poziomy bezpieczeństwa jakie da się uzyskać, można odszukać w niniejszym dokumencie:
<https://support.industry.siemens.com/cs/ww/en/view/27832836>

Producent danego urządzenia bezpieczeństwa zazwyczaj podaje w dokumentacji zgodność z odpowiednimi standardami (np. maksymalny SIL jaki można uzyskać stosując dany komponent) lub deklaruje wartości wspomnianych parametrów (np. PFH lub SFF).

EN/ISO 13849-1

Analogicznie do normy EN/IEC 62061 – w obrębie EN/ISO 13849-1 również będziemy spotykać się z charakterystyczną terminologią oraz parametrami jakościowymi. Warto znać podstawowe pojęcia oraz ich znaczenie:

- **SRP/CS** (*Safety Related Parts of Control System*) – elementy związane z bezpieczeństwem systemu sterowania, które odpowiadają na sygnały wejściowe związane z bezpieczeństwem i generują wyjściowe sygnały bezpieczeństwa.
- **MTTF** (*Mean Time To Failure*) – średni czas między uszkodzeniami. W odniesieniu do elementów urządzenia, MTTF może być określony przez analizę danych eksploatacyjnych lub prognozowanie. Więcej informacji o tym współczynniku można odszukać w następującej lokalizacji sieciowej:
<https://support.industry.siemens.com/cs/ww/en/view/16818490>
- **DC** (*Diagnostic Coverage*) – pokrycie diagnostyczne. Zmniejszenie prawdopodobieństwa niebezpiecznego uszkodzenia sprzętu w wyniku działania automatycznych testów diagnostycznych.
- Zależność pomiędzy oczekiwanym poziomem bezpieczeństwa a wartością współczynnika DC została opisana w poniższym dokumencie:
<https://support.industry.siemens.com/cs/ww/en/view/35444114>
- **CCF** (*Common Cause Failure*) – usterka spowodowana wspólną przyczyną niezależną od parametrów elementów bezpieczeństwa (np. zwarcie w obwodzie). Przyczyną bezpośrednią mogą być warunki środowiskowe, np. wibracje, wilgotność, zapylenie, wysoka lub niska temperatura/ciśnienie, etc. Parametr wyznacza się przez ocenę systemu pod kątem skuteczności zastosowanych środków zapobiegających uszkodzeniom różnych jednostek spowodowanych pojedynczym, niezależnym zdarzeniem.

Przykładowe aplikacje, szczegółowy opis przedstawionych powyżej współczynników, a także wiele innych informacji na temat bezpieczeństwa funkcjonalnego oraz systemów *fail-safe* odnaleźć można w następującej dokumentacji:
https://www.automation.siemens.com/salesmaterial-as/functional_examples/en/functional_example_safety_integrated_fa_en.pdf

26 Realizacja funkcji bezpieczeństwa

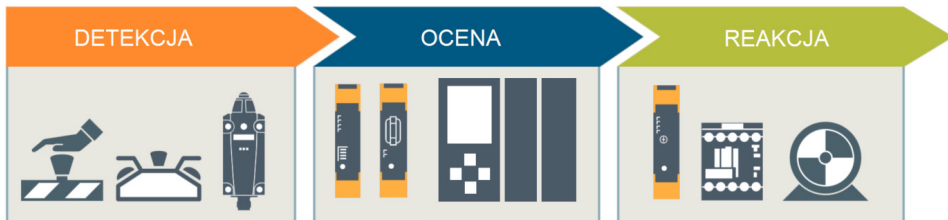
Wiemy już jakie obowiązują normy prawne, jak wyznaczyć ich poziom, a także jakie parametry urządzeń należy wziąć pod uwagę przy doborze komponentów układu. Pozostaje zatem zagadnienie w jaki sposób zbudować system, który spełni określone wymagania bezpieczeństwa

Łańcuch bezpieczeństwa

Rozważając kompleksowo układ sterowania bezpiecznego, będziemy realizować odpowiednie funkcje *fail-safe* w poszczególnych obszarach tzw. łańcucha bezpieczeństwa.

Celem układu jest redukcja ryzyka wystąpienia zdarzeń niebezpiecznych, do poziomu akceptowalnego. Poza standardowo zaimplementowaną funkcjonalnością (nadzór i sterowanie) układ powinien zostać zaprojektowany w taki sposób, aby zarówno błąd urządzenia, jak i usterka spowodowana przyczyną zewnętrzną (np. awaria czujnika czy stycznika) nie powodowały ograniczenia funkcji bezpieczeństwa. W tym celu system oceny funkcji bezpieczeństwa musi monitorować nie tylko status generowanych sygnałów ale również wewnętrzne błędy, zużycie podzespołów czy poprawność zasilania. W razie wykrycia stanu naruszającego bezpieczeństwo, system przechodzi w zdefiniowany stan bezpieczny.

Zakres komponentów bezpiecznych układu sterowania możemy podzielić na trzy grupy.



Detekcja

Urządzeniami bezpieczeństwa służącymi do wykrywania lub sygnalizacji zagrożeń w torze bezpieczeństwa są elementy takie jak: stop awaryjny (E-STOP), wyłącznik pozycyjny (linkowy/zawiasowy/magnetyczny), bariery/kurtyny świetlne, czujniki laserowe, bramy bezpieczeństwa, rygle, maty naciskowe, układy sterowania oburęcznego, panele HMI, etc. Należy zwrócić uwagę, że w zależności od oczekiwanego poziomu bezpieczeństwa każdy z zastosowanych komponentów musi posiadać odpowiednią certyfikację, a co za tym idzie adekwatne funkcje bezpieczeństwa. Może być również wymagane specyficzne połączenie elektryczne (np. jedno – lub dwutorowe) lub sieciowe (komunikacja *PROFIsafe* – również bezprzewodowo).

Ocena

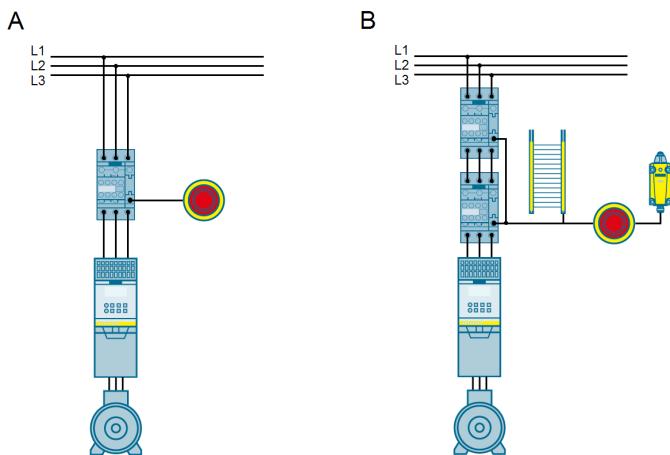
Za logikę bezpieczeństwa funkcjonalnego, czyli ocenę stanu układu detekcyjnego oraz realizację zadań przez elementy wykonawcze, odpowiadać może w zaproponowanej strukturze np. układ elektryczny, przekaźnik bezpieczeństwa, sterownik PLC, przekształtnik częstotliwości czy też układ monitorowania parametrów silnika (SIMOCODE). W zależności od złożoności konstrukcyjnej układu, ilości sygnałów oraz elementów wykonawczych, a także wymaganych funkcji bezpieczeństwa, warto będzie optymalizować system zarządzania bezpieczeństwem. Wybór rozwiązania uzależniać powinniśmy od oczekiwanego poziomu bezpieczeństwa, nakładu pracy (programowanie, okablowanie) czy też kosztu danego systemu. Dokładniejszą analizą tego zagadnienia zajmiemy się w dalszej części dokumentu.

Reakcja

Elementami wykonawczymi na ogół będą podstawowe komponenty systemu automatyki takie jak stycznik, przekaźnik, cewka zaworu/rygla, falownik czy układ rozruchowy czyli wszelkie aktuatory odpowiadające za przekazanie energii do elementów ruchomych (np. silnika bądź siłownika).

26.1 Układ hard-wired

Historycznie, na początku rozwoju technologii bezpieczeństwa system zabezpieczeń ograniczał się do stosowania standardowych elementów układu elektrycznego, które w razie sytuacji niebezpiecznej, pozwalały na odcięcie elementu wykonawczego od źródła energii. Przykładem takiego rozwiązania było zastosowanie pojedynczego stycznika elektromechanicznego, w którego obwód zasilający wpięte były jednorotorowe czujniki bezpieczeństwa (np. stop awaryjny czy kurtyna świetlna) – rys. 1A.



rys. 1

Rozwiązanie takie szybko okazało się mało skuteczne, gdyż w przypadku awarii stycznika (np. sklejenie styków), system stawał się niebezpieczny, gdyż jego wyłączenie nie było możliwe.

Inżynierowie rozpoczęli więc poszukiwania rozwiązań bardziej niezawodnych. Początkowo stosowano obwody elektryczne o wyższej złożoności, np. dwa styczniki połączone szeregowo (rys. 1B) oraz większą ilość czujników aktywujących cewki styczników. W takim układzie awaria jednego ze styczników nie ograniczała funkcji bezpieczeństwa.

Zwiększyło to znacznie poziom bezpieczeństwa jednak nadal brakowało mechanizmu, który informowałby o uszkodzeniach elementów łańcucha bezpieczeństwa i pozwalał na monitorowanie parametrów podzespołów. W późniejszym czasie zapotrzebowanie to doprowadziło do powstania pierwszego przełącznika bezpieczeństwa.

W nowoczesnych systemach bezpieczeństwa nie stosuje się na ogół takich rozwiązań. Budując taką prostą strukturę (podłączając wyłącznik bezpieczeństwa bezpośrednio pod cewkę stycznika) jesteśmy w stanie uzyskać maksymalnie poziom bezpieczeństwa SIL1/PLb. Poziom ten jest jednak uzyskany przez zastosowanie komponentów nieposiadających certyfikacji *fail-safe* (w przeciwieństwie do przełącznika bezpieczeństwa, falownika bądź sterownika programowalnego F-CPU), co za tym idzie producent maszyny musi oszacować poziom bezpieczeństwa na podstawie bazowych parametrów (np. PFH czy MTTF) użytych w projekcie komponentów.

W prostych układach zamiast stycznika zastosować można alternatywnie rozłącznik bezpieczeństwa.

26.2 Przełącznik bezpieczeństwa

Wraz z rozwojem układów bezpieczeństwa określone zostały cechy układu jakie powinien on spełniać. Jednymi z podstawowych informacji jakie system powinien analizować stała się kontrola stanu urządzeń toru bezpieczeństwa – rodzaju oraz poprawności połączeń obwodów, czasów rozbieżności przełączania styków pomiędzy kanałami, stanu styków (np. sklejenie styków stycznika – sprzężenie zwrotne), zwarcie pomiędzy kanałami oraz inne parametry diagnostyczne komponentów odpowiedzialnych za bezpieczeństwo systemu.

Systemy zaczęto projektować w taki sposób aby awaria któregośkolwiek urządzenia *fail-safe* uniemożliwiała ponowne uruchomienie maszyny.

Podstawowym elementem nowoczesnego układu bezpieczeństwa stał się przełącznik bezpieczeństwa. Jest to certyfikowane urządzenie elektryczne lub elektroniczne, które wykonuje i gwarantuje (zgodnie z normą) odłączenie zasilania urządzeń wykonawczych w sytuacji awaryjnej. Przełączniki bezpieczeństwa muszą być zaprojektowane w taki sposób, aby zarówno ich błąd wewnętrzny, jak i usterka spowodowana przyczyną zewnętrzną (czujnikiem lub elementem wykonawczym) nie powodowały ograniczenia funkcji bezpieczeństwa.

W związku powyższym przełącznik bezpieczeństwa, w wykonaniu klasycznym, skonstruowany jest z trzech styków przełącznikowych (dwa przełączniki z rozwieraniem wymuszonym – styki lustrzane). Budowa nadmiarowa zapewnia zachowanie funkcji urządzenia nawet podczas błędów okablowania lub zwarcia elementów aktywujących. Dodatkowo obwód sprzężenia zwrotnego pozwala na monitorowanie stanu urządzeń wykonawczych. W razie awarii któregoś z nich, dopiero wymiana na sprawny element pozwoli na ponowne uruchomienie maszyny.

Funkcje diagnostyczne przełącznika bezpieczeństwa

Podłączenie dwutorowe elementów aktywujących funkcje bezpieczeństwa, daje nie tylko wyższy poziom niezawodności ale również rozszerzone możliwości diagnostyki czujnika. Konfiguracja taka posiada zazwyczaj następujące funkcje:

Czujniki dwutorowe

Warto wspomnieć w tym miejscu o rodzaju połączenia czujników aktywujących funkcje bezpieczeństwa. W zależności od wymaganego poziomu bezpieczeństwa, przykładowy stop awaryjny, możemy podłączyć jako:

- pojedynczy obwód jednostykowy, czyli jednotorowo.
W tym wypadku pojedyncza ścieżka elektryczna pozwoli zrealizować nam tę część projektu na poziomie **SIL 2/PLC**,
- obwód zdublowany, czyli dwa styki NC aktywowane w tym samym momencie przy wciśnięciu guzika bezpieczeństwa, tzw. połączenie dwutorowe. Taki układ pozwoli osiągnąć najwyższy poziom bezpieczeństwa **SIL3/PLe**.

