

STARTER - sinamics_s120 - [Drive_unit_1 - Telegram configuration]

Project Edit Target system View Options Window Help

sinamics_s120

- Insert single drive unit
- Drive_unit_1
 - Automatic Configuration
 - Overview
 - Communication
 - Commiss. interface
 - Telegram configuration**
 - Topology
 - CU_S
 - Infeeds
 - Input/output components
 - Encoder
 - Drives
 - SERVO_02
 - SERVO_03
 - SERVO_04
 - Documentation
 - SINAMICS LIBRARIES
 - MONITOR

IF1: PROFIdrive PZD telegrams | IF2: PZD telegrams

Communication interface: PROFINET - Control Unit onboard (isochronous)
The PROFIsafe communication is performed via this interface

The PROFIdrive telegrams of the drive objects are transferred in the following order:
The input data corresponds to the send and the output data of the receive direction of the drive object.

Master view:

Object	Drive object	-No.	Telegram type	Input data Length	Output data Length
1	SERVO_02	2	Free telegram configuration with BICO	0	0
2	SERVO_03	3	Free telegram configuration with BICO	0	0
3	SERVO_04	4	Free telegram configuration with BICO	0	0
4	TM41_05	5	Free telegram configuration with BICO	0	0
5	CU_S	1	Free telegram configuration with BICO	0	0

DOs that are not assigned to a slot. (No cyclic data exchange)

IF1: PROFIdrive PZD telegrams | IF2: PZD telegrams

Communication interface: PROFINET - Control Unit onboard (isochronous)
The PROFIsafe communication is performed via this interface

The PROFIdrive telegrams of the drive objects are transferred in the following order:
The input data corresponds to the send and the output data of the receive direction of the drive object.

Master view:

Object	Drive object	-No.	Telegram type	Input data Length	Output data Length
1	SERVO_02	2	Standard telegram 1, PZD-2/2	2	2
2	SERVO_03	3	Standard telegram 1, PZD-2/2	2	2
3	SERVO_04	4	Standard telegram 1, PZD-2/2	2	2
4	TM41_05	5	Standard telegram 3, PZD-5/9	9	5
5	CU_S	1	SIEMENS telegram 390, PZD-2/2	2	2

DOs that are not assigned to a slot. (No cyclic data exchange)

Parametryzacja układu napędowego SINAMICS S120 jest spójna pomiędzy wersjami zapisanymi w trybie online oraz offline. W przypadku rozbieżności należy przenieść zawartość pamięci ulotnej do trwałej – „Copy RAM to ROM”, następnie pobrać bieżącą parametryzację przekształtnika zapisaną w jego pamięci wewnętrznej na dysk twardy stacji roboczej.

Konfigurację ramki komunikacyjnej możemy przeprowadzić poprzez dwukrotne kliknięcie w drzewie projektowym w „Telegram configuration” (1).

W oknie głównym projektu wyświetlona zostanie bieżąca konfiguracja telegramu komunikacyjnego (2), której modyfikację możemy przeprowadzić w trybie offline (3).

Omawiana konfiguracja hardwarowa układu napędowego zbudowana jest z trzech modułów silnikowych **SERVO_02**, **SERVO_03** oraz **SERVO_04**, modułu dodatkowych wejść/wyjść **TM41_05** oraz jednostki sterującej **CU_S**. Każdy z wymienionych komponentów może zostać uwzględniony w strukturze ramki wymiany danych z systemem nadrzędnym.

Wstępnie dla każdego z komponentów składowych systemu napędowego **SINAMICS S120** definiujemy telegramy standardowe, których typy zależne są od rodzaju urządzenia (4).

STARTER - sinamics_s120 - [Drive_unit_1 - Telegram configuration]

Project Edit Target system View Options Window Help

sinamics_s120

- Insert single drive unit
- Drive_unit_1
 - Automatic Configuration
 - Overview
 - Communication
 - Commiss. interface
 - Telegram configuration
 - Topology
 - CU_S
 - Infeeds
 - Input/output components
 - Encoder
 - Drives
 - SERVO_02
 - SERVO_03
 - SERVO_04
 - Documentation
 - SINAMICS LIBRARIES
 - MONITOR

IF1: PROFIdrive PZD telegrams | IF2: PZD telegrams

Communication interface: PROFINET - Control Unit onboard (isochronous)
The PROFIsafe communication is performed via this interface

The PROFIdrive telegrams of the drive objects are transferred in the following order:
The input data corresponds to the send and the output data of the receive direction of the drive obj

Master view:

Object	Drive object	-No.	Telegram type	Input data Length	Output data Length
1	SERVO_02	2	Free telegram configuration with BICO	0	0
2	SERVO_03	3	Free telegram configuration with BICO	0	0
3	SERVO_04	4	Free telegram configuration with BICO	0	0
4	TM41_05	5	Free telegram configuration with BICO	0	0
5	CU_S	1	Free telegram configuration with BICO	0	0

DOs that are not assigned to a slot. (No cyclic data exchange)

IF1: PROFIdrive PZD telegrams | IF2: PZD telegrams

Communication interface: PROFINET - Control Unit onboard (isochronous)
The PROFIsafe communication is performed via this interface

The PROFIdrive telegrams of the drive objects are transferred in the following order:
The input data corresponds to the send and the output data of the receive direction of the drive

Master view:

Object	Drive object	-No.	Telegram type	Input data Length	Output data Length
1	SERVO_02	2	Standard telegram 1, PZD-2/2	2	2
2	SERVO_03	3	Standard telegram 1, PZD-2/2	2	2
3	SERVO_04	4	Standard telegram 1, PZD-2/2	2	2
4	TM41_05	5	Standard telegram 3, PZD-5/9	9	5
5	CU_S	1	SIEMENS telegram 390, PZD-2/2	2	2

DOs that are not assigned to a slot. (No cyclic data exchange)

Download to target system (WWBS824)

The data will be downloaded to the target system. This can take several minutes!

☐ Store additional data on the target device
☐ Including DCC chart data
☒ After loading, copy RAM to ROM

Start download?

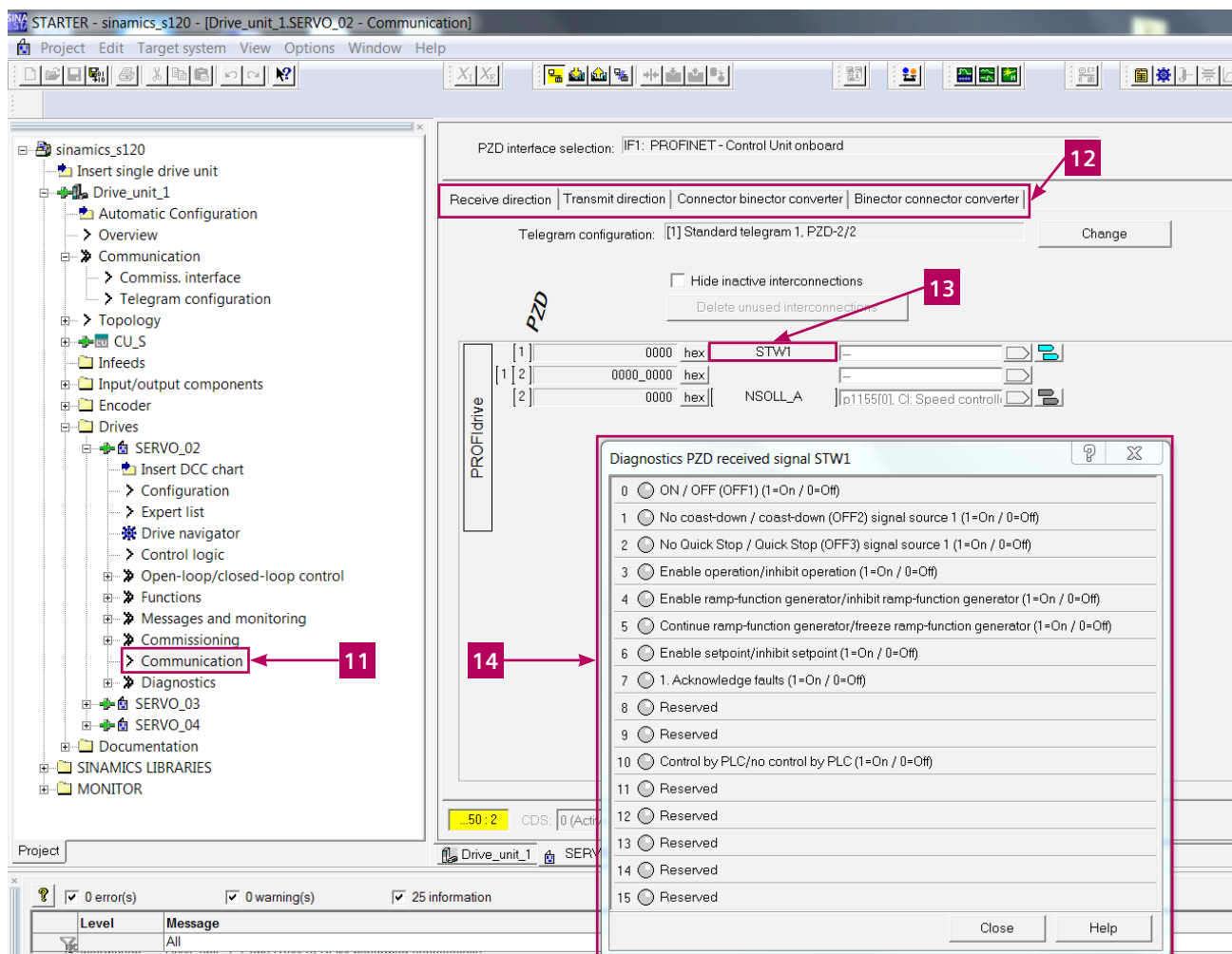
Yes No

W kolejnym kroku zapisujemy projekt i przechodzimy do trybu **online** (5). Program **STARTER** zgłasza rozbieżność parametryzacji pomiędzy wersjami projektu zapisanymi w przeksztaltniku częstotliwości a trybem **offline** (projektem zapisanym na dysku twardym PC/PG) (6).

Dokonana modyfikacja budowy ramki telegramu komunikacyjnego w trybie **offline** (7) nie jest widoczna po przejściu w tryb **online**.

Wprowadzone modyfikacje w wersji **offline** wgrujemy do przeksztaltnika częstotliwości wybierając polecenie „Download project to target system” (8). Proces ładowania nastaw projektu z trybu **offline** akceptujemy w wyświetlonym oknie kontekstowym przyciskiem **OK** (9).

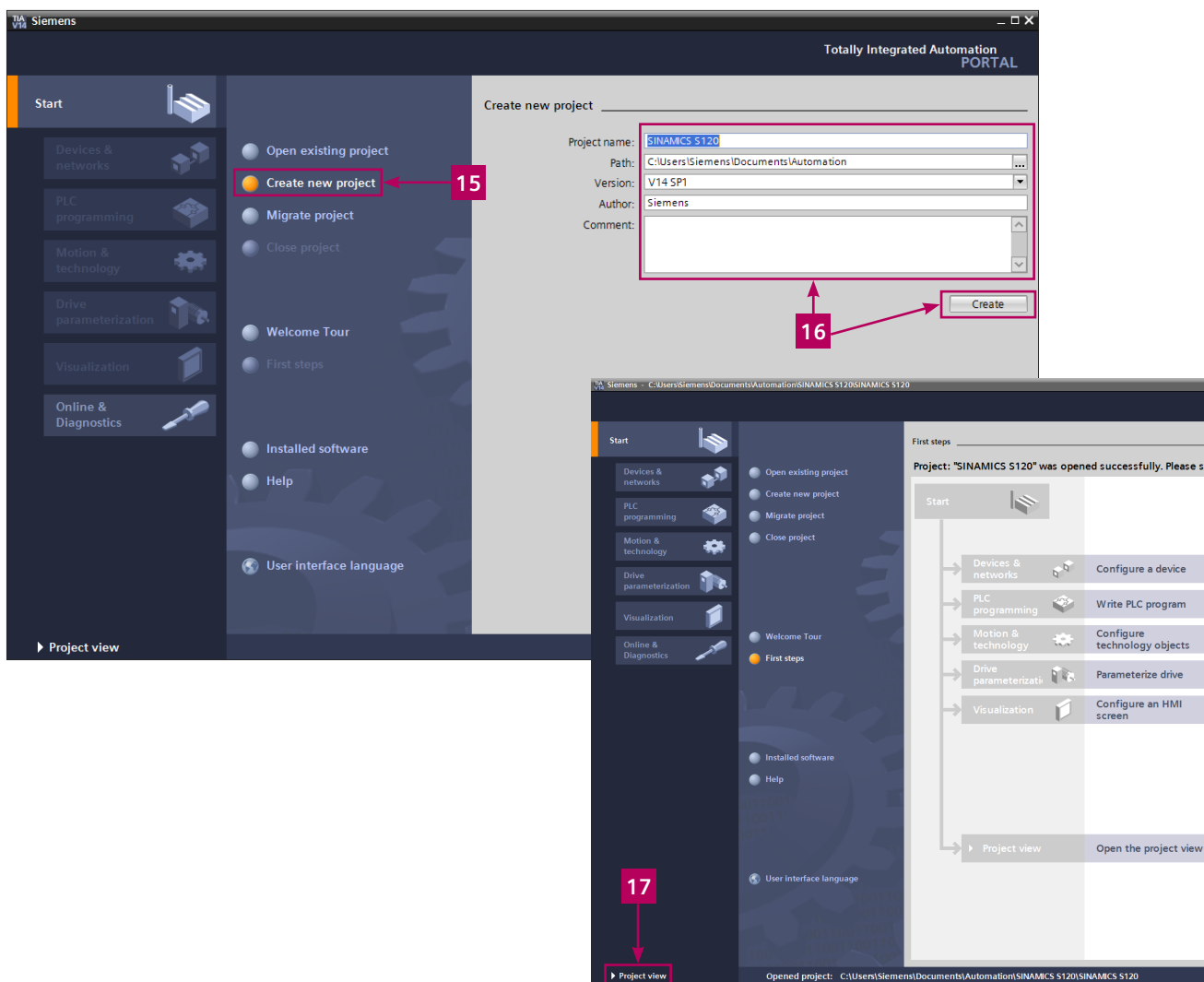
Po zakończeniu ładowania projektu widok budowy telegramu komunikacyjnego zostanie odświeżony zgodnie z wgraną do przeksztaltnika konfiguracją (10).



Szczegółowy widok budowy telegramu komunikacyjnego dla poszczególnych obiektów napędowych dostępny jest poprzez wybranie właściwego obiektu w drzewie projektowym, następnie dwukrotnie klikamy w menu „Communication” (11).

W oknie głównym możemy wyświetlić strukturę telegramu w obszarze danych otrzymywanych z PLC „Receive direction” oraz informacje wysyłane z przekształtnika częstotliwości do systemu sterowania PLC „Transmit direction” (12).

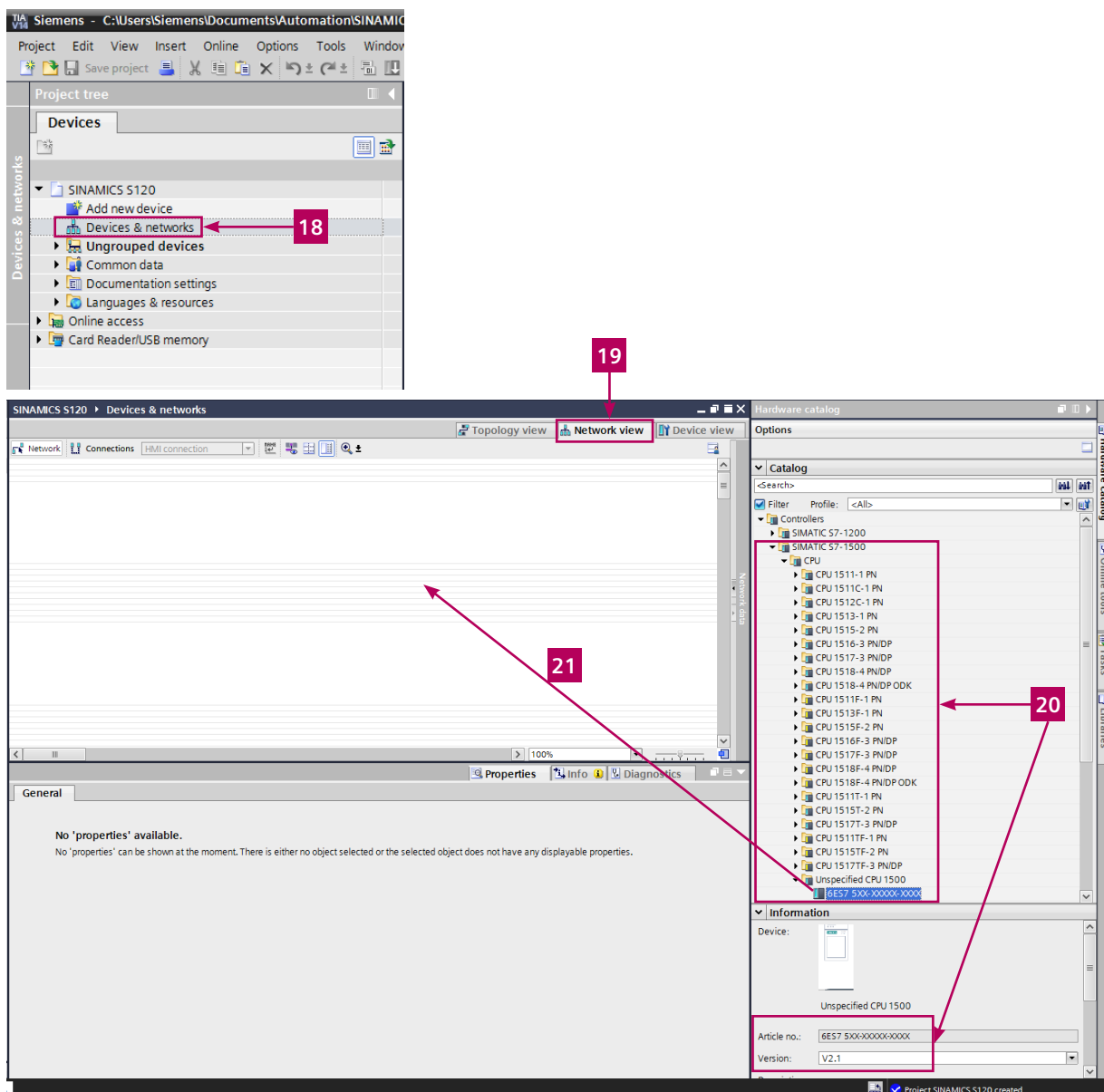
W przypadku wybrania telegramu predefiniowanego (Standardowego), przy każdym słowie sterującym/statusowym dostępny jest przycisk **STW/ZSW** (13) za pomocą którego możemy wyświetlić status poszczególnych bitów interesującego nas słowa (14).



W kolejnych krokach przystępujemy do konfiguracji projektu PLC, w tym celu uruchamiamy TIA Portal V14.

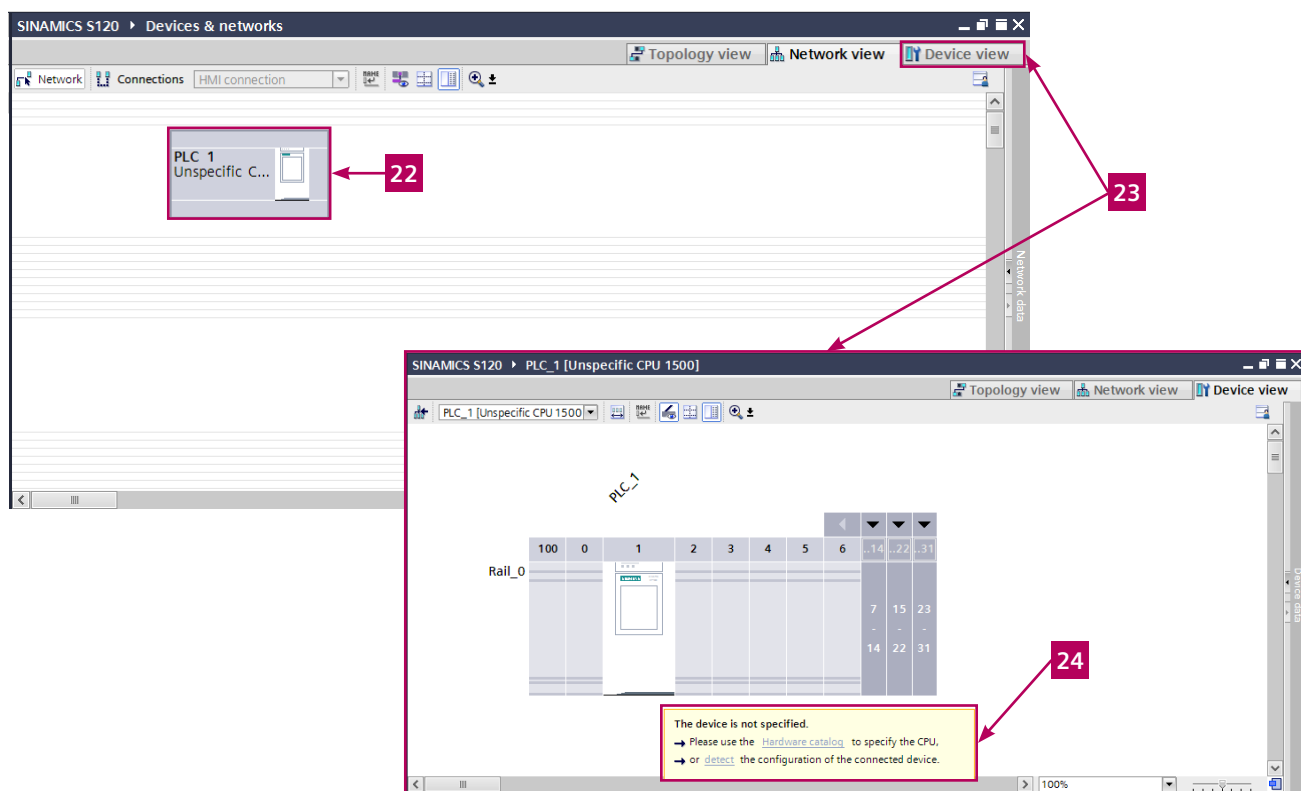
Tworzymy nowy projekt (15) wprowadzając jego nazwę, a następnie klikamy w przycisk „Create” (16).

Przechodzimy do widoku projektu „Project view” (17).