Testowanie toru bezpieczeństwa

Warto wspomnieć w tym miejscu o rodzaju połączenia czujników aktywujących funkcje bezpieczeństwa. W zależności od wymaganego poziomu bezpieczeństwa, przykładowy stop awaryjny, możemy podłączyć jako:

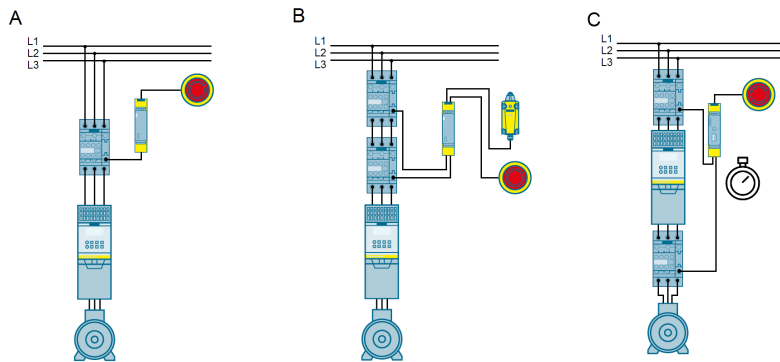
- detekcja zwarcia między kanałami wejściowymi.
Charakterystyczną metodą wykrywania takiej usterki w obwodzie wejściowym, jest różnicowe testowanie impulsowe. Przełącznik bezpieczeństwa wysyła impulsy o wysokiej częstotliwości na kanał pierwszy i drugi – przesunięte fazowo względem siebie. W momencie wystąpienia zwarcia, stykownik otrzymuje w sprzężeniu zwrotnym, ten sam sygnał impulsowy na obu kanałach. Układ mikroprocesorowy analizuje tę zależność, w razie awarii wykrywając zwarcie międzykanałowe.
- monitorowanie czasu przełączeń.
W przypadku podłączenia dwutorowego styków czujnika, istotne jest monitorowanie czasu pomiędzy aktywacją obu kanałów. Mikroprocesor przełącznika bezpieczeństwa wykrywa zarówno niezałączenie jednego z kanałów, jak i zbyt dużą rozbieżność czasową pomiędzy ich aktywacją (np. w przypadku sterowania dwuręcznego). W obu przypadkach awaria spowoduje przejście wyjść przełącznika w stan bezpieczny.
- sprawdzanie poprawności obwodu resetującego.
W momencie przejścia w stan niebezpieczny ponowne zazbrojenie układu nie będzie możliwe jeśli wykryta zostanie usterka obwodu potwierdzającego.
- analiza ciągłości układu wyjściowego.
Niektóre przełączniki bezpieczeństwa posiadają funkcję sprawdzania poprawności połączenia elementów wykonawczych. Przerwanie obwodu wyjściowego lub awaria aktuatora spowoduje wyzwolenie stanu niebezpiecznego. Zwarcia w układzie wyjściowym nie są krytycznym punktem analizy bezpieczeństwa gdyż w momencie aktywacji stanu niebezpiecznego obwody zasilające elementy wykonawcze zostaną odcięte od zasilania.

Przykłady

W zależności od konfiguracji struktury systemu osiągnąć możemy wymagany poziom bezpieczeństwa. Rozważmy kilka typowych przypadków aktywujących funkcje bezpieczeństwa w układach napędowych.

W aplikacjach takich będziemy odcinać od źródła energii silnik elektryczny (ew. wprowadzać go w bezpieczny tryb pracy), podłączony do niego bezpośrednio lub za pośrednictwem przekształtnika częstotliwości. W przykładowych rozważaniach na temat wariacji zastosowań przełącznika bezpieczeństwa (elementu oceny ryzyka) elementem aktywującym może być przycisk awaryjny (aktywator podłączony dwutorowo), natomiast elementem wykonawczym stycznik.

W podstawowej strukturze układu wyposażymy tor bezpieczeństwa w jeden element aktywujący, przełącznik bezpieczeństwa oraz stycznik umiejscowiony przed napędem (lub bezpośrednio przed silnikiem – jeśli układ nie będzie posiadał regulatora częstotliwości). Sytuację taką obrazuje rys. 2A z poniższej ilustracji. Taki układ pozwoli osiągnąć nam maksymalnie poziom bezpieczeństwa **SIL2** lub **PLd**.



rys. 2

W aplikacjach, gdzie zagrożeniem mogą być sytuacje powodujące ciężkie urazy lub śmierć użytkownika, wymagany będzie wyższy poziom bezpieczeństwa. Podobnie jak w poprzedniej sekcji, możemy podnieść go przez wzbogacenie łańcucha bezpieczeństwa o większą ilość elementów aktywujących oraz wykonawczych. Zastosowanie dwóch styczników odcinających dopływ prądu do falownika (opcjonalnie większa ilość aktywatorów, np. wyłącznik krańcowy) – daje sposobność osiągnięcia poziomu bezpieczeństwa **SIL3/PLe**. Układ taki ilustruje rys. 2B.

Umiejscowienie stycznika

Zwróćmy jeszcze uwagę na istotne zagadnienie dla systemu ze zmienną regulacją częstotliwości. W takim układzie pojawia się pytanie – gdzie w obwodzie doprowadzającym zasilanie do silnika powinien się znaleźć stycznik, będący elementem toru bezpieczeństwa. Możemy umieścić go przed lub za przekształtnikiem częstotliwości. Jest to dosyć istotna kwestia i każdy z wyborów ma swoje wady i zalety. W trakcie pracy normalnej nie jest istotne położenie styczników. Jeśli jednak wystąpi zdarzenie niebezpieczne, cechy stycznika położonego za przekształtnikiem mogą doprowadzić do awarii.

Usytuowanie stycznika przed przekształtnikiem nie powinno spowodować zdarzeń nietypowych. Może się jednak okazać, że w pewnego rodzaju aplikacjach większą pewność działania funkcji bezpieczeństwa uzyskamy podłączając drugi

stycznik bezpośrednio przed silnikiem. W takim układzie trzeba zwrócić uwagę na charakterystykę elektryczną obwodu. Należy mieć na uwadze, iż stycznik przewidziany jest zazwyczaj do rozłączania obwodu prądu przemiennego o określonej częstotliwości (np. 50Hz). Jeśli będziemy rozłączać obwód elektryczny o regulowanej częstotliwości (za przekształtnikiem), a będzie pracował on na wysokiej lub niskiej częstotliwości, to w momencie rozłączenia zasilania – falownik przez moment będzie generował energię, która nie będzie miała ujęcia. Na stykach stycznika może wtedy wystąpić przeskok łuku elektrycznego, co może skończyć się pożarem lub sklejeniem styków stycznika. Zjawisko to może doprowadzić również do wytworzenia zwarcia w obwodzie elektrycznym przekształtnika częstotliwości, a co za tym idzie do spalenia tranzystorów IGBT (trwałe uszkodzenie falownika).

Aby zapobiec takiej sytuacji, w pierwszej kolejności odłączone powinno zostać zasilanie falownika (stycznik przed falownikiem), a dopiero po zdefiniowanym opóźnieniu (np. 100ms) obwód zasilania silnika (stycznik za falownikiem), przez który nie powinien płynąć już prąd. Taki układ zwłoczny możemy zrealizować np. przez zastosowanie przekaźnika bezpieczeństwa z funkcją opóźnienia aktywacji sygnałów wyjściowych. Schemat połączenia przedstawiony został na rys 2C.

Przekaźnik programowalny

W zależności od skomplikowania projektu w obszarze układu bezpieczeństwa – zastosować możemy konfigurację przekaźników bezpieczeństwa lub bardziej zaawansowane urządzenie oceny sygnałów *fail-safe*. Poziom wyżej, w hierarchii takich rozwiązań, zajmuje tzw. programowalny przekaźnik bezpieczeństwa. Jest to urządzenie mikroprocesorowe z certyfikacją bezpieczeństwa, które pozwala na swobodą definicję logiki załączania wyjść bezpiecznych (F-DO) na podstawie odczytanych stanów wejść *fail-safe* (F-DI). Programowanie takich urządzeń odbywa się zazwyczaj przez dedykowane oprogramowanie (np. Safety ES dla przekaźników SIRIUS), posiadające możliwość zdefiniowania prostej logiki programu. W przypadku większej ilości wejść/wyjść bezpiecznych – zastosowanie przekaźnika programowalnego może okazać się łatwiejsze oraz szybsze w implementacji, a może i tańsze od rozwiązania klasycznego.

Więcej informacji na temat przekaźników bezpieczeństwa z rodziny SIRIUS odnaleźć można w następującej lokalizacji sieciowej:
publikacje.siemens-info.com

26.3 Zintegrowane funkcje bezpieczeństwa

Nowoczesne trendy rozwoju technik napędowych oraz systemów bezpieczeństwa prowadzą do popularyzacji zintegrowanych funkcji bezpieczeństwa. Oznacza to, że przekaźnik bezpieczeństwa zostaje włączony konstrukcyjnie w budowę przekształtnika częstotliwości.

Rozwiązanie takie niesie za sobą sporo zalet oraz udogodnień, m.in.:

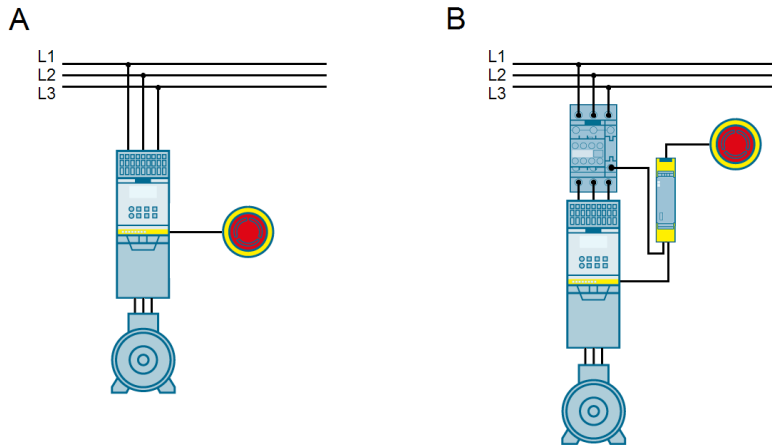
- redukcja kosztów – ograniczamy wydatki na sprzęt (przekaźniki bezpieczeństwa, styczniki) oraz okablowanie niezbędne do połączenia elementów elektrycznych,
- oszczędność miejsca – mniejsza ilość urządzeń w projekcie prowadzi do uszczuplenia gabarytów maszyny,
- szybki montaż – brak dodatkowych komponentów sprzętowych oznacza również brak konieczności ich łączenia w kompletny układ elektryczny,
- czas reakcji układu – zintegrowane funkcje bezpieczeństwa napędu, dzięki zastosowaniu szybkich układów elektronicznych, mają znacznie krótsze czasy reakcji na sygnał aktywujący niż zastosowanie konwencjonalnych komponentów elektromechanicznych,

- przejrzystość projektu – wykluczenie zbędnych elementów sprzętowych, prowadzi do łatwiejszego, ustandaryzowanego oraz szybszego uruchomienia oraz minimalizuje możliwość popełnienia błędu montażowego; zaoszczędzimy również czas podczas napraw, konserwacji lub rozbudowy systemu,
- ograniczenie ryzyka – warto mieć również na uwadze, że im mniejsza ilość elementów w strukturze projektu, tym mniejszy stopień skomplikowania, a co za tym idzie niższe prawdopodobieństwo wystąpienia awarii.

Przykłady

Poza wymienionymi powyżej udogodnieniami zintegrowanych funkcji bezpieczeństwa, ich zastosowanie jest analogiczne jak w przykładach omówionych w poprzedniej sekcji dla przekaźników bezpieczeństwa. Wykorzystując wejścia bezpieczne napędu uzyskamy również takie same poziomy bezpieczeństwa.

Podłączając zatem aktywator (np. przycisk bezpieczeństwa) dwutorowo bezpośrednio do wejścia bezpiecznego napędu (rys. 3A) uzyskamy maksymalnie poziom nienaruszalności **SIL2** lub **Pld**. Jeśli struktura układu bezpieczeństwa zostanie wzbogacona o zewnętrzny przekaźnik bezpieczeństwa oraz nadmiarowy element wykonawczy (np. stycznik) – możemy osiągnąć najwyższy poziom bezpieczeństwa – **SIL3/PLe**. Struktura taka została zobrażowana na rys. 3B.



rys. 3

Pozostaje nam rozważyć kwestię – jaka funkcja bezpieczeństwa w danej aplikacji powinna zostać aktywowana. Zróżnicowanie funkcji bezpieczeństwa wynika z wieloznaczności pojęcia „stanu bezpiecznego” – w każdej aplikacji sformułowanie to może oznaczać inną charakterystykę zatrzymania lub jedynie spowolnienia pracy napędu.

Szczegółowy opis najczęściej stosowanych funkcji zintegrowanych zostanie przedstawiony w kolejnym rozdziale.

26.4 Sterownik *fail-safe* (F-CPU)

Znamy już szereg urządzeń oceny bezpieczeństwa – od podstawowych komponentów bez certyfikacji, przez przekaźniki bezpieczeństwa, kończąc na funkcjach zintegrowanych napędów.

Najwyżej w hierarchii urządzeń oceny bezpieczeństwa możemy umieścić swobodnie programowalny sterownik PLC rozszerzony o funkcjonalność bezpieczeństwa. W takim przypadku komplet funkcji przekaźnika bezpieczeństwa zawieramy w nadrzędnym systemie sterowania. Czujniki oraz aktywatory bezpieczeństwa będą w tym przypadku bezpośrednio podłączone pod wejścia bezpieczne sterownika (F-DI), a obsługę sterowania elementami wykonawczymi realizować będą wyjścia bezpieczne (F-DO).

Struktura taka pozwala jeszcze bardziej odchudzić system pod kątem ilości komponentów sprzętowych, a także ustandaryzować projekt w jego centralnym punkcie. Zyskujemy także otwartość w zakresie programistycznej – logika programu bezpieczeństwa może być swobodnie zdefiniowana jako część programu sterownika.

Przykłady

Ponownie przeanalizujemy wcześniej zaproponowane układy bezpieczeństwa w tym wypadku jednak uwzględnimy możliwość przesłania informacji z czujników oraz do aktuatorów przez sieć przemysłową za pośrednictwem sterownika PLC w wykonaniu *fail-safe*.

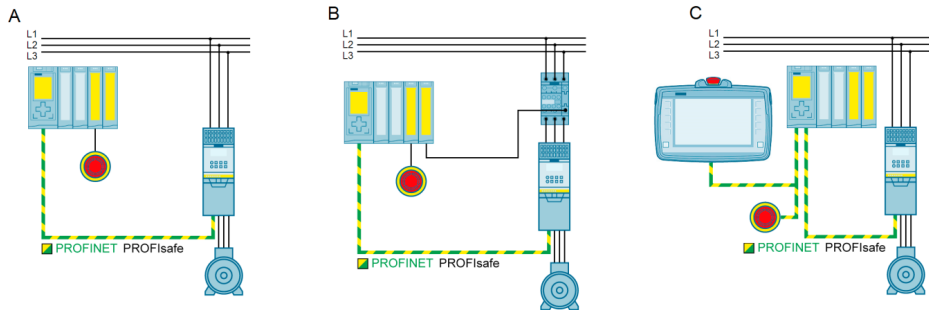
Komunikacja PROFIsafe

Dzięki komunikacji *PROFIsafe* – możemy aktywować oraz realizować funkcje bezpieczeństwa przez urządzenia połączone fizycznie przez sieci przemysłową PROFINET. Standard bezpiecznej komunikacji *PROFIsafe* wspierany jest zarówno przez urządzenia aktywujące – np. panele operatorskie SIMATIC HMI (zintegrowany przycisk bezpieczeństwa) bądź przyciski z rodziny SIRIUS ACT z komunikacją PROFINET, jak i przez elementy wykonawcze – np. napędy SINAMICS (realizacja zintegrowanych funkcji bezpieczeństwa) lub rozproszone systemy wejść/wyjść ET 200.

Komponenty obsługujące komunikację bezpieczną pozwalają na realizację układu bezpieczeństwa bez konieczności okablowania elektrycznego (*hard-wired*) – możemy swobodnie skorzystać z obsługi protokołu *PROFIsafe* w sieci przemysłowej PROFINET.

Szczegółowy opis standardu *PROFIsafe* odszukać można na stronach organizacji PNO Polska (www.profibus.org.pl). Szerszy opis komponentów z rodziny SIRIUS ACT dostępny jest pod następującym adresem sieciowym:
www.siemens.pl/sirius-act.

Posługując się więc analogią do poprzedniej sekcji rozważmy najpierw przypadek, gdzie aktywujemy zintegrowaną funkcję bezpieczeństwa napędu przez stop awaryjny. W strukturze ze sterownikiem *fail-safe* nie musimy stosować zewnętrznych przekaźników bezpieczeństwa. Przycisk stopu awaryjnego możemy podłączyć elektrycznie pod wejścia bezpieczne F-DI, natomiast funkcja zatrzymania napędu aktywowana może zostać przez sieć *PROFIsafe*. Taka struktura pozwoli na osiągnięcie poziomu bezpieczeństwa **SIL2/PLd**. Przykładowa konfiguracja została przedstawiona na rys. 4A.



rys. 4

W celu osiągnięcia wyższego poziomu bezpieczeństwa (**SIL3/PLe**) w strukturze scentralizowanej – konieczne będzie zastosowanie dodatkowego stycznika oraz jego wystawienie przez wyjścia bezpieczne F-DO (rys. 4B).

Idąc krok dalej – w nowoczesnych układach – całość łańcucha bezpieczeństwa jesteśmy w stanie zrealizować przez komunikację w sieci PROFINET, przy założeniu, że zastosowane w projekcie elementy wspierają obsługę wymiany danych w standardzie **PROFIsafe**. W przykładowym układzie (rys. 4C) będziemy mogli uzyskać maksymalnie poziom bezpieczeństwa **SIL2/PLd**.

Struktura jest możliwie prosta, przejrzysta oraz łatwa w implementacji. Podwyższenie poziomu bezpieczeństwa wymagać będzie jednak zastosowania dodatkowego stycznika (np. przed napędem), gdyż przekształtniki częstotliwości SINAMICS mogą maksymalnie zapewnić **SIL2/PLd** w zakresie funkcji zintegrowanych.

Układy łagodnego rozruchu ET 200SP

Warto nadmienić w tym miejscu, że w grupie komponentów rozruchowych z rodziny SIMATIC ET 200SP (układy napędowe bez regulacji częstotliwości) – istnieje możliwość uzyskania poziomu bezpieczeństwa **SIL3/PLe** bez konieczności zastosowania zewnętrznego obwodu stycznikowego odcinającego zasilanie. Wynika to z faktu, iż możemy skonfigurować systemowy zasilacz *fail-safe* (F-PM) w celu odcięcia zasilania grupy starterów *fail-safe*. Sterowanie odcięciem zasilania może zostać wykonane przez wejścia cyfrowe modułu F-PM (np. przycisk awaryjny) lub przez magistralę komunikacyjną PROFINET z obsługą ramek **PROFIsafe**.

Podsumowanie

Kompletne kompendium informacji na temat systemów bezpieczeństwa SIMATIC, obowiązujący norm oraz koncepcji, charakterystyki elementów aktywujących/ wykonawczych, komunikacji bezpiecznej oraz wielu innych zagadnień można znaleźć na stronie internetowej: www.siemens.pl/safety.

27

Funkcje bezpieczeństwa *Motion Control*

Wymagania norm bezpieczeństwa określają kilka kategorii bezpiecznego zatrzymania układu napędowego. Kategoria 0 definiuje procedurę natychmiastowego odłączenia zasilania od napędu maszyny (zatrzymanie niekontrolowane). Kategoria 1 mówi o odłączeniu zasilania dopiero po kontrolowanym doprowadzeniu układu napędowego do stanu spoczynku. Z kolei kategoria 2 określa zatrzymanie w sposób kontrolowany, gdzie zasilanie po unieruchomieniu maszyny nie zostaje w ogóle odcięte.

Potraktujmy te wytyczne jako początek naszych rozważań w zakresie funkcji bezpieczeństwa jakie mogą zostać zaaplikowane w naszym projekcie.

Zatrzymanie bezpieczne

W związku z powyższym wstępem – określając funkcje bezpieczeństwa oraz metodę ich realizacji, należy się również zastanowić co oznacza doprowadzenie osi napędowej naszego układu do stanu bezpiecznego.

Być może w większości przypadków będzie to zatrzymanie napędu. W wielu aplikacjach konieczne jednak okaże się zatrzymanie z odpowiednią dynamiką. Nie zawsze odcięcie źródła zasilania będzie dobrym rozwiązaniem, zarówno nagłe jak i zbyt powolne zatrzymanie elementu ruchomego może spowodować zagrożenie dla ludzi, uszkodzenie urządzeń lub wygenerowanie innych poważnych strat.

Wiele układów nie będzie mogło być w pełni zatrzymanych, a jedynie zostanie ograniczona prędkość pracy.

Jeszcze inną sytuacją może być system, gdzie siła zewnętrzna (np. grawitacyjna) będzie powodować przyspieszenie pracy napędu po odcięciu jego zasilania – w takim wypadku może okazać się konieczne zastosowanie silnika z hamulcem.

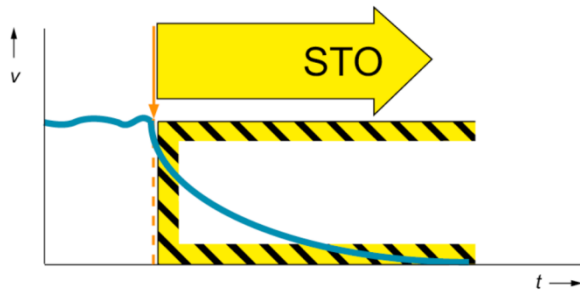
W zakresie bezpieczeństwa układów napędowych zdefiniowana została grupa funkcji, których realizacja powinna wprowadzić system w określony stan bezpieczny. W dalszej części rozważymy kilka typowych funkcji bezpieczeństwa stosowanych w celu zabezpieczenia układów napędowych.

27.1 STO – *Safe Torque Off*

Podstawową oraz najprostszą funkcją bezpieczeństwa jest odcięcie zasilania układu napędowego. Wynikiem aktywacji funkcji STO jest bezpieczne odcięcie przepływu energii do silnika, za czym idzie wstrzymanie generowania momentu obrotowego. Silnik przechodzi w stan wybiegu, czyli wyhamowuje zgodnie z tarcieciem oraz bezwładnością układu mechanicznego. W związku z powyższym – funkcja może zostać zastosowana jedynie w aplikacjach gdzie dynamika zatrzymania nie wpływa na bezpieczeństwo, np. przy przenośnikach taśmowych.

Charakterystyczną cechą tej procedury jest fakt, że po jej aktywacji – napęd nie może wytworzyć momentu obrotowego oraz ruchu stwarzającego zagrożenie – wymagane jest zdjęcie sygnału aktywującego STO oraz potwierdzenie (ponowne załączenie zasilania).

Poniższy wykres, zależności prędkości napędu w czasie, przedstawia reakcję układu na wywołanie funkcji STO.



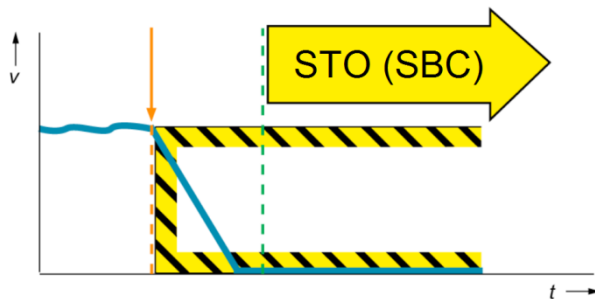
Zgodnie z opisem z poprzedniego rozdziału – implementacja procedury STO (określonej przez normę bezpieczeństwa) może zostać zrealizowana w sposób dowolny. Zasadniczą kwestią jest więc określenie oczekiwanego poziomu SIL/PL oraz dobór odpowiednich komponentów sprzętowych.

Najniższy poziom bezpieczeństwa uzyskamy przez elektryczne odcięcie układu od źródła zasilania. Stosując urządzenie certyfikowane (przełącznik bezpieczeństwa, napęd z funkcjami zintegrowanymi lub sterownik F-PLC), ewentualnie włączając w obwód nadmiarowy stycznik odcinający źródło zasilania – otrzymamy wyższym poziom bezpieczeństwa realizacji funkcji STO.

STO jest funkcją elementarną i najczęściej wbudowywaną w przemienniki częstotliwości. W zakresie rodziny SINAMICS większość urządzeń posiada tę funkcję w standardowej konfiguracji.

27.2 SS1 – Safe Stop 1

W przeciwieństwie do funkcji STO, gdzie zatrzymanie napędu odbywa się wybiegiem (w sposób niekontrolowany) – przy zatrzymaniu kontrolowanym SS1, następuje zmniejszenie prędkości aż do całkowitego zatrzymania (lub do osiągnięcia zdefiniowanej prędkości granicznej), a dopiero potem napęd zostaje odizolowany od źródła zasilania (STO). Jeśli silnik wyposażony jest w hamulec – zostanie on również bezpiecznie uruchomiony (SBC – Safe Break Control) równocześnie z funkcją odcięcia momentu. Procedura SS1 jest więc dwuetapowa, co zostało zaprezentowane na poniższym przebiegu.



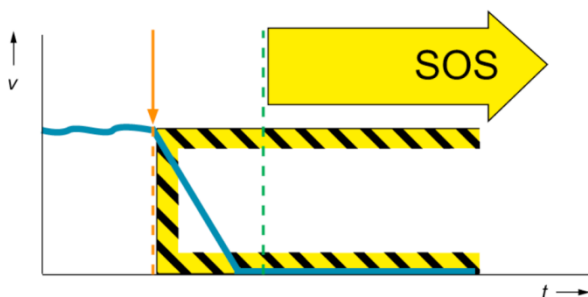
Funkcja SS1 stosowana jest gdy zdarzenie niebezpieczne wymaga możliwie szybkiego zatrzymania urządzenia – instalacje gdzie wybieg napędu stanowi zagrożenie. Są to zazwyczaj układy o dużej bezwładności lub pracujące z wysokimi prędkościami, np. piły, wrzeciona, wirówki, nawijarki lub prasy. W aplikacjach gdzie

działając zewnętrzne siły (np. grawitacyjna), niezbędne będzie zastosowanie zewnętrznego hamulca w celu bezpiecznego unieruchomienia napędu po odłączeniu siły generującej moment obrotowy.

Podobnie jak w przypadku poprzednim realizacja funkcji SS1 może odbywać się przez klasyczny układ elektryczny z przekaźnikiem bezpieczeństwa lub przez nadrzędny sterownik *fail-safe*. Funkcja zintegrowana w napędzie znacznie ułatwia konfigurację oraz redukuje konieczność okablowania obwodów elektrycznych.

27.3 SS2 – Safe Stop 2

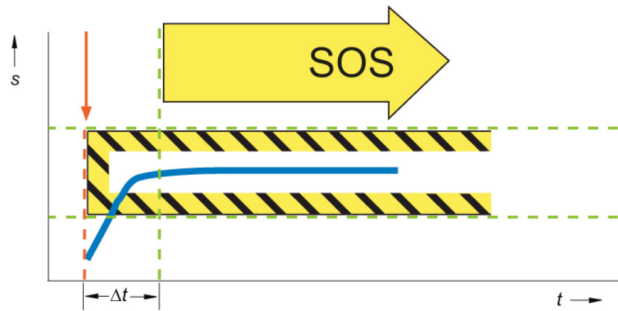
Funkcja bezpiecznego stopu 2, działa podobnie jak SS1 – jej przebieg możemy również podzielić na dwa etapy. W pierwszym następuje możliwie szybkie zatrzymanie napędu. W drugim zaś aktywowana zostaje funkcja SOS (Safe Operating Stop), która w przeciwieństwie do STO nie odcina siły generującej moment obrotowy. W drugiej fazie silnik zostaje więc unieruchomiony w określonej pozycji, której wartość jest monitorowana i utrzymywana przez pełen moment obrotowy silnika.



Stała kontrola pozycji bezpiecznej realizowana jest przez sterownik napędu. Dzięki temu, że zasilanie nie jest odłączane, restart silnika jest szybszy. W związku z powyższym funkcja SS2 stosowana jest zazwyczaj w urządzeniach gdzie zatrzymanie powinno zostać wykonane na krótki okres czasu bez wyłączenia zasilania. Tryb taki może być determinowany przez specyfikę procesu produkcyjnego bądź konieczność wykonywania częstych okresowych zadań konserwacyjnych, np. usuwanie zabrudzeń, nadmiaru materiału czy smarowanie komponentów mechanicznych.

27.4 SOS – Safe Operating Stop

Stosowana w przypadku bezpiecznego stopu 2 funkcja SOS odpowiada za bezpieczne monitorowanie stanu bezruchu napędu. Sterowanie zostaje nieprzerwanie aktywne – utrzymanie pozycji oraz doprowadzenie do zatrzymania układu realizowane jest przez system nadrzędny.