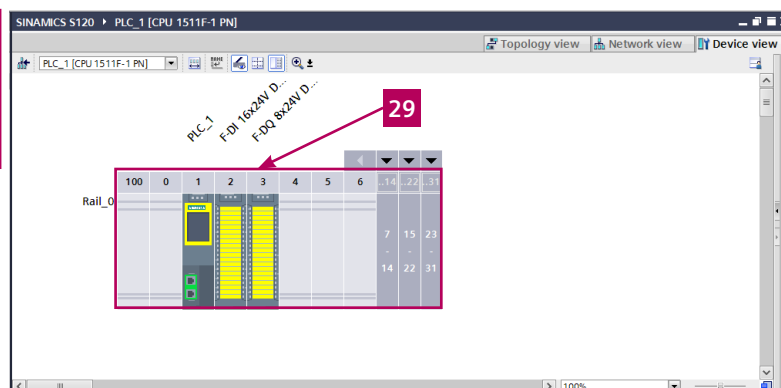
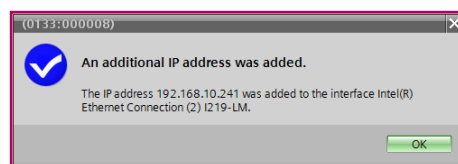
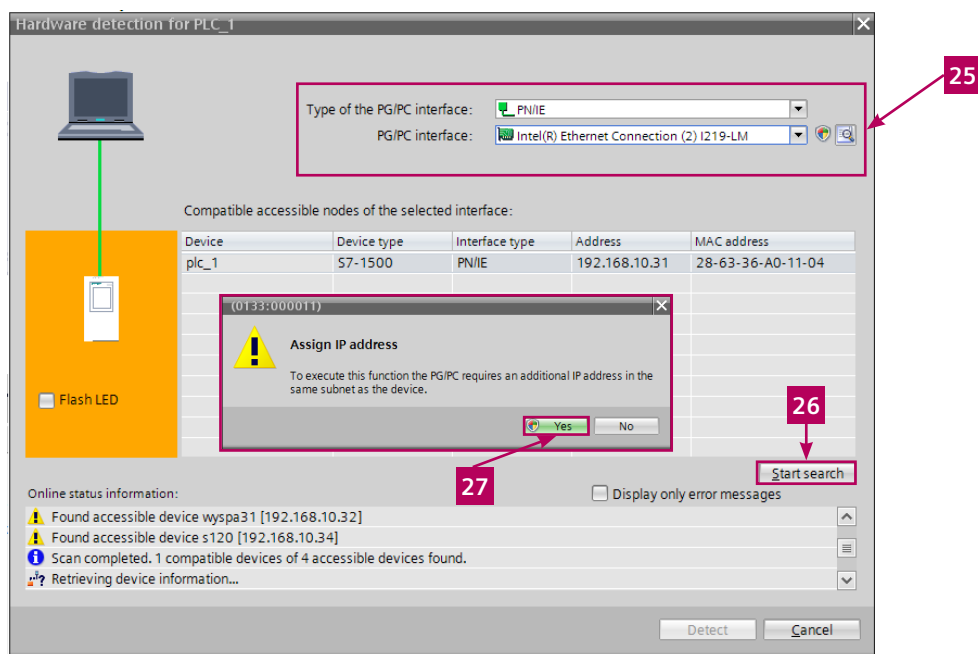


W drzewie projektowym dwukrotnie klikamy w polecenie „**Devices and networks**” (18) – przystępujemy do konfiguracji hardwarowej aplikacji. W widoku „**Network view**” (19) przechodzimy do katalogu sprzętowego z którego wybieramy odpowiedni sterownik PLC. W naszym przykładzie posługujemy się wersją nieokreśloną sterownika, który zostanie rozpoznany w trybie online (20).

Pamiętajmy aby zwrócić uwagę na wersję programową sterownika, wybieramy właściwą a następnie przeciągamy obiekt do widoku sieciowego (21).



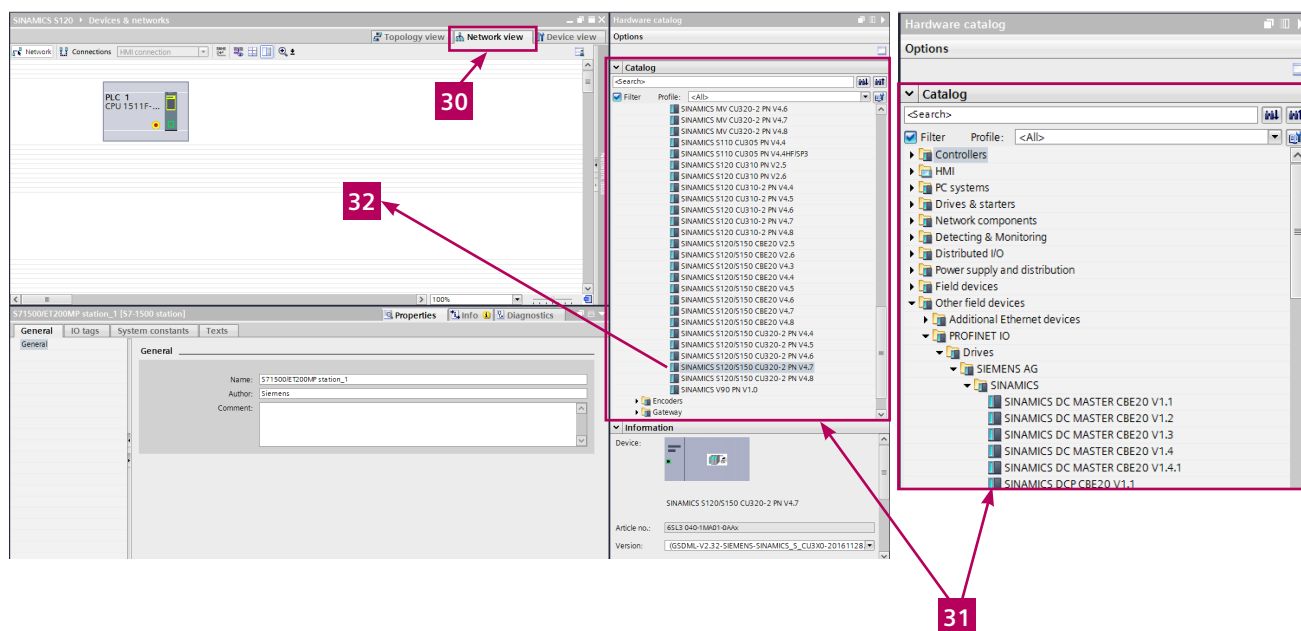
Dwukrotnie klikamy w stworzony w ten sposób obiekt (22) (lub zaznaczamy i przechodzimy do widoku urządzenia – 23). Następnie wybieramy polecenie „detect” (24).



Określamy interfejs komunikacyjny i rozpoczynamy przeszukiwanie magistrali w celu wyświetlenia kompatybilnych urządzeń (25). Jeżeli adres IP stacji roboczej jest inny niż przypisany do PLC, TIA zaproponuje przyjęcie tymczasowej adresacji z puli adresów właściwych dla PLC (27). Zmiana adresu IP na adres tymczasowy jest wyświetlona w oknie kontekstowym (28), okno zamykamy przyciskiem OK.

Następnie dokonujemy rozpoznania HW plc poprzez kliknięcie w przycisk „Detect”.

Rozpoznany sterownik PLC widoczny jest w racku (29), poza jednostką sterującą PLC zostaje również pobrana struktura podłączonych dodatkowych modułów IO. W naszym przykładzie posługujemy się sterownikiem PLC z rodziny S7-1500 w wersji F.

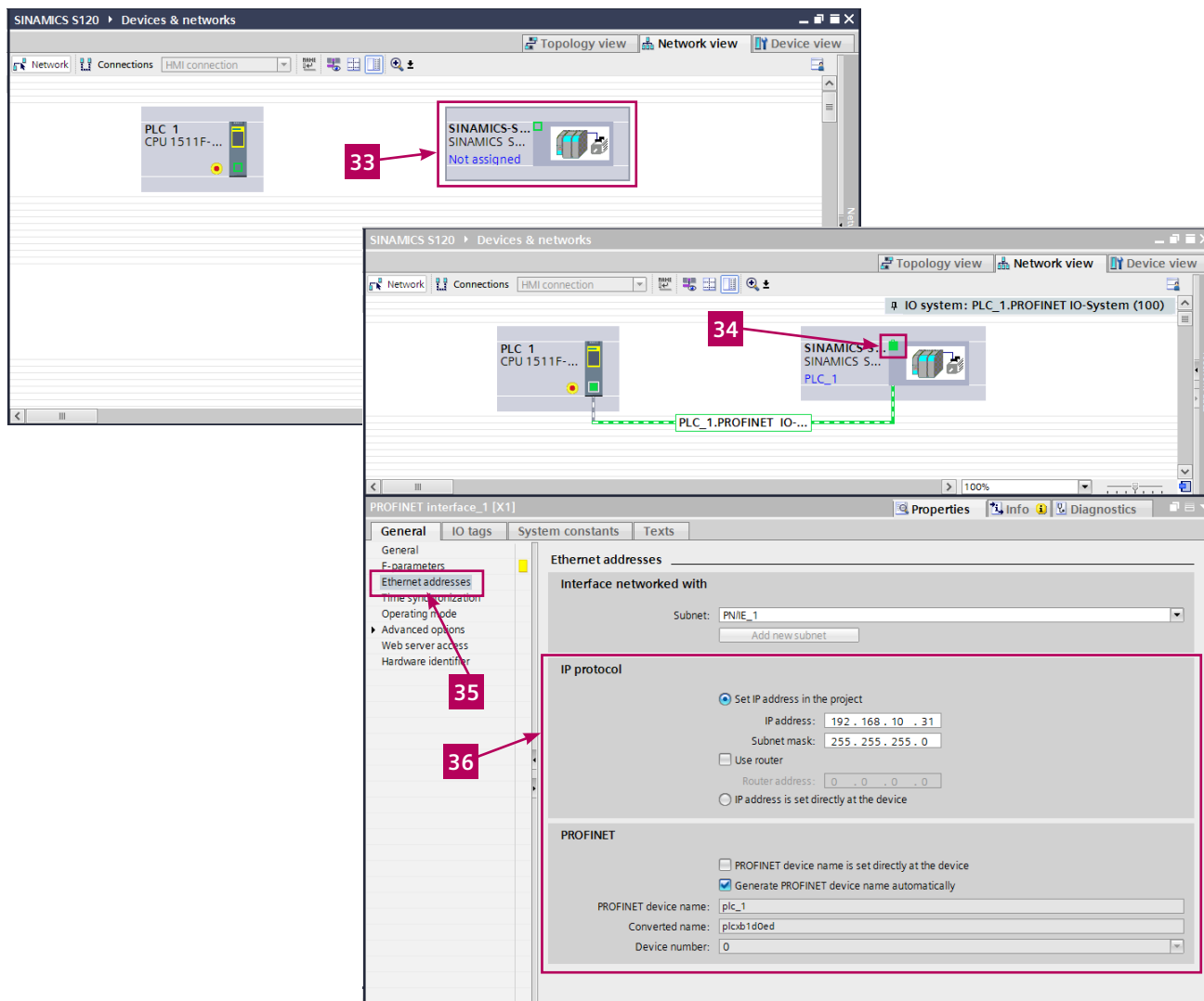


Kolejnym krokiem konfiguracji jest wstawienie w widok sieciowy (30) właściwego układu napędowego.

W wersji TIA V14 SINAMICS S120 może zostać dodany do projektu poprzez plik **GSD/GSDML** który znajdziemy w katalogu HW (31) w lokalizacji:

Other field devices → PROFINET IO → Drives → SIEMENS AG → SINAMICS.