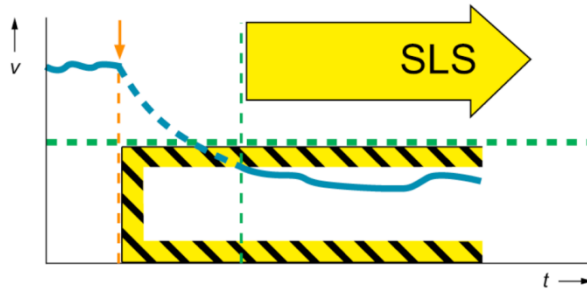


Po dezaktywacji funkcji SOS nie jest konieczna kalibracja maszyny lub bazowanie układu – praca osi może być natychmiast wznowiona bez zbędnego przestoju.

Funkcja znajduje zastosowanie np. w obrabiarkach, nawijarkach, prasach, windach czy urządzeniach pakujących.

27.5 SLS – *Safely-Limited Speed*

W wielu aplikacjach, stan niekoniecznie będzie wiązał się z całkowitym zatrzymaniem maszyny. Co więcej, może się zdarzyć iż ze względu na specyfikę produkcji lub jej efektywność, proces nie będzie mógł być zatrzymany. W takich przypadkach możemy określić warunki stanu bezpiecznego, który będzie sprowadzał się do ograniczenia prędkości pracy napędu oraz jej bezpiecznego monitorowania.

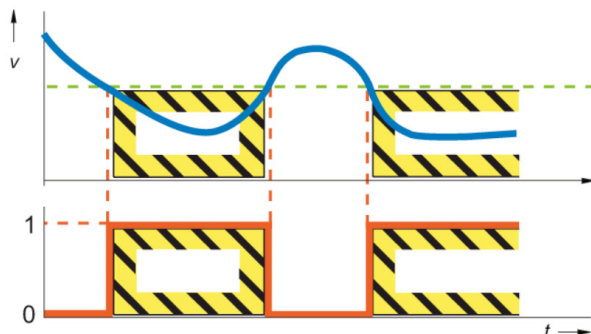


W razie przekroczenia zdefiniowanych warunków granicznych, napęd powinien przejść w stan określony na etapie konfiguracji. Może być to bezpieczne wyłączenie momentu lub zatrzymanie – zgodnie z opisem z poprzednich sekcji.

Funkcja stosowana jest w obszarach gdzie zagrożenie dla osób oraz mienia może zostać wyeliminowane przez ograniczenie prędkości pracy maszyny. Może wiązać się to z koniecznością wykonania konfiguracji urządzenia bądź czynności konserwacyjnych. Przykładem mogą być maszyny gdzie wymagana jest operacja ręczna, np. uzupełnienie materiału, zmiana narzędzia bądź podsuniecie detalu.

27.6 SSM – Safe Speed Monitor

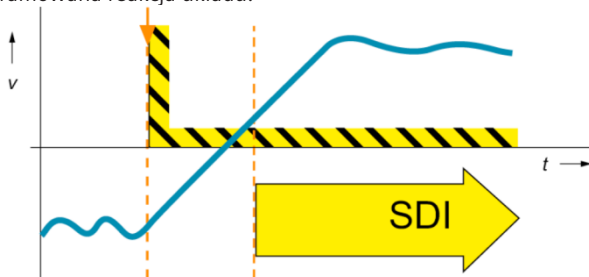
Funkcją pokrewną, niejako rozszerzającą SLS – jest funkcja monitorowania prędkości bezpiecznej (SSM). W przypadku gdy prędkość spadnie poniżej określonej wartości – system wygeneruje sygnał bezpieczeństwa. Przebieg prędkości oraz sygnału *fail-safe* zaprezentowany został na poniższym wykresie.



Wyjściowy sygnał bezpieczny (gwarantujący bezpieczną prędkość) może zostać wykorzystany w sterowniku programowalnym w celu wykonania odpowiedniej funkcji – np. zwolnienie blokady drzwi obszaru zamkniętego lub otwarcie pokrywy wirówki w celu napełnienia jej materiałem.

27.7 SDI – Safe Direction

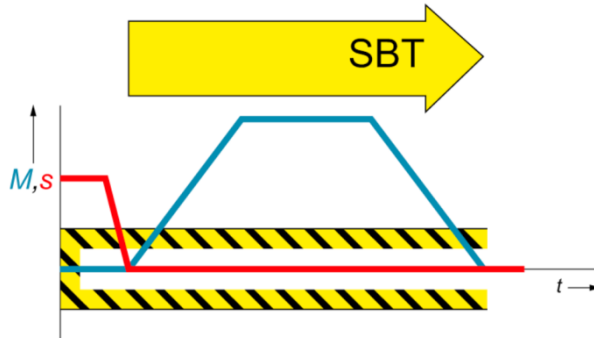
Od momentu aktywacji – funkcja bezpiecznego kierunku (SDI) gwarantuje, że oś napędowa będzie poruszać się tylko w jednym kierunku – zdefiniowanym jako niestanowiący zagrożenia. Kierunek pracy napędu monitorowany jest w sposób bezpieczny. W razie wykrycia zmiany kierunku – aktywowana zostaje zaprogramowana reakcja układu.



Przykładem mogą być maszyny, w których odpowiedni kierunek osi będzie gwarantował bezpieczeństwo operatorowi, np. w sytuacji gdzie konieczne będzie usunięcie brudu lub resztek materiału, a wsteczna praca maszyny nie pozwoli na wciągnięcie lub zgniecenie kończyny.

27.8 SBT – Safe Brake Test

Wśród funkcji bezpieczeństwa znajduje się również mechanizm cyklicznego testowania hamulca silnika lub hamulca zewnętrznego. W celu sprawdzenia sprawności hamowania, zwiększony zostaje moment obrotowy generowany przez napęd, podczas gdy hamulec jest załączony.



Okresowe testowanie układu hamującego daje pewność, że w razie potrzeby system zostanie sprawnie wyhamowany (funkcja SBC). Pozwala również na wykrycie zmian eksploatacyjnych, wymagających wykonania odpowiednich prac konserwacyjnych.

27.9 SLP – Safely-Limited Position

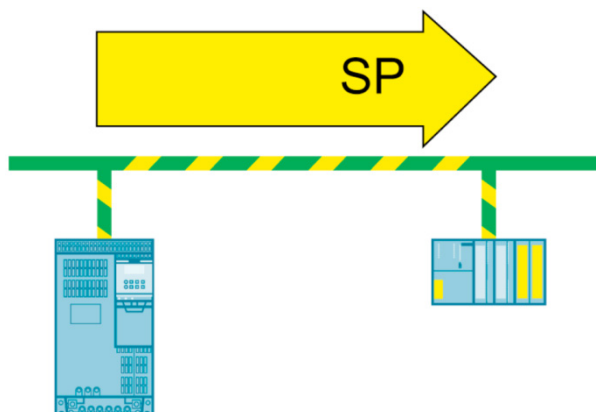
Bezpieczne monitorowanie pozycji (SLP) zapewnia, że oś porusza się jedynie w dopuszczalnym zakresie ruchu. Ograniczenie programowe pozycji zapewnia, iż w razie przekroczenia zdefiniowanej granicy – system zainicjuje odpowiednią procedurę wprowadzającą układ w stan bezpieczny. SLP umożliwia zazwyczaj przełączanie pomiędzy dwoma zdefiniowanymi zakresami – również podczas pracy systemu.



Funkcja może zostać zaaplikowana w obszarach zagrożenia dla personelu, gdzie ograniczenie pozycji ruchu osi zagwarantuje bezpieczeństwo podczas prac serwisowych bądź podczas uzupełniania lub usuwania materiału. Ograniczenie ruchu osi może dotyczyć przykładowo robotów, dźwigów lub suwnic.

27.10 SP – Safe Position

Funkcja SP służy do wyznaczenia pozycji, która zostanie zakwalifikowana jako bezpieczna. Wartość aktualna pozycji bezpiecznej zostaje przesłana do nadrzędnego sterownika F-PLC przez protokół *PROFIsafe*. W przeciwieństwie do SLP, funkcja ta służy do wyznaczenia bezpiecznego zakresu pracy układu, a nie do monitorowania pozycji aktualnej względem zdefiniowanych barier.



Dzięki funkcji SP jesteśmy w stanie, w bardzo elastyczny sposób, wyznaczyć bezpieczny zakres pracy skomplikowanych układów wieloosiowych bądź trajektorii krzywkowych. Możemy również wykonać implementację koncepcji wielowymiarowych stref bezpieczeństwa – bez konieczności stosowania dodatkowych czujników bądź enkoderów bezpiecznych.

27.11 Funkcje zintegrowane napędów SINAMICS

Funkcji bezpieczeństwa dedykowanych dla układów *Motion Control* jest więcej, aczkolwiek ich użyteczność ma stosunkowo wąskie pasmo zastosowań. Monitorowanie bezpiecznego ruchu osi może polegać również na analizie przyspieszenia, momentu obrotowego, przyrostu pozycji czy temperatury silnika. W zależności od wymogów konkretnej aplikacji – poziomu bezpieczeństwa, dostępnych parametrów czy skomplikowania układu – zastosujemy odpowiednie funkcje gwarantujące bezpieczeństwo.

Poniższa tabela prezentuje zintegrowane funkcje bezpieczeństwa dostępne w układach napędowych przykładowych klas produktowych z rodziny SINAMICS.

Grupa produktowa	Zintegrowane funkcje bezpieczeństwa
SINAMICS V90	STO
SINAMICS S110	STO, SS1, SS2, SOS, SBC, SLS, SSM, SDI
SINAMICS S120	STO, SS1, SS2, SOS, SBC, SBT, SLS, SSM, SDI, SLP, SP
SINAMICS G120	STO, SS1, SBC, SLS, SSM, SDI

28

Konfiguracja F-PLC w TIA Portal

W ostatniej sekcji opiszemy sposób realizacji funkcji bezpieczeństwa od strony programistycznej. W przypadku zastosowania prostego układu elektrycznego lub przekaźnika bezpieczeństwa, nie będzie konieczności uwzględnienia obwodów bezpieczeństwa w programie sterownika nadrzędnego, co więcej – w podstawowych aplikacjach może on w ogóle nie występować w strukturze projektu. W takich urządzeniach będziemy więc bazować na cechach obwodów elektrycznych (czujnik, stycznik, napęd, przekaźnik bezpieczeństwa, etc).

Kwestia zaaplikowania logiki oceny sygnałów bezpieczeństwa i wystawienia wyjść bezpiecznych pojawi się w układach wyższej klasy – np. przy programowalnych przekaźnikach bezpieczeństwa. Tutaj jednak zazwyczaj spotkamy się z uproszczonym, dedykowanym temu zadaniu oprogramowaniem konfiguracyjnym.

Bardziej zaawansowany projekt logiczny układu oceny bezpieczeństwa stworzyć możemy dla sterownika programowalnego PLC w wersji *fail-safe*. W przypadku środowiska inżynierskiego TIA Portal całość konfiguracji programu bezpieczeństwa wykonać możemy w jednym spójnym narzędziu inżynierskim. W zależności od rodziny sterowników F-PLC potrzebny będzie w tym celu pakiet inżynierski rozszerzony o STEP 7 Safety Basic (dla sterowników z rodziny S7-1200) lub o STEP 7 Safety Advanced (dla wszystkich aktualnie dostępnych sterowników *fail-safe*).

W niniejszym rozdziale rozważymy opisane uprzednio rozwiązania: oprogramowanie klasycznego układu bezpieczeństwa opartego o obwody elektryczne oraz komunikację bezpieczną z napędami przez protokół *PROFIsafe*. Każde z rozwiązań daje nam możliwość stworzenia aplikacji certyfikowanej pod kątem zgodności z normami bezpieczeństwa odpowiedniego poziomu.

28.1 Klasyczny układ elektryczny

Pierwsza z dostępnych dróg prowadzi do realizacji układu bezpieczeństwa (detekcja, ocena, reakcja) w oparciu o obwody elektryczne oraz wejścia/wyjścia sterownika programowalnego F-PLC.

28.1.1 Cechy projektu *Safety*

Zanim przejdziemy do sekcji konfiguracyjnej, zaznajomimy się z kilkoma pojęciami charakterystycznymi dla programowalnych układów bezpieczeństwa, które pozwolą łatwiej zrozumieć ideę pracy sterownika klasy *fail-safe*.

Zacznijmy od faktu, iż należy mieć świadomość, że tworzenie aplikacji bezpieczeństwa w oparciu o sterownik F-PLC jest zadaniem dosyć specyficznym i różni się od podejścia standardowego – zarówno od strony wymaganych urządzeń jak i programu sterowania.

Konfiguracja sprzętowa

W celu stworzenia nadrzędnego układu oceny bezpieczeństwa, należy stosować certyfikowane komponenty sprzętowe sterownika. Przede wszystkim użyta musi być jednostka centralna F-CPU. Wszystkie obwody bezpieczeństwa – zarówno wejściowe jak i wyjściowe muszą być realizowane przez moduły F-I/O.

Struktura projektu

Program bezpieczeństwa realizowany jest w niezależnym bloku organizacyjnym – tzw. grupa *F-runtime* (F-OB). Jest to blok przerwania czasowego (w procesorach S7-1200F/1500F), który cyklicznie wywołuje program bezpieczeństwa. W grupie tej wywoływane są wszystkie bloki bezpieczeństwa – zarówno te utworzone przez użytkownika jak i wygenerowane systemowo. W jednym sterowniku możemy stworzyć maksymalnie dwie grupy *F-runtime*.

W bloku organizacyjnym F-OB wywoływany jest główny blok programu bezpieczeństwa F-FB wraz z instancją bloku danych F-DB.

Blok *Main safety* (F-FB)

Generowany automatycznie blok programowy, w którym zlokalizowany jest program bezpieczeństwa. W bloku można pisać kod programu lub wykorzystać go jako miejsce wywoływania podprocedur bezpieczeństwa (segmentacja kodu). Pozostałe bloki programowe mogą być tworzone przez użytkownika (F-FB/F-FC/F-DB) lub generowane przez system (F-SB, F-shared DB F-I/O DB, F-I-DB), ewentualnie inne certyfikowane bloki (dostępne z biblioteki systemowej) służące do obsługi konkretnych funkcji bezpieczeństwa (np. ESTOP). Bloki F wyróżnione są w projekcie kolorem żółtym.

Safety program

Program bezpieczeństwa użytkownika. Algorytm sterowania, który uwzględnia nadrzędność zadziałania podzespołów bezpieczeństwa.

Tryb Safety

Tryb pracy F-CPU, w którym program bezpieczeństwa przetwarzany jest dwutorowo. W trybie tym, wszystkie procedury testu obwodów wejść/wyjść, jakości przetwarzania programu przez F-CPU są aktywne. W trybie Safety nie można wykonywać modyfikacji programu bezpieczeństwa, może być przetwarzany tylko program pochodzący ze spójnej, całościowej (sprzęt i program) kompilacji. Użytkownik może wyłączyć tryb Safety w F-CPU.

Stan bezpieczny

Podstawowe założenie koncepcji bezpieczeństwa, mówiące o tym, że w systemie bezpieczeństwa (*fail-safe*) istnieje – dla wszystkich zmiennych procesowych – stan bezpieczny. Dla wejść/wyjść cyfrowych (czujniki oraz elementy wykonawcze) wg normy IEC jest to stan „0” (odłączenie od zasilania).

Pasywacja

Wyłączenie z przetwarzania przez system *fail-safe* obszarów obrazu procesu, które zostały oznaczone przez system diagnostyki jako będące w stanie awarii. Są trzy typy pasywacji: kanału (awaria w kanale), modułu (awaria całego modułu) i komunikacji (brak komunikacji z modułem lub urządzeniem w sieci *PROFIsafe*). W obrazie procesu sterownika w miejsce wartości procesowej wpisywany jest gwarantowany (bezpieczny) stan „0”.

Przykładowo – jeśli ocena układu bezpieczeństwa w F-CPU ustawia stan wysoki wyjścia cyfrowego (logiczna „1”) – pasywacja (wykryty błąd kanału/modułu/komunikacji) spowoduje ustawienie wyjścia na wartość logiczną „0”.

Depasywacja (reintegracja)

Proces odwrotny do pasywacji. Po usunięciu awarii system F sygnalizuje gotowość do aktywacji przetwarzania pasywowanych kanałów/modułów/lurządzeń. Włączenie następuje po skwitowaniu awarii (automatyczne lub ręczne).

28.1.2 Ustawienia sprzętowe (Device configuration)

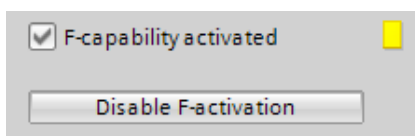
Opis konfiguracji projektu bezpieczeństwa w środowisku TIA Portal rozpoczniemy od omówienia parametrów sprzętowych elementów sterownika F-PLC. W systemach takich kluczowe będą dla nas atrybuty samej jednostki centralnej, a także modułów wejść/wyjść binarnych.

Dodanie jednostki centralnej, modułów I/O bądź urządzeń pracujących w strukturze zdecentralizowanej odbywa się tak samo jak w przypadku jednostek standardowych.

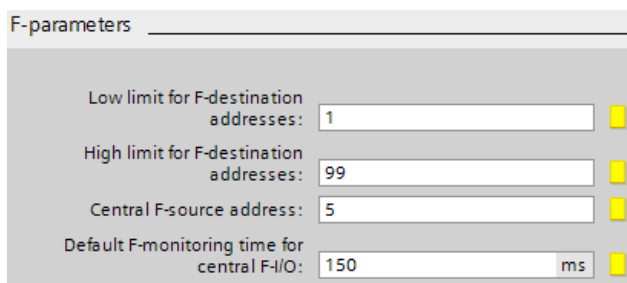
Wszystkie parametry sterownika istotne z punktu widzenia programu bezpieczeństwa oznaczone są w projekcie żółtym znacznikiem.

Parametry F-CPU

Pierwszym krokiem w konfiguracji sterownika F-PLC jest sprawdzenie czy aktywny jest tryb Safety (Device configuration → General → Fail-safe → F-activation).



Następnie ustawiamy adres *PROFIsafe* mastera (F-CPU) oraz konfigurujemy dopuszczalną przez jednostkę nadrzędną (*F-source*) adresację modułów I/O (*F-destination*).



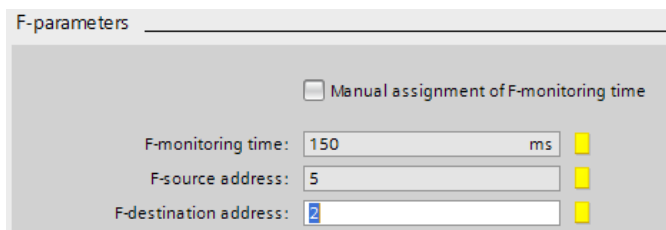
Adresacja *PROFIsafe*

Komunikacja *PROFIsafe* bazuje na dwóch typach adresów: adresach źródłowych (*F-source address*) nadawanych masterom komunikacji (*F-Host*) oraz adresach urządzeń (*F-destination address*) nadawanych wszystkim podrzędnym urządzeniom bezpieczeństwa znajdującym się w podsieci PN (lub DP).

UWAGA – każdy moduł F-I/O (również ulokowany bezpośrednio przy F-CPU) musi posiadać unikalny adres *PROFIsafe*! Adresy *PROFIsafe* F-CPU muszą być unikalne w globalnej sieci systemu, natomiast adresy modułów F-I/O muszą być unikalne w obrębie podsieci F-CPU.

Parametry F-I/O

W związku z powyższym – kolejnym krokiem jest przypisanie adresów *PROFIsafe* w ustawieniach modułów wejść/wyjść.



W karcie ustawień F-parameters modułów I/O ustawiamy również m.in. czas monitorowania sygnałów w programie F-CPU (F-monitoring time), a także tryb pasywacji (kanał/moduł) czy reintegracji (automatycznie/ręcznie).

Po wgraniu do sterownika tak przygotowanej konfiguracji sprzętowej, należy jeszcze przypisać adresy *PROFIsafe* do urządzeń w trybie on-line. Procedura działa podobnie jak przypisywanie nazw urządzeń w sieci PROFINET I/O. W konfiguracji sprzętowej z menu podręcznego wybieramy opcję  **Assign PROFIsafe address**, a następnie w oknie konfiguracyjnym przypisujemy adresy modułów sieci *PROFIsafe*.

W kroku kolejnym przechodzimy do parametryzacji kanałów modułów I/O.

Parametry F-DI

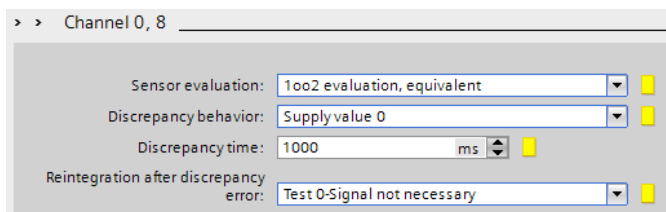
Poniżej przedstawimy parametry istotne z punktu widzenia konfiguracji sprzętowej *fail-safe* urządzeń wejściowych.

Zasilanie czujników

W przypadku wejść cyfrowych – po pierwsze określamy, czy wewnętrzny zasilacz zostanie wykorzystany do zasilania czujników (*Sensor supply*), ew. decydujemy dla której grupy kanałów.

Ewaluacja sygnałów wejściowych

W kolejnym kroku charakteryzujemy typ oraz połączenie sygnałów z aktywatorów układu bezpieczeństwa. Zgodnie z wcześniejszym opisem – w zależności od wymaganego poziomu bezpieczeństwa – czujniki możemy podłączyć jedno- lub dwutorowo. W ustawieniach modułu F-DI będziemy definiować właśnie tę charakterystykę obwodów wejściowych.



Do wyboru mamy dwa tryby:

- ewaluacja 1oo1 (1 out of 1) – układ, w którym czujnik podłączony zostaje do jednego kanału modułu wejść F-DI
 - ewaluacja 1oo2 (1 out of 2) – połączenie, w którym dwa kanały wejściowe modułu F-DI zostają zajęte przez:
 - sygnał z czujnika dwutorowego (ze stykami zgodnymi bądź przeciwnymi – *equivalent/non-equivalent*)
- lub
- dwa czujniki jednokanałowe mierzące tę samą wartość.

W trybie ewaluacji 1oo2 sygnały wejściowe porównywane są wewnętrznie w celu określenia ich zgodności (lub przeciwstawności).

Zwróćmy również uwagę, że w tym trybie dwa kanały fizyczne zostają połączone w jeden kanał logiczny. Taka metoda oceny wejściowego układu bezpieczeństwa pozwala osiągnąć najwyższą klasę bezpieczeństwa, jednak wymaga to zastosowania zdwojonej ilości fizycznych kanałów wejściowych. Przykładowo moduł 16xF-DI może obsłużyć 16 kanałów wejściowych układu bezpieczeństwa poziomu SIL2/PLD lub 8 kanałów na poziomie SIL3/PLe.

W wyniku wewnętrznej ewaluacji (zgodności lub przeciwności) – do obrazu procesu sterownika zapisywana jest wartość logiczna kanału o niższym poziomie w parze. Jeśli więc dwutorowy czujnik podłączony jest do kanału 0/8 (Channel 0,8) – w programie użytkownika odwoływać się będziemy do fizycznego adresu wejścia kanału 0.

Analiza rozbieżności (Discrepancy analysis)

Wykorzystując pomiar z czujnika podłączonego dwutorowo lub z dwóch czujników służących do pomiaru tej samej wartości procesowej – należy przyjąć, że wyniki pomiarów (bądź detekcji zdarzenia), zostaną przesłane do modułu F-DI z drobną rozbieżnością w czasie. Dzieje się tak ze względu na ograniczoną dokładność takiego układu pomiarowego.

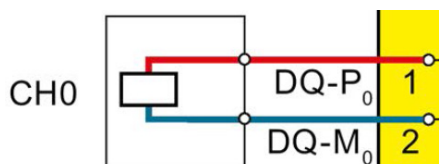
Analiza rozbieżności dostarcza do systemu oceny bezpieczeństwa informacji na temat błędów czujnika na podstawie różnicy czasu odczytu dwóch sygnałów o tej samej funkcji.

Analiza zostaje rozpoczęta w momencie, gdy odczytane stany dwóch wejść (określonego kanału) przyjmą wartość odmienną niż ta zdefiniowana w konfiguracji sprzętowej. Innymi słowy (dla ewaluacji 1oo2 typu *equivalent*) – system oczekuje tego samego stanu logicznego na obu fizycznych kanałach. Jeśli dostarczone sygnały różnią się – rozpoczyna się test, który trwa tak długo jak zostało to określone w konfiguracji kanału (*Discrepancy time*). Jeśli do końca testu stan porównania nie zmieni się – zgłoszony zostaje błąd rozbieżności (*discrepancy error*).

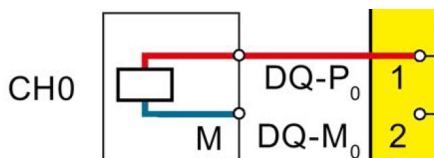
Parametry F-DO

Dla wyjść cyfrowych typu *fail-safe* należy zdefiniować tryb pracy:

Załączanie PM:



Załączanie PP:



Każdy kanał karty F-DO posiada dwa przyłącza zasilające obciążenie. Wybór sprowadza się więc do trybu zasilania elementu wykonawczego oraz jego uziemienia.

Więcej informacji oraz przykłady aplikacji dla danego trybu pracy odszukać możemy w dokumentacji modułów F-DO:

<https://support.industry.siemens.com/cs/us/en/view/109482427>

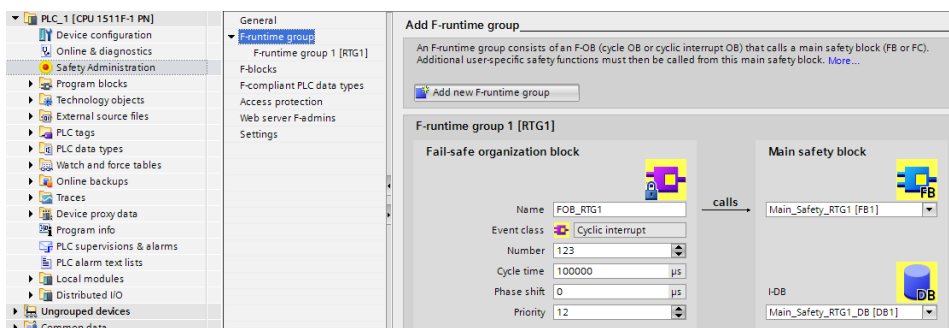
Dodatkowo w ustawieniach wyjść F – parametryzujemy funkcje testujące kanały wyjściowe (dark test/light test – więcej informacji w powyższej dokumentacji lub w plikach pomocy systemowej TIA Portal). Mamy również sposobność aktywacji funkcji diagnostycznych oraz określenia trybu reintegracji kanału.

28.1.3 Program Safety

W dalszej części skupimy się na konfiguracji programowej części projektu układu oceny bezpieczeństwa. Podstawowym edytorem charakterystycznym dla programu bezpieczeństwa jest narzędzie administracyjne.

Safety Administration

Edytor służy m.in. do wyświetlania statusu aplikacji bezpieczeństwa oraz sygnatury zbiorczej programu, organizacji grup F-runtime czy konfiguracji poziomów dostępu. Możemy również w przejrzysty sposób wykonać parametryzację bloku organizacyjnego *fail-safe* oraz powiązać z nim główny blok programu Safety.



Sygnatura kolektywna

Istotnym parametrem projektu bezpieczeństwa jest sygnatura programu (*Collective F-signature*). Jest to swego rodzaju suma kontrolna, czy też baza danych typu *fail-safe*, o ustandaryzowanej strukturze wewnętrznej. Wynik kompilacji projektu sterownika oraz programu podsumowany jest właśnie taką sygnaturą. Zostaje ona przypisana do konkretnego modułu lub urządzenia typu *fail-safe*. Znacznik ten przedstawia bieżący stan układu od strony programowej – służy on do sygnalizacji awarii modułów i urządzeń oraz ich reintegracji. Jeśli układ bezpieczeństwa znajduje się w poprawnym

stanie pracy oraz projekt został bezbłędnie wgrany do sterownika PLC – wartość sygnatury *on-line* oraz *off-line* powinna być zgodna. Jeśli nie jest – należy spróbować wgrać projekt ponownie.

Bloki danych modułów F-I/O

Przeglądając strukturę projektu sterownika bezpieczeństwa – wśród systemowych bloków danych natkniemy się na bloki danych F-I/O DB, czyli kolejną cechę charakterystyczną dla systemów F.