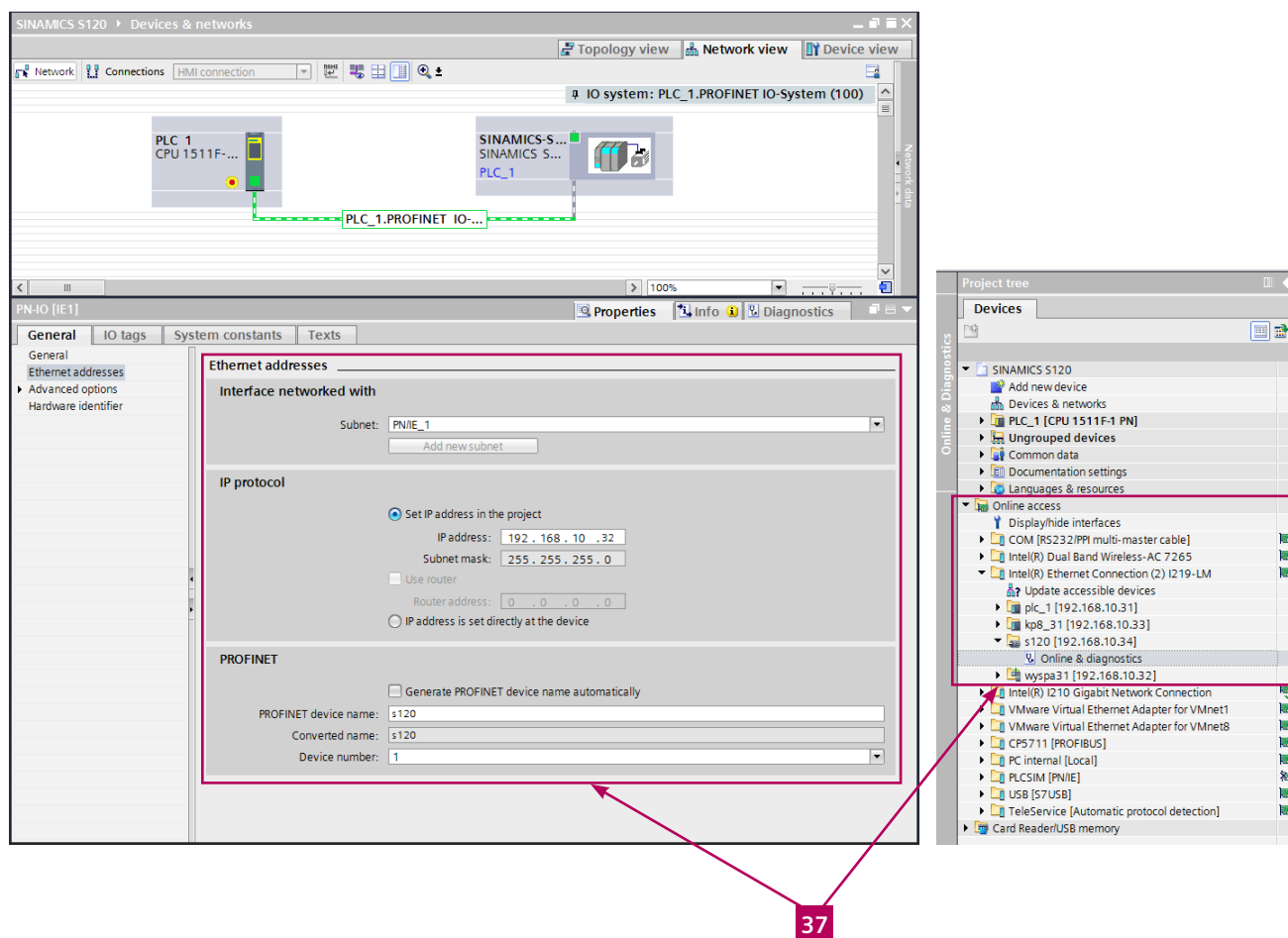
Odszukujemy właściwą jednostkę sterującą „**CU320-2 PN**” z odpowiednią wersją firmware – w naszym przykładzie FW 4.7, którą następnie przeciągamy do okna głównego TIA Portal (widok sieciowy – 32).



W kolejnym kroku łączymy port komunikacyjny przekształtnika częstotliwości (33) z portem właściwego sterownika PLC. Możemy również kliknąć w napis „**Not assigned**” i z listy dostępnych PLC wskazać właściwego mastera.

Następnie prawym przyciskiem myszy klikamy w port komunikacyjny przekształtnika częstotliwości, z wyświetlonego podmenu wybieramy opcję „**Properties**” (34).

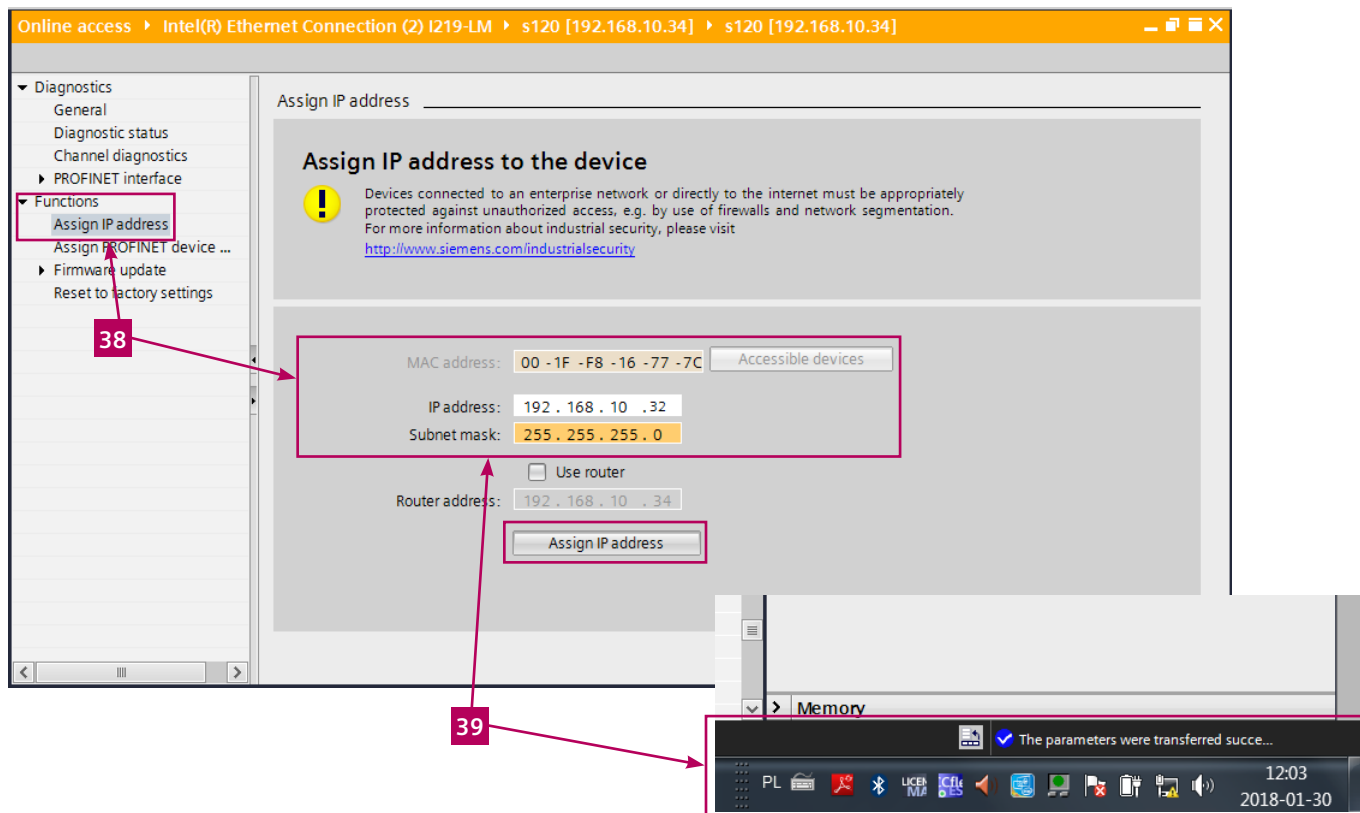
W oknie właściwości przekształtnika przechodzimy do zakładki „**Ethernet addresses**” (35) w celu ustalenia właściwej adresacji IP oraz **nazwy profinetowej** urządzenia (36).



Zarówno adres **IP** jak i **nazwa profinetowa** przyjęta w projekcie TIA musi być zgodna z konfiguracją urządzenia **online**.

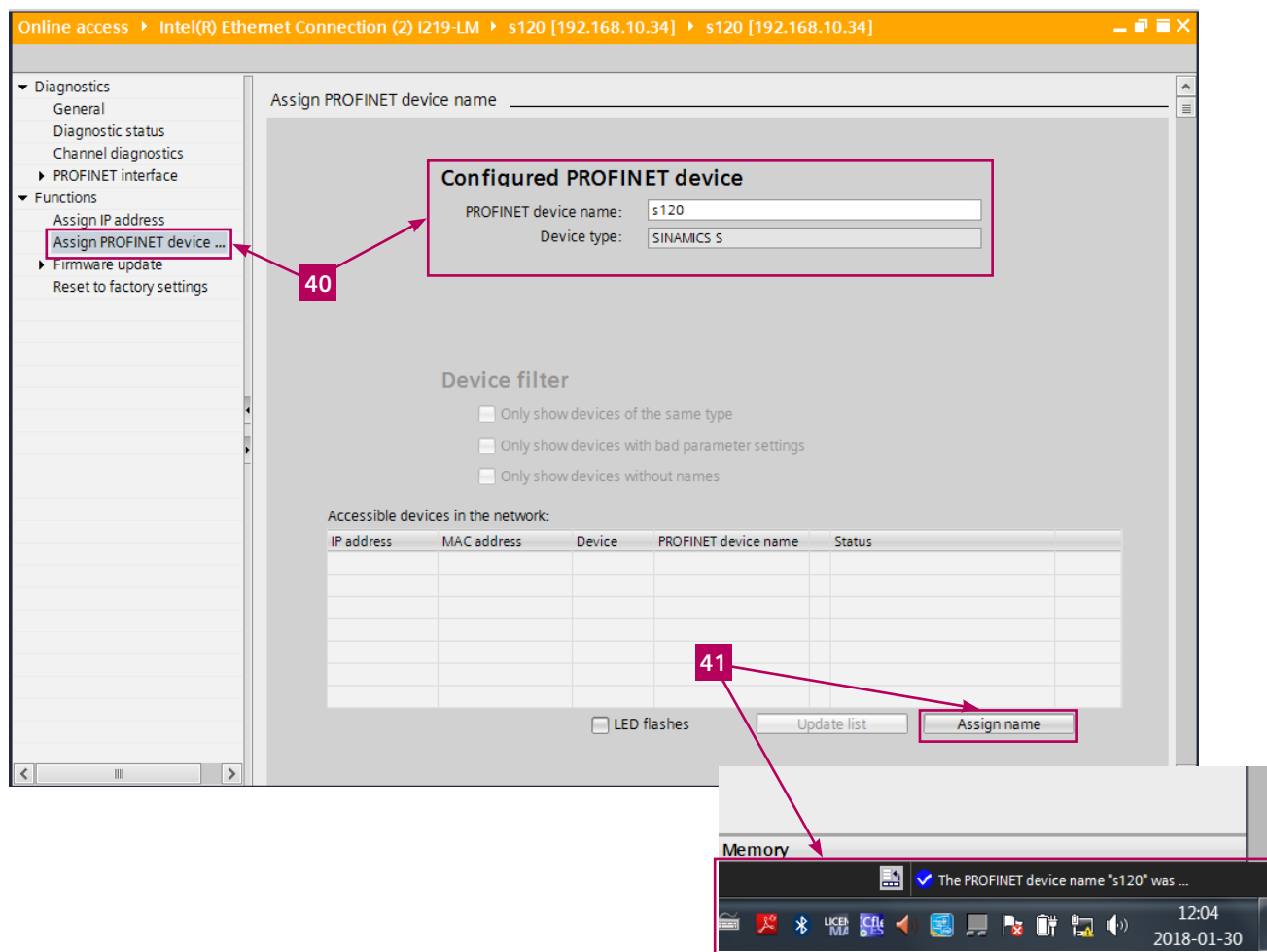
W celu sprawdzenia danych zapisanych w urządzeniu sieciowym przechodzimy do widoku odpowiedniej karty sieciowej, pod którą znajdziemy wymagane informacje dostępnych urządzeń (37).