F-I/O data blocks	
	F00000_F-DI16x24VDC_1 [DB30002]
	F00009_F-DQ8x24VDC/2APPM_1 [DB30003]
	F00016_F-DI8x24VDC/2APPM_1 [DB30011]
	F00022_F-DQ4x24VDC/2APPM_1 [DB30012]
	F00027_F-PM-E24VDC/8APPMST_1 [DB30013]

W konfiguracji F-PLC każdy moduł wejść/wyjść bezpiecznych posiada swój własny blok danych. Zawartość takiego bloku danych zasilana jest przez system informacjami statusowymi w zakresie stanu bezpieczeństwa. Interfejs wejściowy pozwala z kolei na konfigurację modułu (np. aktywację manualnej depasywacji) oraz jego reintegrację po pasywacji.

Pasywacja

Zgodnie z informacją ze wstępu do rozdziału – wiemy już, że system przechodzi w stan pasywacji jeśli wykryta zostanie usterka – może dotyczyć ona kanału, modułu lub komunikacji. Awaria taka sygnalizowana jest błędem widocznym w buforze diagnostycznym sterownika, a także czerwoną diodą na danym komponencie sprzętowym. Pasywacja wprowadza obwód elektryczny w stan bezpieczny.

Jeżeli karta F-I/O skonfigurowana została w taki sposób, iż podczas błędu pasywowany zostaje cały moduł – informacja o stanie pasywacji dostępna jest w zmiennej PASS_OUT dostępnej w bloku F-I/O-DB. Jeśli pasywacja dotyczy jedynie kanału, informację na temat pasywacji odczytać należy z przestrzeni adresowej interfejsu modułu F-I/O.

Zmienna QBAD informuje, że wyjścia zostały ustawione w stan bezpieczny.

Reintegracja

Przywrócenie modułu lub kanału do pracy normalnej po pasywacji wymaga przede wszystkim usunięcia usterki powodującej błąd.

Jeśli w ustawieniach kanału zdefiniowana została depasywacja automatyczna – usunięcie przyczyny awarii będzie wystarczające do przywrócenia pracy normalnej układu.

Jeśli jednak reintegracja dla kanału (bądź modułu) zdefiniowana została jako manualna – po usunięciu usterki będzie wymagane dodatkowo skwitowanie błędu. Potwierdzenie może zostać wykonane na dwa sposoby:

- przez zmienną z bloku F-I/O-DB. Podanie zbocza narastającego na wejście ACK_REI spowoduje reintegrację modułu.

F00000_F-DI16x24VDC_1	
	Name
1	Input
2	PASS_ON
3	ACK_NEC
4	ACK_REI
5	IPAR_EN
6	Output
7	PASS_OUT
8	QBAD
9	ACK_REQ
10	IPAR_OK
11	DIAG
12	InOut
13	Static

- przez globalną funkcję `ACK_GL` – dostępną z biblioteki funkcji systemowych *fail-safe* (o czym w dalszej części).

UWAGA – reintegracja manualna modułów S7-1500F/ET 200MP może zostać skonfigurowana przez ich parametry w środowisku inżynierskim. W przypadku systemów rozproszonych wejść/wyjść ET 200SP – reintegrację manualną należy aktywować przez ustawienie w programie użytkownika zmiennej `ACK_NEC` na wartość „1”. Domyślnie reintegracja odbywa się bowiem w trybie automatycznym.

Reintegracja manualna może zostać wykonana, gdy usterka powodująca pasywację została usunięta, a co za tym idzie status modułu F-I/O (wyjście `ACK_REQ` z bloku F-I/O DB) został ustawiony na wartość „1”.

Więcej informacji na temat pasywacji oraz reintegracji znaleźć można w następującym przykładzie aplikacji:

<https://support.industry.siemens.com/cs/ww/en/view/22304119>

Programowanie

Program bezpieczeństwa składa się z kilku bloków *fail-safe*, które tworzone są w edytorach F-FBD lub F-LAD oraz bloków *fail-safe* generowanych automatycznie przez system.

Należy mieć na uwadze podstawowe różnice pomiędzy programowaniem sterownika standardowego, a sterownika bezpieczeństwa:

- do wejść EN i wyjść ENO funkcji bezpieczeństwa nie można podłączać żadnych sygnałów (w programie nie można z nich korzystać),
- ograniczona ilość instrukcji,
- ograniczenie użycia typów danych oraz przestrzeni adresowych,
- sygnały, symbole oraz funkcje *fail-safe* podświetlane są w edytorach F-LAD/F-FBD na żółto.

Wymiana informacji pomiędzy obrazem procesu standardowego i Safety

W obszarze programowania natkniemy się na więcej ograniczeń mających na celu zabezpieczenie, odizolowanie oraz ustandaryzowanie programu bezpieczeństwa. Poniższa tabela prezentuje możliwe kombinacje zapisu oraz odczytu wejść/wyjść z przestrzeni adresowej programu standardowego w odniesieniu do programu bezpieczeństwa.

		PROGRAM STANDARDOWY		PROGRAM SAFETY	
		ODCZYT	ZAPIS	ODCZYT	ZAPIS
STANDARDOWY OBRAZ PROCESU	WEJŚCIA	✓	✓	✓	✗
	WYJŚCIA	✓	✓	✗	✓
OBRAZ PROCESU SAFETY	WEJŚCIA	✓	✗	✓	✗
	WYJŚCIA	✓	✗	✗	✓

Wymiana informacji pomiędzy blokami danych standardowymi i Safety

Analogiczna sytuacja występuje w przypadku bloków danych F oraz ST. W tym wypadku jednak, jedynym (aczkolwiek istotnym) niuansiem jest brak możliwości zapisu informacji do bloków danych F z poziomu standardowego programu użytkownika.

		PROGRAM STANDARDOWY		PROGRAM SAFETY	
		ODCZYT	ZAPIS	ODCZYT	ZAPIS
STANDARDOWY DB		✓	✓	✓	✓
BLOK DANYCH F		✓	✗	✓	✓

Funkcje interfejsu użytkownika

Łatwo zauważyć, że funkcje bezpieczeństwa sterownika F-PLC realizowane są w głównej mierze na poziomie sprzętu. Przedstawiona już ewaluacja czujników dwutorowych, funkcje testujące, diagnostyka kanałów czy też tryb pasywacji/reintegracji – to parametry konfiguracyjne hardware'u.

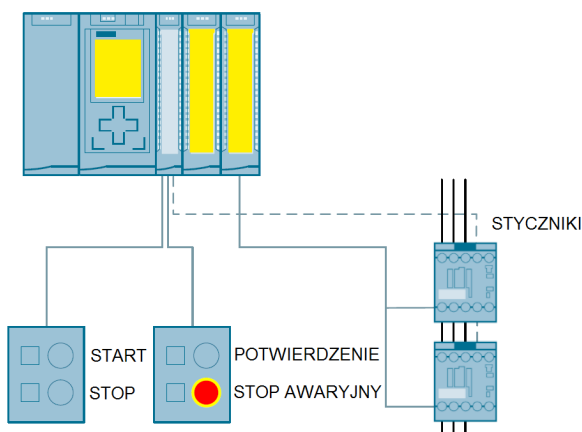
W związku z powyższym w sterowniku wystarczą nam już bardzo proste instrukcje, które pozwolą zaprogramować dodatkowo logikę układu oceny bezpieczeństwa. I w zasadzie tak właśnie jest – do dyspozycji mamy ograniczoną ilość podstawowych komend programowych języka F-FBD oraz F-LAD. Niemniej jednak w obszarze instrukcji systemowych *fail-safe*, znajdziemy kilka gotowych bloków funkcyjnych,

które mogą ułatwić nam programowanie, a także uprościć oraz ustandaryzować strukturę projektu bezpieczeństwa.

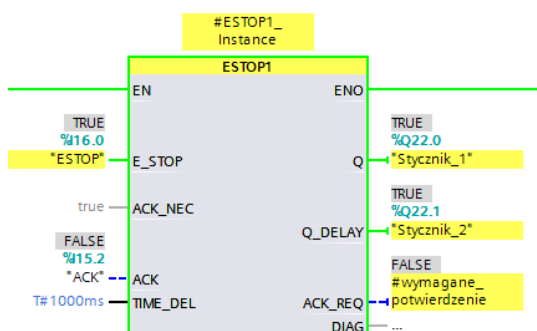
Safety functions	
ESTOP1	Emergency STOP/emergency OFF up to stop category 1
TWO_H_EN	Two-hand monitoring with enable
MUT_P	Parallel muting
EV1oo2DI	1oo2 evaluation with discrepancy analysis
FDBACK	Feedback monitoring
SFDOOR	Safety door monitoring
ACK_GL	Global acknowledgment of all F4I/Os in an F-runtime group

Pomijając standardowe instrukcje programistyczne – poniżej znajdziemy krótki opis funkcji predefiniowanych (S7-1500F) oraz przykłady aplikacyjne, które można wykorzystać jako gotowe rozwiązanie projektowe:

- **ESTOP** – obsługa przycisku wyłącznika awaryjnego.
Funkcja służy do realizacji wyłączenia (*emergency OFF*) lub zatrzymania (*emergency STOP*) awaryjnego (opcjonalnie z potwierdzeniem). Zatrzymanie kategorii 0 oraz 1 – opisane w poprzednim rozdziale.



Przykładowa parametryzacja bloku widoczna jest na poniższym zrzucie ekranu.



Fizyczne wejście (ewaluacja 1oo1 lub 1oo2) podłączone zostaje pod *E_STOP*. Wejście *ACK_NEC* = true oznacza konieczność potwierdzenia (wartość domyślna). Sygnał potwierdzający podłączamy pod parametr wejściowy *ACK*. Dodatkowo mamy możliwość skonfigurowania dwóch wyjść, które zostaną rozłączone w zdefiniowanym odstępie czasu – *TIME_DEL*. Możliwość określenia czasu rozłączenia wyjścia pomocniczego pozwala na realizację funkcji SS1 opisaną w poprzednim rozdziale (zatrzymanie kontrolowane, a następnie odłączenie zasilania po wyhamowaniu). Omówiliśmy również cechy układu zwłocznego, gdzie w celu osiągnięcia poziomu SIL3/PLe należy zastosować dwa styczniki. Ze względów bezpieczeństwa sprzętu, odłączenie silnika od przekształtnika powinno odbywać się po odłączeniu falownika od sieci.

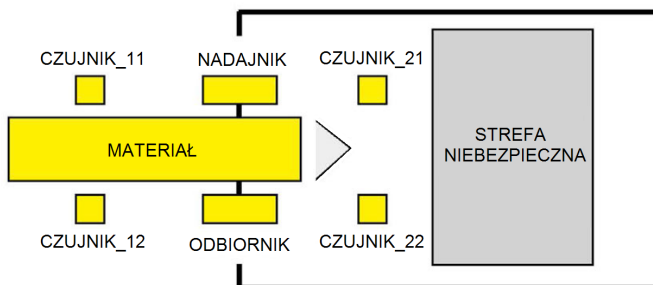
W powyższym przykładzie zdjęcie sygnału 1 (wciśnięcie przycisku awaryjnego) spowoduje natychmiastowe odłączenie zasilania od stycznika, a po 1 sekundzie odłączenie drugiego stycznika. Na koniec procedury (oraz przywróceniu stanu „1” na wejściu *E_STOP*) – status wymogu potwierdzenia *ACK_REQ* ustawiony zostanie na wartość „1”. Potwierdzenie (zbocze narastające na wejściu *ACK*) spowoduje ponowne zasilenie wyjść *Q* oraz *Q_DELAY*.

Przykład aplikacji stopu awaryjnego dostępny jest poniżej:

<https://support.industry.siemens.com/cs/pl/en/view/21064024>

- **TWO_H_EN** – operacja dwuręczna.

Pulpity oburęczne służą do wymuszenia na operatorze trzymania obu dłoni na przyciskach aktywacyjnych, które znajdują się poza strefą zagrożenia, np. pod osłonami lub w miejscu odpowiednio oddalonym. Uruchomienie maszyny możliwe jest tylko przez jednoczesne wciśnięcie dwóch przycisków (z określonym czasem rozbieżności). Zwróćmy również uwagę, iż przypadkowe naciśnięcie jednego przycisku nie spowoduje uruchomienia urządzenia.



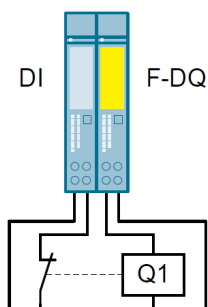
Od strony programowej definiować będziemy czas rozbieżności wykrycia obecności materiału przez pary czujników przed oraz za kurtyną świetlną oraz maksymalny dozwolony czas mutingu. Wśród analizowanych przez funkcję systemową parametrów, znajdują się również – aktualny stan modułu F-I/O (QBAD), a także status kurtyny świetlnej oraz transportera.

Więcej informacji oraz szczegółowy opis systemowej funkcji realizującej powyższe zadanie dostępny jest w dokumentacji układów F lub w plikach pomocy TIA Portal.

Przykład aplikacji mutingu dostępny jest poniżej:

<https://support.industry.siemens.com/cs/pl/en/view/58793869>

- **FDBACK** – funkcja monitorowania sprzężenia zwrotnego.
W aplikacjach wymagających osiągnięcia wysokiego poziomu bezpieczeństwa SIL/PL kluczowe jest monitorowanie poprawności działania elementów wykonawczych. Typowym przykładem jest sprawdzanie stanu stycznika przez odczyt stanu styków pomocniczych – ilustracja poniżej.

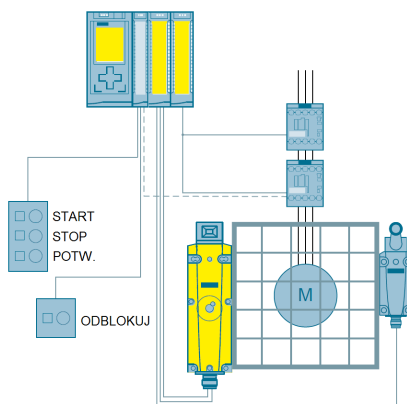


Takie sprzężenie zwrotne pozwala na osiągnięcie w układzie bezpieczeństwa wysokiego pokrycia diagnostycznego. Jest to szczególnie istotne w przypadku konfiguracji redundantnej (dwa styczniki załączające napęd). Jeśli styki jednego ze styczników zostaną zespawane, nie zostanie to zauważone przez użytkownika, a co za tym idzie system dwukanałowy stanie się systemem jednokanałowym. Stosując funkcję monitorowania styków pomocniczych, wykryjemy niepoprawne zachowanie stycznika, co pozwoli zapobiec ponownemu włączeniu urządzenia do momentu usunięcia usterki.

Przykład aplikacji monitorowania sprzężenia zwrotnego dostępny jest poniżej:

<https://support.industry.siemens.com/cs/pl/en/view/21331098>

- **SFDOOR** – funkcja monitorowania drzwi ochronnych.
Aby uniemożliwić dostęp do strefy zagrożenia podczas pracy maszyny, stosowane są drzwi ochronne, a ich położenie monitorowane jest przez system bezpieczeństwa. Ze względu na zagrożenie wynikające z pracy elementów maszyny, drzwi ochronne są zablokowane i można je otworzyć tylko po zatrzymaniu maszyny. Otwarcie drzwi podczas pracy urządzenia powoduje jego zatrzymanie lub przejście w tryb pracy bezpiecznej (np. SLS).

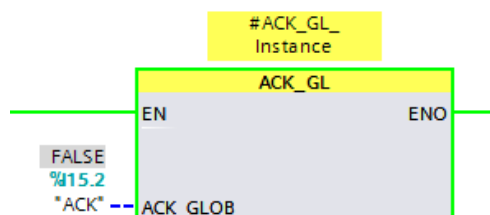


W celu osiągnięcia wymaganego poziomu bezpieczeństwa – monitorowanie położenia drzwi ochronnych powinno odbywać się przy pomocy dwóch czujników położenia (krańcówki, rygle). Jeśli jeden wyłącznik pozycyjny ulegnie awarii, otwarcie drzwi ochronnych zostanie wykryte przez drugi czujnik. Niepoprawna praca jednego z detektorów, wykrywana jest przez system oceny bezpieczeństwa (F-PLC) na podstawie analizy rozbieżności czasu (*discrepancy analysis*). Ponowne uruchomienie układu zablokowane jest do momentu przywrócenia sprawności wszystkich elementów (np. wymiana czujnika pozycyjnego).

Przykład aplikacji monitorowania drzwi ochronnych dostępny jest poniżej:

<https://support.industry.siemens.com/cs/pl/en/view/21063946>

- **ACK_GL** – globalne potwierdzenie błędów układu bezpieczeństwa. Funkcja służy do równoczesnej reintegracji wszystkich błędów modułów F (lub kanałów) po ich depasywacji.



Błędy prowadzące do depasywacji systemu wynikać mogą z awarii kanału, modułu lub z problemów komunikacyjnych.

Dokumentacja

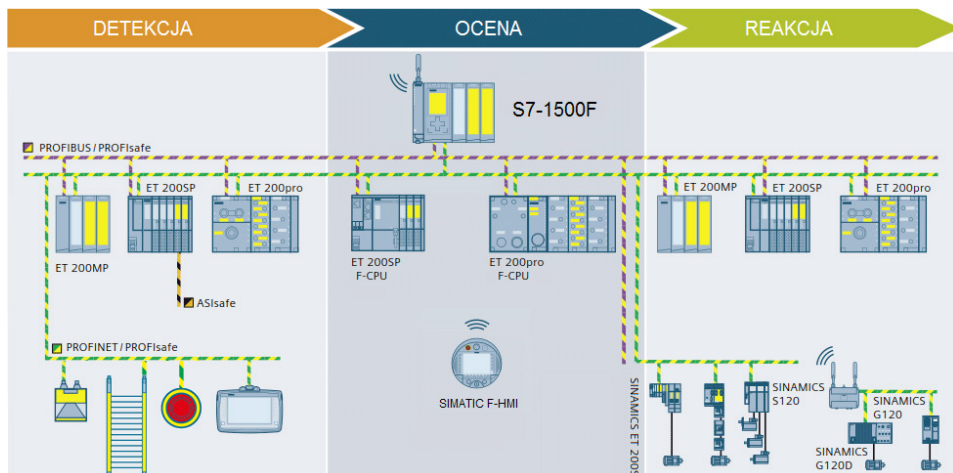
Kompletny kurs konfiguracji projektu sterownika F-PLC w środowisku STEP 7 Safety Advanced (w języku polskim) dostępny jest w poniższej lokalizacji:

publikacje.siemens-info.com

Dokument ten jest bardzo przystępnym wstępem do konfiguracji sterownika bezpieczeństwa, nawet dla osób niemających doświadczeń z podobnymi układami. W środowisku TIA Portal stworzenie takiego projektu jest łatwe oraz szybkie, a także spójne ze standardowym projektem sterownika.

28.2 Komunikacja PROFI-safe

Nowoczesną alternatywą dla klasycznego układu bezpieczeństwa, gdzie urządzenia wykrywające oraz reagujące na sygnały bezpieczeństwa podłączone są do sterownika F-PLC elektrycznie (przez sygnały I/O) – jest komunikacja przez sieć przemysłową. W związku z koniecznością zapewnienia zgodności z normami bezpieczeństwa nie może być to jednak standardowa sieć komunikacyjna. Dlatego w celu obsługi bezpiecznej wymiany danych (sumy kontrolne, priorytety ramek komunikacyjnych, etc.) powstał profil komunikacyjny *PROFI-safe*. Protokół taki może zostać narzucony na sieć PROFINET (również IWLAN) lub PROFIBUS, a w połączeniu z bezpieczną siecią ASIsafe (dedykowaną dla czujników/aktuatorów) utworzyć możemy spójną strukturę komunikacyjną systemu bezpieczeństwa. Przykład tak skonfigurowanego systemu widać na poniższym schemacie.



PROFIsafe

Profil komunikacyjny *PROFIsafe* wykorzystuje standardowe usługi niskiego poziomu (w ramach systemu usług magistrali) aby zapewnić pewną oraz szybką komunikację. Podczas transmisji danych, *PROFIsafe* wykorzystuje dodatkowe algorytmy testujące, aby odpowiednio filtrować oraz monitorować dane pod kątem bezpieczeństwa:

- cykliczna numeracja kolejnych paczek danych,
- monitorowanie czasu odpowiedzi (*watchdog*),
- sprawdzenie adresów *PROFIsafe*,
- obliczenie sumy kontrolnej ramki CRC (*Cyclic-Redundancy-Check*).

Komunikacja *fail-safe*, bazująca na wielu protokołach komunikacyjnych w ramach jednej magistrali, udowodniła swoją przydatność w tysiącach rozwiązaniach opartych na sieci przemysłowej PROFIBUS oraz PROFINET. Trzeba mieć na uwadze, iż każde urządzenie pracujące z wykorzystaniem profilu *PROFIsafe* musi obsługiwać komunikację w tym standardzie. Dobierając więc komponenty układu (czujnik, sterownik, system I/O, panel HMI, napęd, etc.) należy sprawdzić ich funkcjonalność w zakresie komunikacji bezpiecznej.

Przykładowe produkty z portfolio SIEMENS wspierające komunikację bezpieczną *PROFIsafe* to przyciski SIRIUS ACT PN (F), wyłączniki pozycyjne, sterowniki SIMATIC (F), panele HMI (F), napędy (G/S) czy systemy rozproszonych wejść/wyjść ET 200.

W opisywanej konfiguracji interesować nas będzie zagadnienie połączenia bezpiecznego *PROFIsafe* pomiędzy F-PLC SIMATIC a napędem SINAMICS. W zależności od zastosowanego w projekcie przekształtnika częstotliwości będziemy mieli sposobność wykonania tego zadania przez odpowiednie narzędzie inżynierskie.

Ogólnie, zadanie sprowadza się do rozszerzenia telegramu komunikacyjnego o ramkę *PROFIsafe*, która będzie zawierać niezbędne informacje z punktu widzenia funkcji bezpieczeństwa. Zgodnie z powyższym opisem – ta wymiana danych będzie traktowana przez system z odpowiednim priorytetem, a także monitorowana pod kątem wspomnianych parametrów.

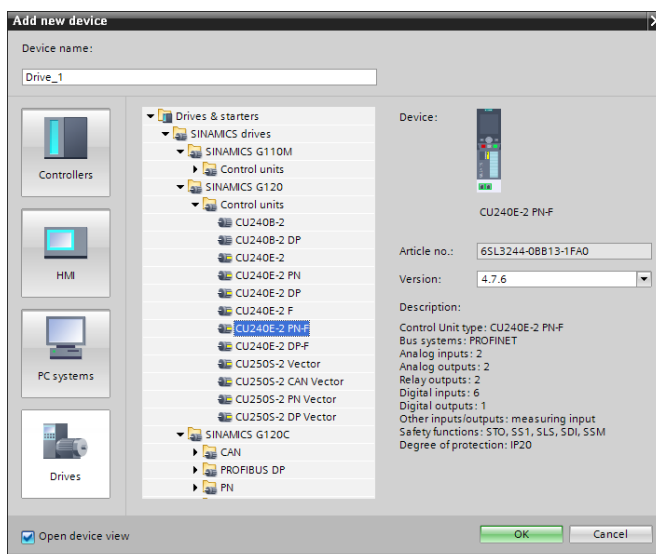
28.2.1 Konfiguracja w TIA Portal Startdrive

Z poprzedniej sekcji wiemy już jak wykonać konfigurację sterownika F-PLC przez STEP 7 Safety Advanced. Jeśli naszym urządzeniem wykonawczym będzie napęd standardowy z rodziny SINAMICS G – komunikacja *fail-safe* może zostać zestawiona spójnie w środowisku TIA Portal. Do obsługi napędów wymagany jest pakiet Startdrive, który można pobrać bezpłatnie na stronach wsparcia technicznego Siemens.

Ze względu na fakt, iż opisywany w niniejszej dokumentacji napęd serwo SINAMICS V90 nie wspiera komunikacji *PROFIsafe*, jako przykład rozważymy tutaj napęd G120.

Dodanie napędu do projektu

Rozbudowując wcześniej utworzony projekt sterownika F, możemy dołączyć do niego napęd, który obsługiwać będzie komunikację bezpieczną. Przykładowo możemy zastosować jednostkę sterującą CU240E-2 PN-F.

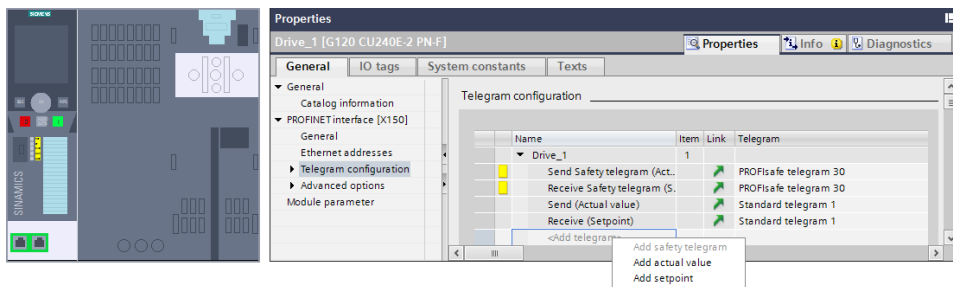


Po dodaniu przekształtnika z interfejsem PN, wykonujemy standardową konfigurację urządzenia klasy *PROFINET I/O device*, bo właśnie takim komponentem jest w strukturze projektu napęd. Nadajemy więc adres IP, nazwę urządzenia czy zakres przestrzeni adresowej.

Do konfiguracji napędu G120 należy również dodać w projekcie moduł mocy (PM).

Rozszerzenie telegramu komunikacyjnego

Interfejs wymiany danych pomiędzy napędem a sterownikiem nadrzędnym, definiowany jest poprzez standardowy telegram komunikacyjny. Wykonując konfigurację przez Startdrive system automatycznie przypisuje domyślny telegram (np. Standard telegram 1 dla napędów SINAMICS G PN lub DP). W celu zdefiniowania komunikacji *fail-safe* należy rozszerzyć ramkę komunikacyjną o telegram *PROFIsafe* 30 (lub jego wersję rozbudowaną – 900).



PROFIsafe telegram 30

Telegram 30 *fail-safe* składa się z 6 bajtów wejściowych oraz 6 bajtów wyjściowych. Poza systemowymi informacjami istotnymi z punktu widzenia testów bezpieczeństwa, w telegramie znajdziemy standardowo słowo statusowe:

SŁOWO STATUSOWE PROFIsafe (S_ZSW1 w telegramie 30)															
BYTE 0								BYTE 1							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
STO aktywne	SS1 aktywny	-	-	SLS aktywne	-	-	Zdarzenie wewnętrzne	-	SLS BIT 0	SLS BIT 1	-	SDI ujemny kierunek	SDI dodatni kierunek	-	SSM

oraz słowo sterujące:

SŁOWO STERUJĄCE PROFIsafe (S_STW1 w telegramie 30)															
BYTE 0								BYTE 1							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
STO	SS1	-	-	SLS	-	-	Potw. zd. wewn.	-	Akt. SLS BIT 0	Akt. SLS BIT 1	-	SDI pozytywny	SDI negatywny	-	-

Odwołując się w programie sterownika bezpośrednio do powyższych bitów możemy sprawdzić stan lub aktywować zintegrowane funkcje bezpieczeństwa (pod warunkiem, że wybrany napęd je obsługuje).

Wsparcie techniczne Siemens, jako alternatywę do takiego podejścia – przygotowało bibliotekę aplikacyjną (LDrvSafe), która ułatwia bardziej zaawansowane zarządzanie funkcjami bezpieczeństwa napędów SINAMICS. Bloki funkcyjne przewidziane są głównie dla sterowników serii S7-1200F/1500F, a ich zastosowanie wymaga środowiska inżynierskiego w wersji V14 lub nowszego. Dokumentacja oraz biblioteka dostępne są do pobrania w poniższej lokalizacji sieciowej:

<https://support.industry.siemens.com/cs/ww/en/view/109485794>

Parametry F napędu

Końcowym etapem konfiguracji ramki komunikacyjnej napędu jest ustawienie adresu *PROFIsafe*, wartości monitorowania czasu odpowiedzi oraz ewentualnie ręczne określenie numeru bloku F-DB przypisanego do napędu w projekcie sterownika.

Podobnie jak w projekcie sterownika – wszystkie parametry istotne z punktu widzenia funkcjonalności bezpieczeństwa oznaczone są żółtym znacznikiem.