Jeżeli chcemy dokonać modyfikacji adresu IP lub **nazwy profinet** urządzenia należy kliknąć polecenie „**Online and diagnostics**” w widoku dostępnym pod kartą sieciową.



Modyfikacja **adresu IP** dostępna jest poprzez menu „**Functions**” następnie „**Assign IP address**” (38).

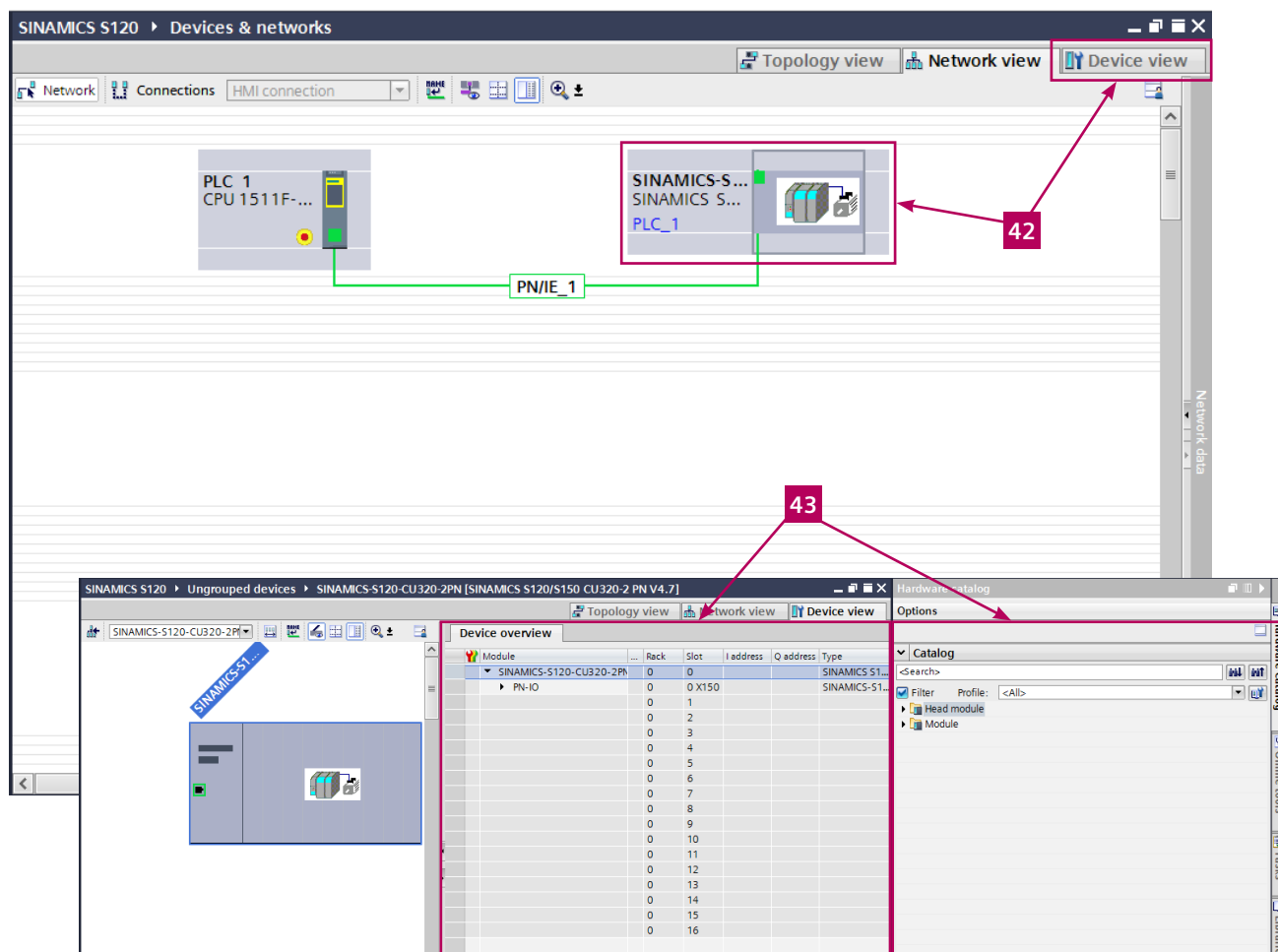
Status operacji przypisania/zmiany adresu IP urządzenia sieciowego widoczny będzie w dolnym prawym rogu TIA portal (39).



Podobnie postępujemy w celu modyfikacji nazwy profinet urządzenia:

menu „**Functions**” następnie „**Assign Profinet device...**”, wprowadzamy nazwę sieciową (40).

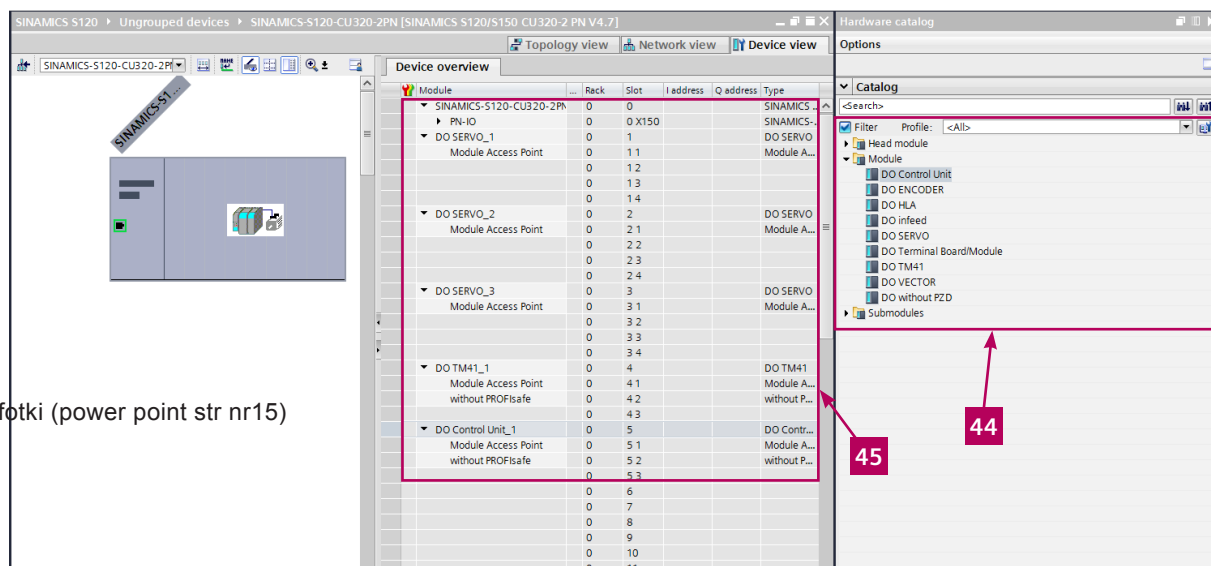
Status przypisania/zmiany nazwy urządzenia sieciowego widoczny będzie w dolnym prawym rogu TIA portal (41).



W kolejnym kroku przystępujemy do definiowania struktury telegramu komunikacyjnego przekształtnika częstotliwości. W tym celu zaznaczamy przekształtnik częstotliwości a następnie przechodzimy do widoku urządzenia (42).

W oknie **Device overview** musimy odwzorować strukturę telegramu komunikacyjnego zdefiniowaną po stronie konfiguracji przekształtnika częstotliwości. Poszczególne elementy składowe znajdziemy w katalogu hardwarem TIA (43).

brak fotki (power point str nr15)



Dla przypomnienia – telegram komunikacyjny określony podczas parametryzacji przekształtnik częstotliwości zbudowany jest z trzech obiektów **SERVO** – obsługujących **telegramy standardowe nr: 1**, modułu dodatkowych wejść/wyjść **TM41** obsługującego telegram **standardowy nr: 3** oraz jednostki sterującej **CU320-2** obsługującej **telegram standardowy nr: 390 (46)**.

Kolejność poszczególnych obiektów oraz definicja telegramów komunikacyjnych pomiędzy parametryzacją w programie **STARTER** a budową ramki komunikacyjnej w **TIA Portal** muszą być **identyczne**.

Z katalogu HW (44) wybieramy poszczególne **obiekty DO**: **DO Servo**, **DO TM41** oraz **DO Control Unit** (45).