Zwróćmy uwagę, iż analogicznie jak w przypadku modułów F-I/O, rozszerzając telegram komunikacyjny o dane *fail-safe* – system wygenerował w sterowniku blok danych F-DB, który zawiera informacje statusowe oraz zmienne aktywujące bądź potwierdzające pasywną ukladu spowodowaną błędami komunikacyjnymi.

Name	Comment
1	Input
2	PASS_ON 1=Enable passivation
3	ACK_NEG 1=Acknowledgment for reintegration required
4	ACK_REI 1=Acknowledgment for reintegration
5	IPAR_EN Tag for parameter reassignment of fail-safe DP standard slave
6	Output
7	PASS_OUT Passivation output
8	QBAD 1=Fail-safe values are output
9	ACK_REQ 1=Acknowledgment requirement for reintegration
10	IPAR_OK Tag for parameter reassignment of fail-safe DP standard slave
11	DIAG Non-fail-safe service information

Aktywacja zintegrowanych funkcji bezpieczeństwa napędu

Powyższe kroki stanowią podstawową konfigurację, jaką należy wykonać w projekcie TIA Portal, aby sterownik nadrzędny F-PLC znał strukturę interfejsu wymiany danych napędu, a także jej parametry. Poza tym należy wykonać standardową konfigurację przekształtnika częstotliwości.

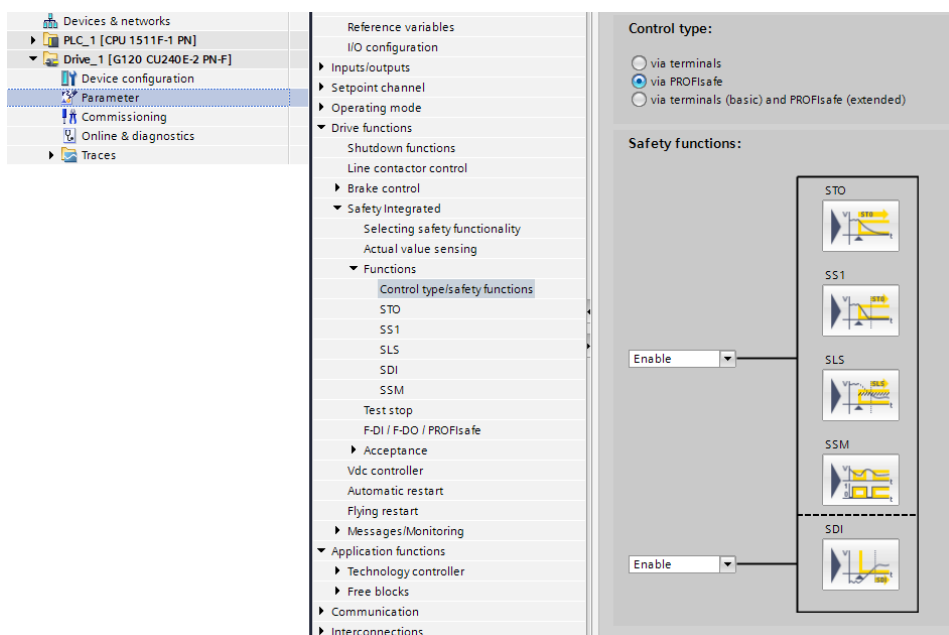
W przypadku napędów serii G jesteśmy w stanie wykonać ją całościowo przez Startdrive – instrukcja dla początkujących (w języku polskim) dostępna jest na stronie:

publikacje.siemens-info.com

lub w wersji oryginalnej poniżej:

https://w3.siemens.com/mcsm/mc-solutions/en/engineering-software/startdrive/Documents/Startdrive_GettingStarted_en-US.pdf

Od strony parametrów *fail-safe* istotne dla nas będzie przede wszystkim określenie metody wyzwolenia funkcji bezpieczeństwa oraz ich aktywacja w napędzie. Ustawienia te dostępne są w widoku parametrów napędu (*Parameter* → *Drive functions* → *Safety integrated* → *Functions*).



Zgodnie z powyższym zrzutem ekranu należy określić metodę komunikacji – w naszym przypadku *PROFIsafe*. Oraz aktywować dostępne funkcje bezpieczeństwa (*Enable*).

Warto również zajrzeć w zakładkę *F-DI/F-DO/PROFIsafe* aby upewnić się, że z konfiguracji telegramu (*F-address*) przeniesiony został poprawnie adres *PROFIsafe* konfigurowanego napędu.

Dokumentacja

Bardziej szczegółowa instrukcja konfiguracji komunikacji *PROFIsafe* pomiędzy sterownikiem F-PLC SIMATIC, a napędem SINAMICS G120 w środowisku Startdrive, dostępna jest w poniższej lokalizacji sieciowej:

<https://support.industry.siemens.com/cs/ww/en/view/109746271>

28.2.2 Konfiguracja przez Starter

W związku z faktem, że na dzień powstania niniejszej dokumentacji, nie wszystkie napędy SINAMICS obsługiwane są przez TIA Portal Startdrive – może zaistnieć konieczność zastosowania zewnętrznego narzędzia uruchomieniowego dedykowanego konfiguracji napędów – Starter. Narzędzie jest ogólnodostępne oraz bezpłatne, można pobrać je ze stron wsparcia technicznego Siemens:

<https://support.industry.siemens.com/cs/pl/en/view/26233208>

Dla przykładu rozważmy napęd serwo SINAMICS S120. Parametryzacja, uruchomienie oraz optymalizacja napędu poprzez narzędzie zewnętrzne jest procedurą standardową. W przypadku rozważanych konfiguracji, gdzie nadrzędnym urządzeniem w sieci jest sterownik F-PLC, wymagane będzie sprzężenie go z napędem w środowisku TIA Portal. W niniejszej dokumentacji nie będziemy zajmować się konfiguracją napędu od strony narzędzia Starter, jest to tematyka bardzo obszerna, która została opisana w wielu innych publikacjach. Pełna dokumentacja narzędzia dostępna jest w poniższej lokalizacji:

<https://support.industry.siemens.com/cs/pl/en/view/109740026>

Rozszerzenie telegramu komunikacyjnego – Starter

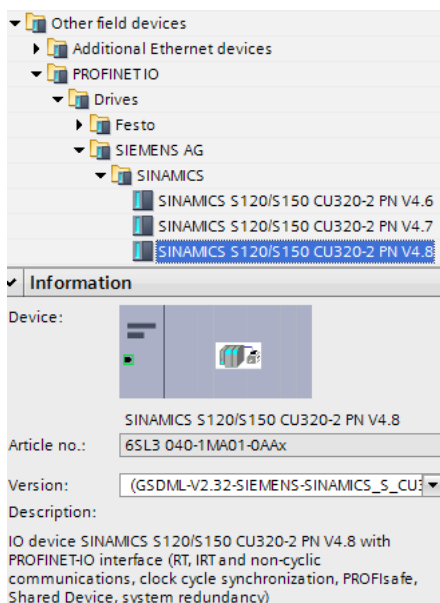
W naszej aplikacji interesować nas będzie kwestia aktywacji funkcji bezpieczeństwa napędu oraz dostosowanie interfejsu komunikacyjnego (telegramu) do wymiany danych w profilu *PROFIsafe*. Podobnie jak w przypadku napędu SINAMICS G120 – zaistnieje więc konieczność rozszerzenia telegramu o ramkę *fail-safe*. Procedura ta została opisana w poniższej dokumentacji – zarówno dla sieci PROFIBUS DP jak i PROFINET IO:

<https://support.industry.siemens.com/cs/pl/en/view/68624711>

Wstawienie napędu do projektu TIA Portal

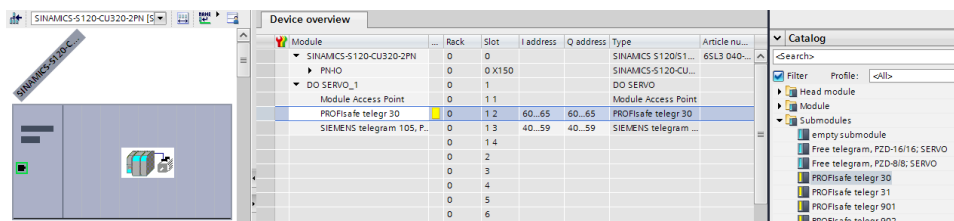
Zastosowanie napędów niezintegrowanych w narzędziu Startdrive nie stanowi problemu, wystarczy jedynie posiadać plik GSD (dla urządzeń PROFIBUS DP-slave) lub GSDML (dla PROFINET IO-device). Plik konfiguracyjny pobrać można ze stron wsparcia technicznego Siemens, dosyć rozbudowana baza danych znajduje się jednak już w standardowej instalacji TIA Portal.

Dodanie pliku odbywa się z poziomu widoku sieci (*Devices&networks*). Z katalogu urządzeń PROFINET IO możemy wybrać przykładowo napęd SINAMICS S120 z jednostką sterującą obsługującą komunikację w sieci PROFINET z uwzględnieniem profilu *PROFIsafe*.



Konfiguracja struktury napędu

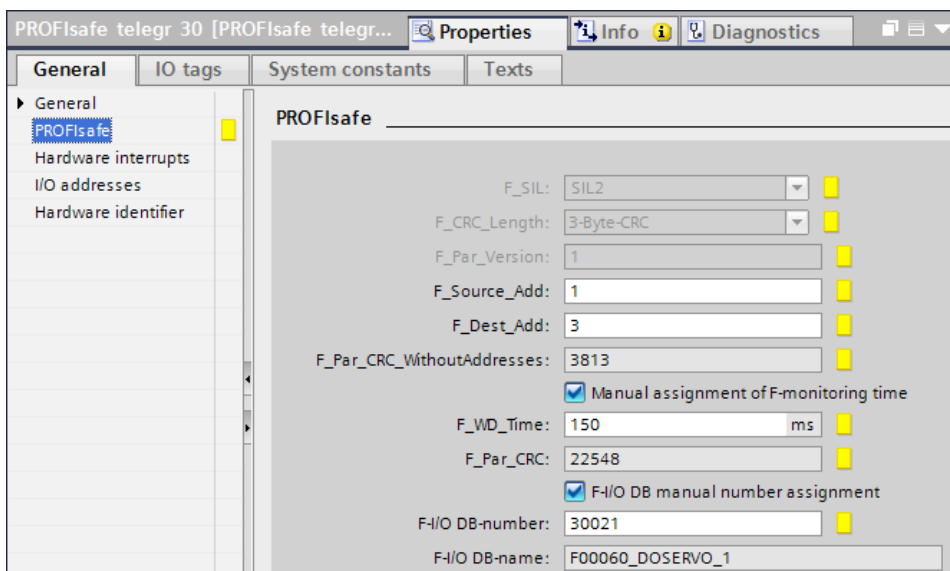
W kolejnym kroku, z dostępnych elementów konfiguracyjnych wybrać należy elementy składowe napędu – czyli np. jednostkę sterującą, obiekt serwo lub enkoder. Następnie określamy interfejs wymiany danych z systemem nadrzędnym, przez zdefiniowanie struktury telegramu komunikacyjnego. W rozważanym przypadku poza standardowym telegramem (np. SIEMENS telegram 105), który stosowaliśmy już w przypadku napędów serwo SINAMICS V90, musimy zastosować rozszerzenie w postaci ramki komunikacyjnej *PROFIsafe*. Analogicznie do konfiguracji napędu SINAMICS G120, rozbudowanie przestrzeni adresowej może odbywać się przez telegram 30.



Parametry F napędu

Znamy już doskonale parametry jakie należy wziąć pod uwagę podczas konfiguracji serwonapędu w sieci PROFINET IO. Poznane w poprzednich rozdziałach zasady znajdują zastosowanie również i w konfiguracji przez pliki GSD/GSDML. W związku z powyższym – podstawowe kroki konfiguracyjne, takie jak stworzenie synchronicznego bloku OB w sterowniku nadrzędnym, parametryzacja urządzenia w sieci PROFINET IO (adres, nazwa), określenie topologii sieci czy aktywacja trybu izochronicznego – wykonujemy zgodnie z poznaną procedurą.

Uzupełnieniem parametryzacji standardowej będzie sprawdzenie parametrów napędu pod kątem funkcji bezpieczeństwa – co wykonamy analogicznie do poprzedniego przykładu (uwzględniającego napęd z rodziny SINAMICS G).

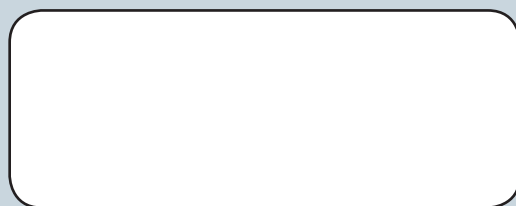


Istotne będzie sprawdzenie adresacji *PROFIsafe*, ewentualnie określenie wartości monitorowania czasu odpowiedzi lub ręczne przypisanie numeru bloku F-DB. Zwróćmy uwagę, iż dla urządzenia opisanego plikiem konfiguracyjnym – w projekcie sterownika również powinien zostać wygenerowany systemowo blok danych F o standardowej strukturze.

Aktywacja zintegrowanych w napędzie funkcji bezpieczeństwa odbywa się przez narzędzie Starter – opis we wskazanej powyżej dokumentacji.

Biura sprzedaży:

Siemens Sp. z o.o.
Digital Factory
Factory Automation
03-821 Warszawa
ul. Żupnicza 11
tel.: 22 870 8200
fax: 22 870 9868



Regionalne biura sprzedaży:

80-387 Gdańsk
ul. Arkonska 6
tel.: 58 764 6092
fax: 58 764 6099

40-527 Katowice
ul. Gawronów 22
tel.: 32 208 4134
fax: 32 208 4139

30-443 Kraków
ul. Józefa Marcika 14B
tel.: 12 363 8220
fax: 12 363 8229

92-333 Łódź
ul. Wydawnicza 1/3
tel.: 42 677 1791
fax: 42 677 1799

60-164 Poznań
ul. Ziębicka 35
tel.: 61 664 9843
fax: 61 664 9864

87-100 Toruń
ul. Gdańska 4A
tel.: 56 656 4216
fax: 56 656 4250

53-611 Wrocław
ul. Strzegomska 52
tel.: 71 777 5070
fax: 71 777 5050