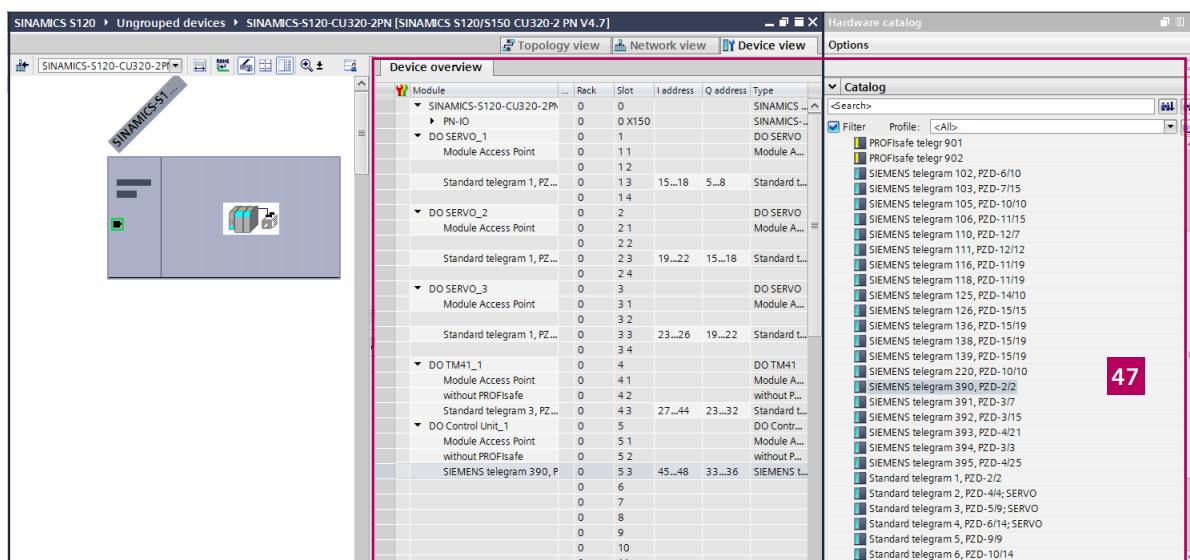
TIA Portal

DO SERVO_1
DO SERVO_2
DO SERVO_3
DO TM41_1
DO Control Unit_1

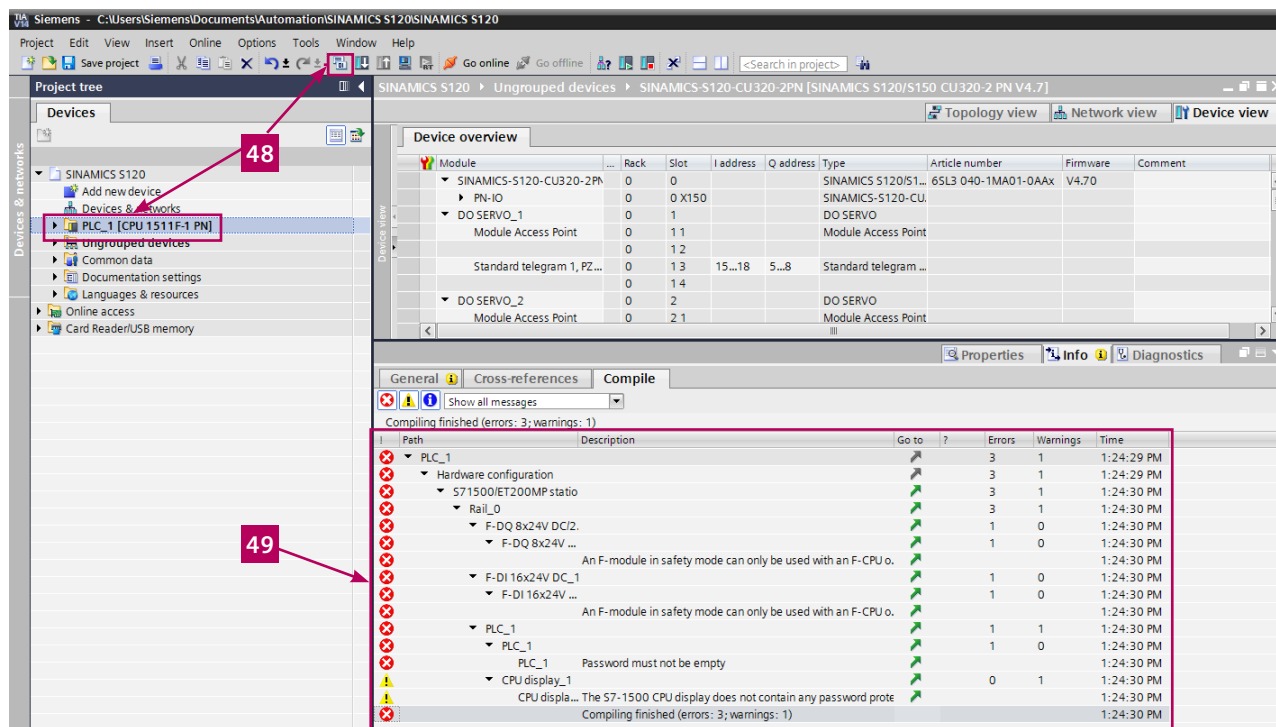


STARTER

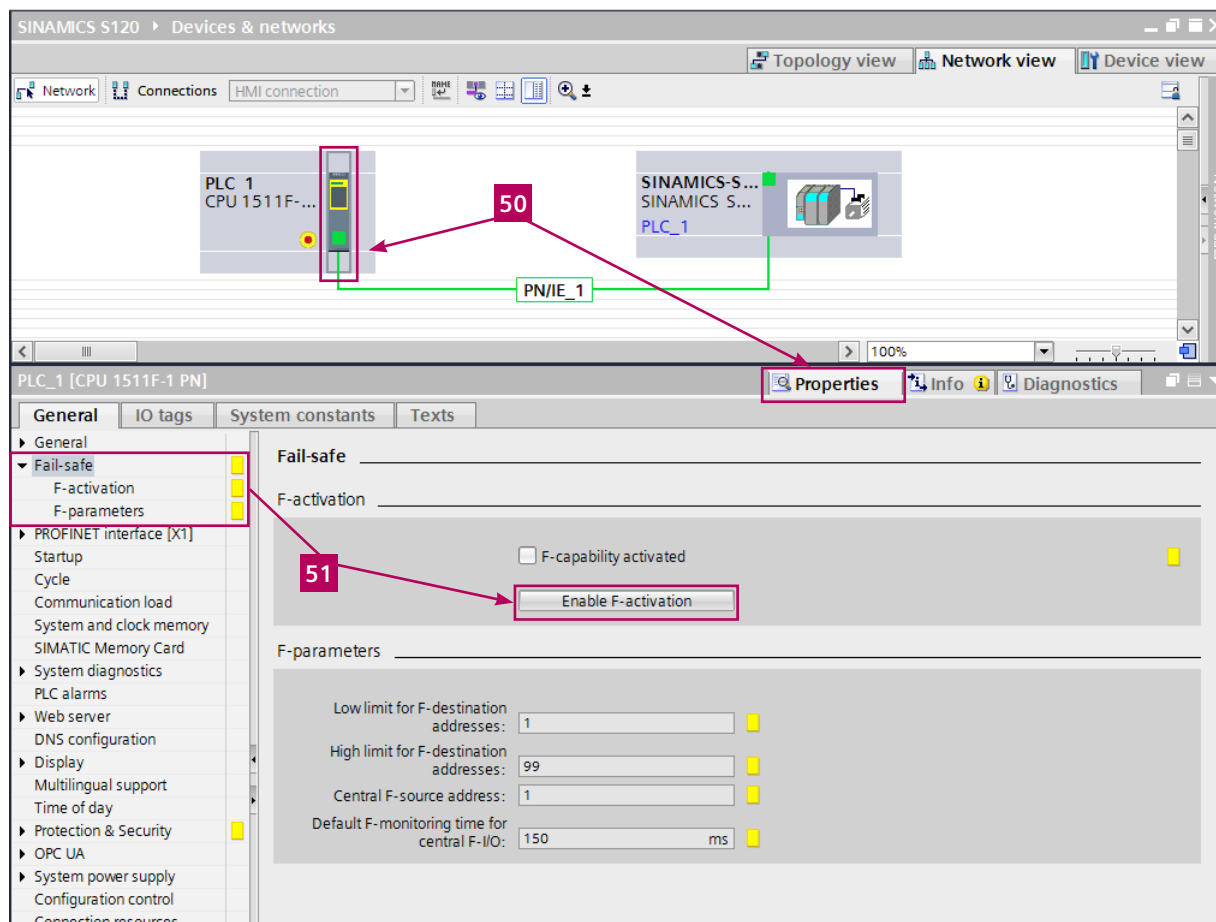
SERVO_02
SERVO_03
SERVO_04
TM41_04
CU_S



Następnie dla wprowadzonych obiektów definiujemy właściwe telegramy komunikacyjne – które wybieramy z katalogu HW, folder „Submodules” (47).



W związku z tym iż sterownik PLC na bazie którego budowana jest aplikacja jest sterownikiem bezpiecznym, kompilacja może zakończyć się wygenerowaniem błędów. Podczas identyfikacji PLC rozpoznane zostały również moduły bezpieczne – ich stosowanie możliwe jest wyłącznie po załączeniu funkcji **SF w PLC**.



W widoku sieci i urządzeń zaznaczamy **PLC**, następnie w zakładce właściwości (50), przechodzimy do menu **Fail-safe** w którym aktywujemy funkcjonalność związaną z bezpieczeństwem maszynowym (51).

Access level

Select the access level for the PLC.

Update password encryption

Access level	HMI	Read	Write	Fail-safe	Access per...
☐ Full access incl. fail-safe (no protection)	✓	✓	✓	✓	Access per...
☒ Full access (no protection)	✓	✓	✓	✓	Access per...
☐ Read access	✓	✓	✓	✓	Access per...
☐ HMI access	✓	✓	✓	✓	Access per...
☐ No access (complete protection)					Access per...

Full access (no protection):
TIA Portal users will have access to standard functions.
HMI applications can access all functions (fail-safe and standard).

Mandatory password:
For additional access to the fail-safe functions, TIA Portal users need to enter the "full access incl. fail-safe" password.

Project tree:

- PLC_1 [CPU 1511F-1 PN]
- PROFINET interface_1
- F-DI 16x24V DC_1
- F-DQ 8x24V DC/2A PPM_1

Device overview:

Module	Rack	Slot	I address	Q address	Type	Article no.	Firmware	Comment
PLC_1	0	1			CPU 1511F-1 PN	6ES7 511-1FK01-0AB0	V2.1	
PROFINET interface_1	0	1X1			PROFINET interface	6ES7 526-1BH00-0AB0	V1.0	
F-DI 16x24V DC_1	0	2	0...8	0...4	F-DI 16x24V DC	6ES7 526-1BH00-0AB0	V1.0	
F-DQ 8x24V DC/2A PPM_1	0	3	9...14	9...14	F-DQ 8x24V DC/2A	6ES7 526-2BF00-0AB0	V1.0	

Compile:

Protect the coded blocks F0B_RTG1 (0B123)

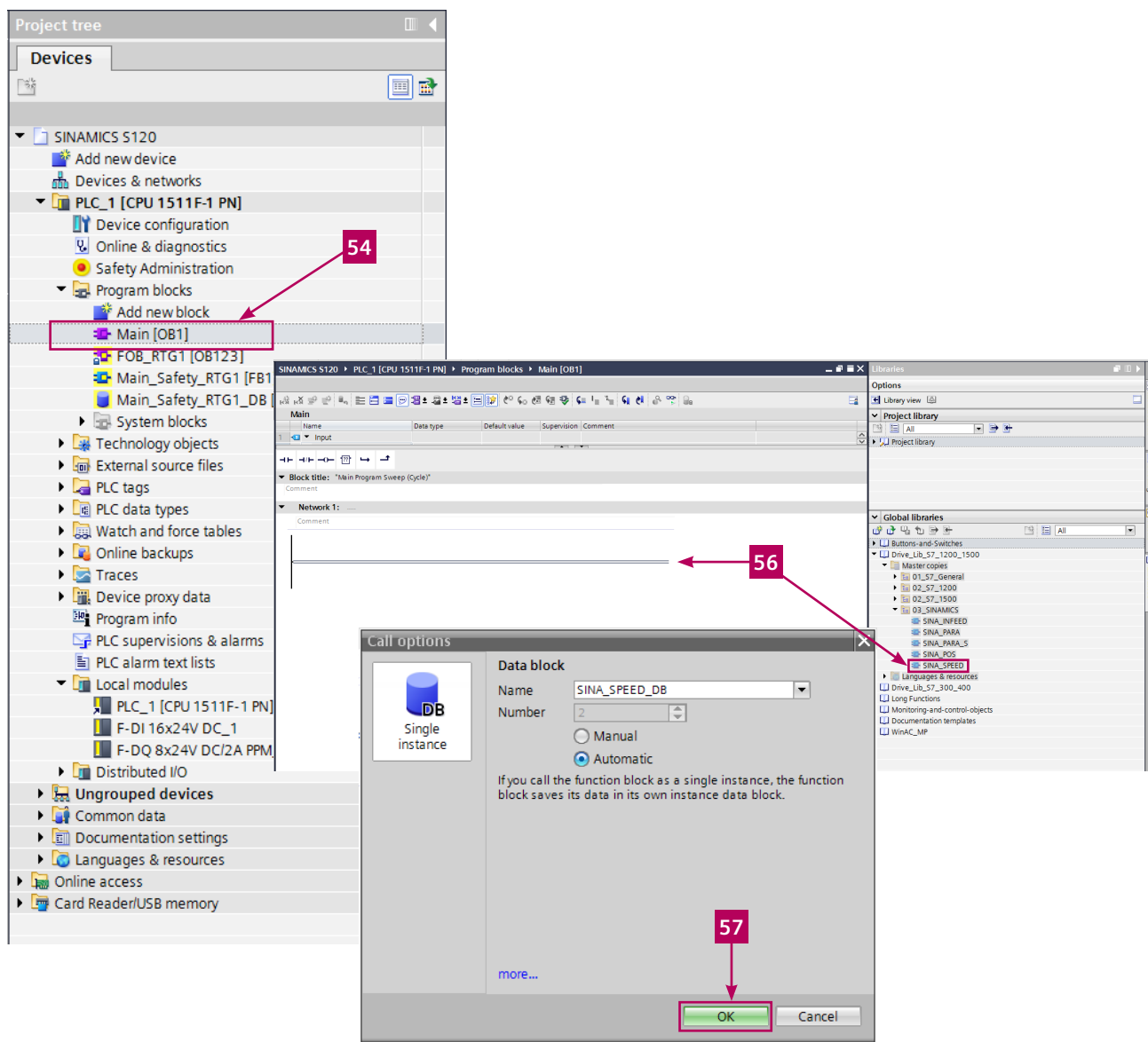
Protect the coded blocks F0B1_C (FB32774)

Compiling configuration

Time

Path	Description	Time
PLC_1	PLC_1 does not contain a configured protection level	1:27:22 PM
Hardware configuration	S7-1500RT200MP statio	1:27:22 PM
PLC_1	PLC_1	1:27:23 PM
PLC_1	PLC_1	1:27:23 PM
PLC_1	PLC_1 does not contain a configured protection level	1:27:23 PM
CPU display_1	The S7-1500 CPU display does not contain any password protection	1:27:23 PM
Safety	Compile safety program 'Safety Administration'.	1:27:24 PM
PLC data types		1:27:25 PM
Program blocks		1:27:26 PM
System blocks		1:27:26 PM
S7-1500 Safety		1:27:26 PM
Compiler blocks		1:27:26 PM
F_PS_OUT_1_...	Block was successfully compiled.	1:27:26 PM
F_PS_IN_2_0...	Block was successfully compiled.	1:27:26 PM
F-IO data blocks		1:27:26 PM
F00000_F-DQ	Block was successfully compiled.	1:27:26 PM
F00000_F-DI...	Block was successfully compiled.	1:27:26 PM
F_SystemInfo_DB...	Block was successfully compiled.	1:27:26 PM
RTG1SysInfo (DB...	Block was successfully compiled.	1:27:26 PM
Main_Safety_RTG1 (PB1)	Block was successfully compiled.	1:27:26 PM
Main_Safety_RTG1_DB (...	Block was successfully compiled.	1:27:26 PM
Main (OB1)	Block was successfully compiled.	1:27:28 PM

Dodatkowo wyłączamy ochronę funkcji lub wprowadzamy hasło dostępu (52). Następnie przechodzimy do kompilacji projektu (53) – brak błędów.

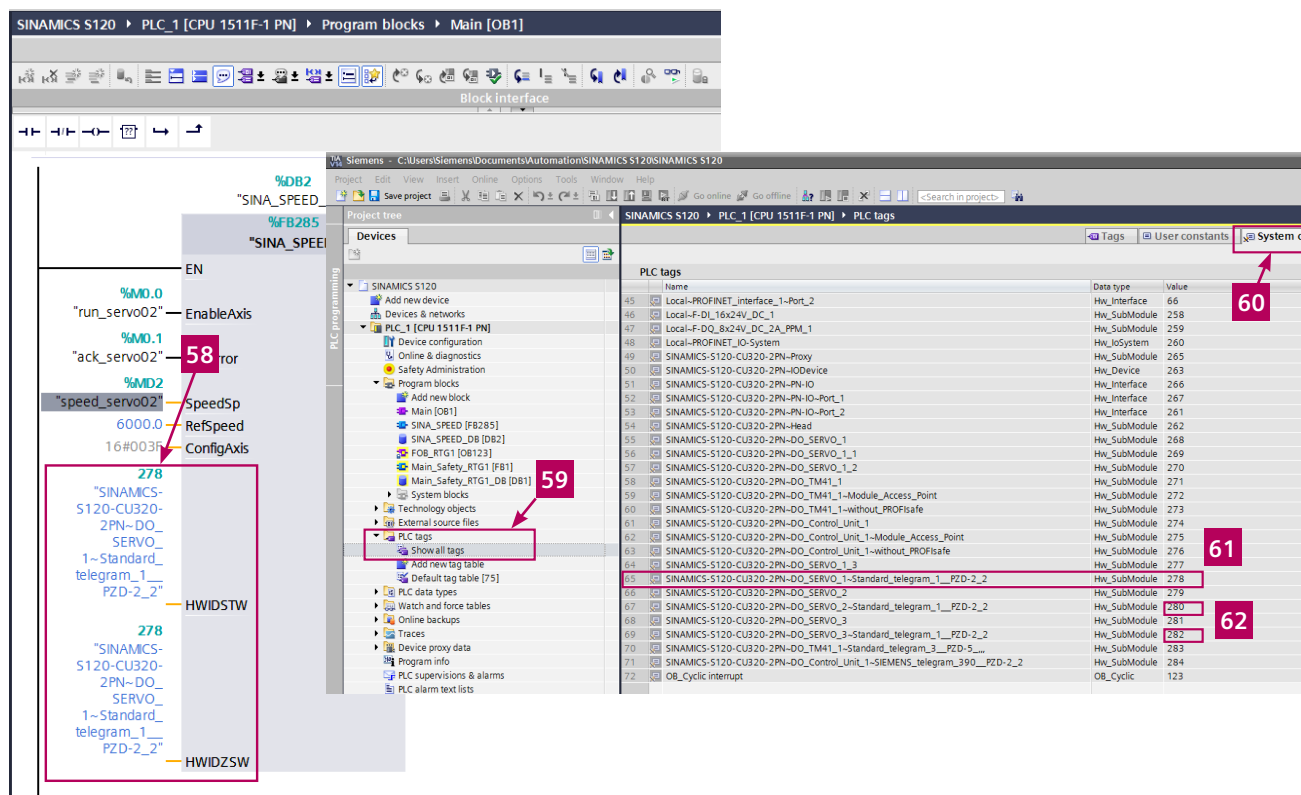


Sterowanie pracą przekształtnika częstotliwości rozpoczniemy od osi:

DO SERVO_1 → SERVO_02

W drzewie projektu TIA przechodzimy do menu „**Program blocks**” a następnie klikamy dwukrotnie w blok **OB1 „Main”** (54).

Z zakładki biblioteki „**Libraries**” (55) otwieramy bibliotekę napędową „**Drive_Lib_S7_1200_1500**”, następnie z folderu „**03_SINAMICS**” wybieramy bloczek „**SINA_SPEED**” który przeciągamy do okna głównego programu umieszczając go w sieci Network 1 (56). Do bloku sterowania tworzony jest blok danych **DB**, którego nazwę oraz numerację akceptujemy kliknięciem w przycisk „**OK**” (57).



Blok komunikacyjny **SINA_SPEED** łączymy z hardwarowym adresem telegramu komunikacyjnego przypisanego do sterowania wybranym urządzeniem (58).

Adresy HW poszczególnych elementów składowych ramki komunikacyjnej znajdują się w menu **PLC Tags** (59) – wybieramy polecenie **Show all tags**. W oknie głównym TIA przechodzimy do zakładki „**System constants**” (60) a następnie odszukujemy interesujące nas obiekty.

Zgodnie z przyjętym nazewnictwem oś **SERVO_02** „**STARTER**” (61):

DO SERVO_1 → 278

Pozostałe osie dostępne są z adresami HW (62):

DO SERVO_2 → 280

DO SERVO_3 → 282

Opis wejść/wyjść bloku **SINA_SPEED**:

WHIDSTW – adres związany z kanałem wyjść (sterowaniem pracą falownika) Q

HWIDZSW – adres związany z kanałem wejść (statusem pracy falownika) I

Pozostałe wejścia/wyjścia bloku **SINA_SPEED**:

EnableAxis – polecenie ON/OFF

AckError – kwitowanie błędów (operacja działająca na zbocze narastające się)

SpeedSP – wartość zadana prędkości w rpm

RefSpeed – wartość referencyjna prędkości (wartość P2000 odpowiedniej osi odczytana poprzez program STARTER)

AxisEnabled – oś pracuje

Lockout – 1 = brak możliwości załączenia (switching on inhibited)

ConfigAxis	Znaczenie	PZD	podłączona funkcja w przekształtniku	Wartość domyślna
Bit0	OFF2 - wybieg	1	r2090.1 = p 844[0]	1
Bit1	OFF3 – niezależna rampa hamowania	1	r2090.2 = p 848[0]	1
Bit2	Zwolnienie przekształtnika	1	r2090.3 = p 852[0]	1
Bit3	Zwolnienie generatora rampy	1	r2090.4 = p1140[0]	1
Bit4	Praca generatora rampy	1	r2090.5 = p1141[0]	1
Bit5	Zwolnienie wartości zadanej prędkości	1	r2090.6 = p1142[0]	1
Bit6	Wybór kierunku pracy	1	r2090.11 = p1113[0]	0
Bit7	Bezwarunkowe otwarcie hamulca	1	r2090.12 = p855[0]	0
Bit8	MOP zwiększanie prędkości	1	r2090.13 = p1035[0]	0
Bit9	MOP zmniejszanie prędkości	1	r2090.14 = p1036[0]	0
Bit10	Rezerwa – wolny (bit 8 PZD 1)	1	r2091.0	0
Bit11	Rezerwa – wolny (bit 9 PZD 1)	1	r2091.1	0
Bit12	Rezerwa – wolny (bit 15 PZD 1)	1	r2091.7	0
Bit13	0			
Bit14	0			
Bit15	0			

ActVelocity – aktualna prędkość wyjściowa w rpm

Error – błąd sterowania osią

Status – Status osi:

16#7002: No error – block is being processed

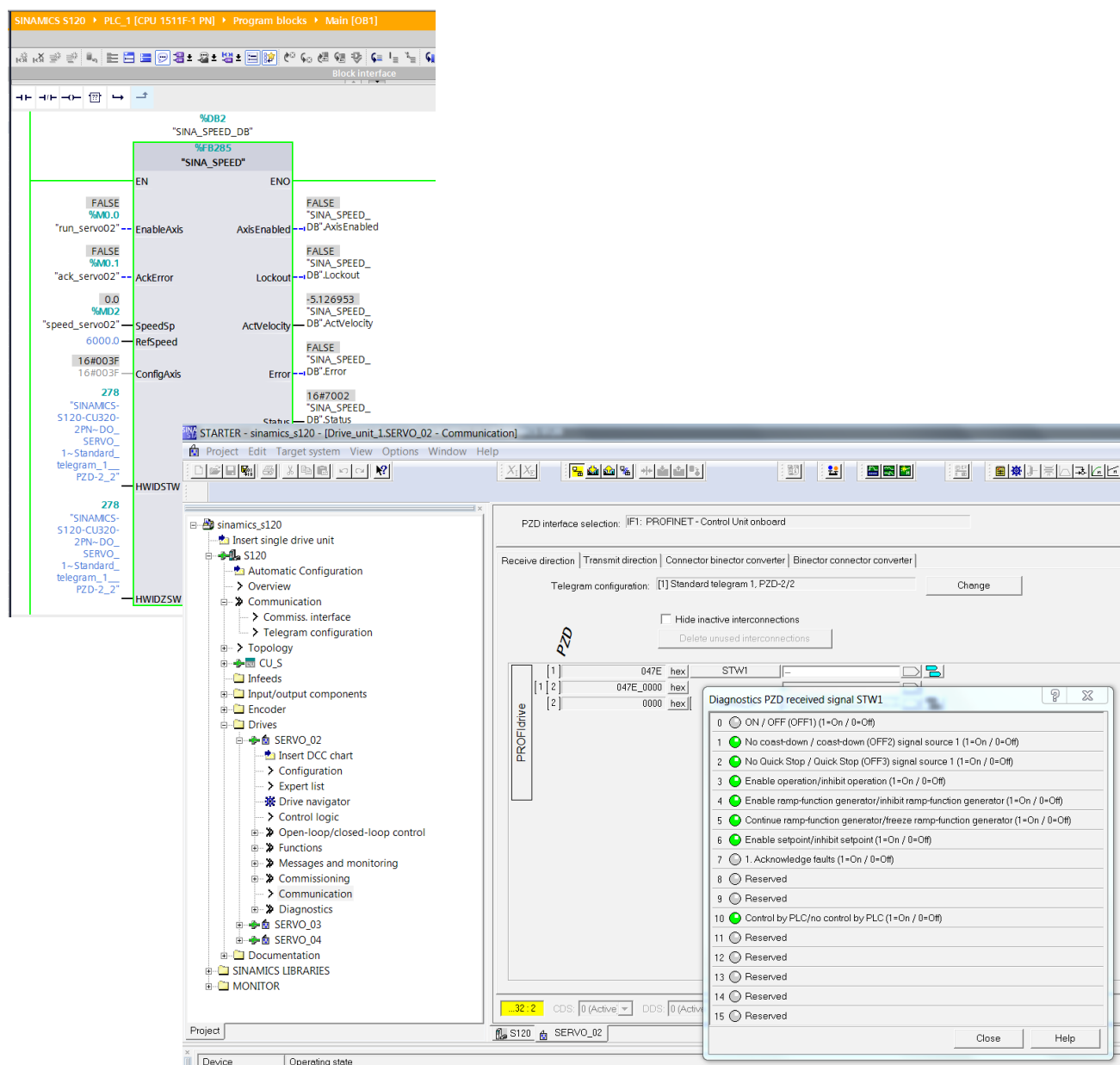
16#8401: Fault in the drive

16#8402: Switching-on inhibit

16#8600: Error DPRD_DAT

16#8601: Error DPWR_DAT

DiagId – rozszerzony błąd komunikacyjny



Po kompilacji projektu i wgraniu bieżącej wersji do PLC przechodzimy do widoku **online**. Prawidłowość wysyłanych oraz otrzymywanych sygnałów sterujących/statusowych w przekształtniku częstotliwości możemy skontrolować poprzez program STARTER. W widoku TIA

możemy dokonać modyfikacji słowa sterującego oraz zmienić wartość zadaną prędkości. Nastawione stany mogą być następnie odczytane w programie STARTER. W czytelny sposób wszystkie informacje wysyłane oraz nadawane do systemu nadrzędnego sterowania mogą być walidowane.

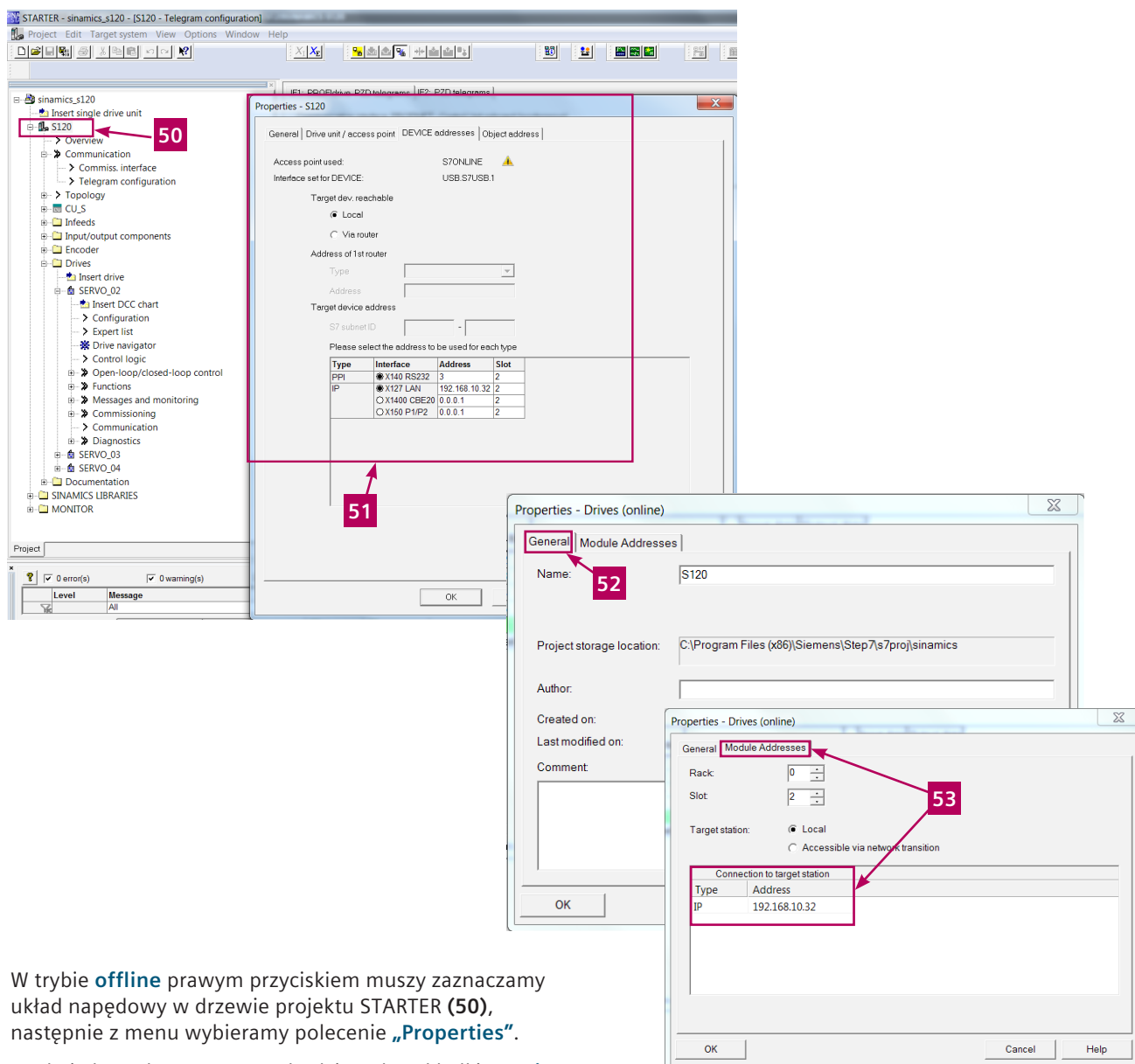
Zmiana interfejsu komunikacyjnego X127 na PN

Jednostki sterujące SINAMICS S120 posiadają dedykowany interfejs uruchomieniowy – złącze **X127**. Złącze to posiada fabrycznie nadany adres oraz nazwę **PROFINET**.

Jeżeli jednocześnie chcemy podglądać status układu napędowego – podłączonego po stacji PC poprzez złącze X127 oraz stan sterownika PLC, nasza stacja robocza musi posiadać dwie karty sieciowe.

Zdecydowanie łatwiejszym rozwiązaniem jest przełączenie dostępu komunikacyjnego przekształtnika częstotliwości na punkt sieciowy – w jednej sieci widoczny będzie sterownik oraz przekształtnik częstotliwości.

Zmiana sposobu dostępu sieciowego do przekształtnika częstotliwości wymaga ręcznego określenia sposobu dostępu.



W trybie **offline** prawym przyciskiem muszy zaznaczamy układ napędowy w drzewie projektu STARTER (50), następnie z menu wybieramy polecenie „**Properties**”.

W oknie kontekstowym przechodzimy do zakładki „**Device addresses**” (51) w którym możemy odczytać bieżące nastawy komunikacji z układem napędowym.

Dokonujemy modyfikacji punktu dostępowego z **X127** na **X150P1/P2** wprowadzając sieciowy adres IP przekształtnika częstotliwości, zamykamy okno kontekstowe przyciskiem **OK**.

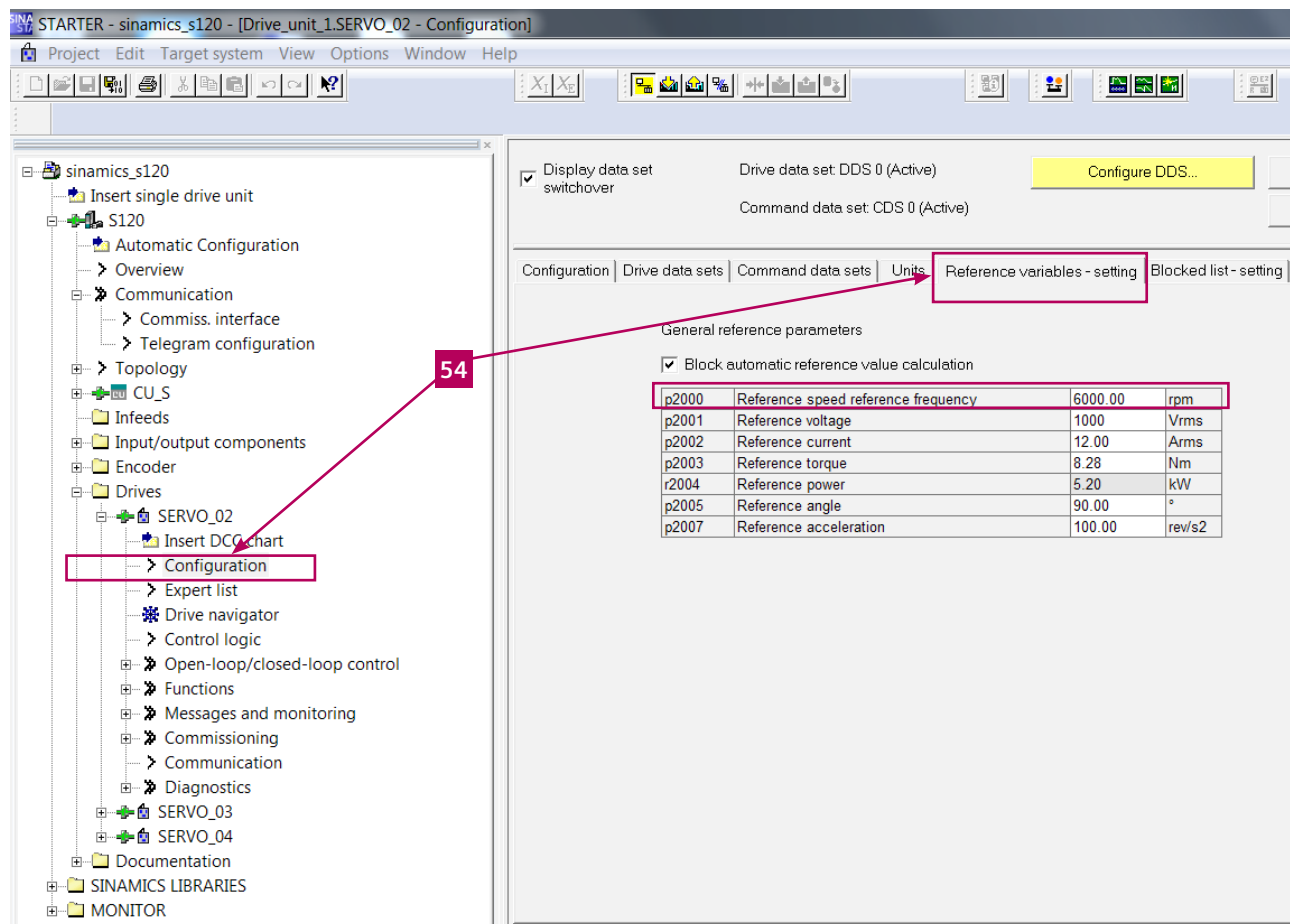
Następnie ponownie prawym przyciskiem myszy klikamy w nazwę układu napędowego w drzewie projektowym w wyświetlonym menu przechodzimy do opcji „**Target**

device” i wybieramy polecenie „**Online access**”.

W zakładce „**General**” widoczna jest nazwa urządzenia (52), adres sieciowy definiujemy w zakładce „**Module addresses**” 53 – adres musi być zgodny z adresem sieciowym urządzenia.

Zmiany jeśli występują akceptujemy przyciskiem **OK**.

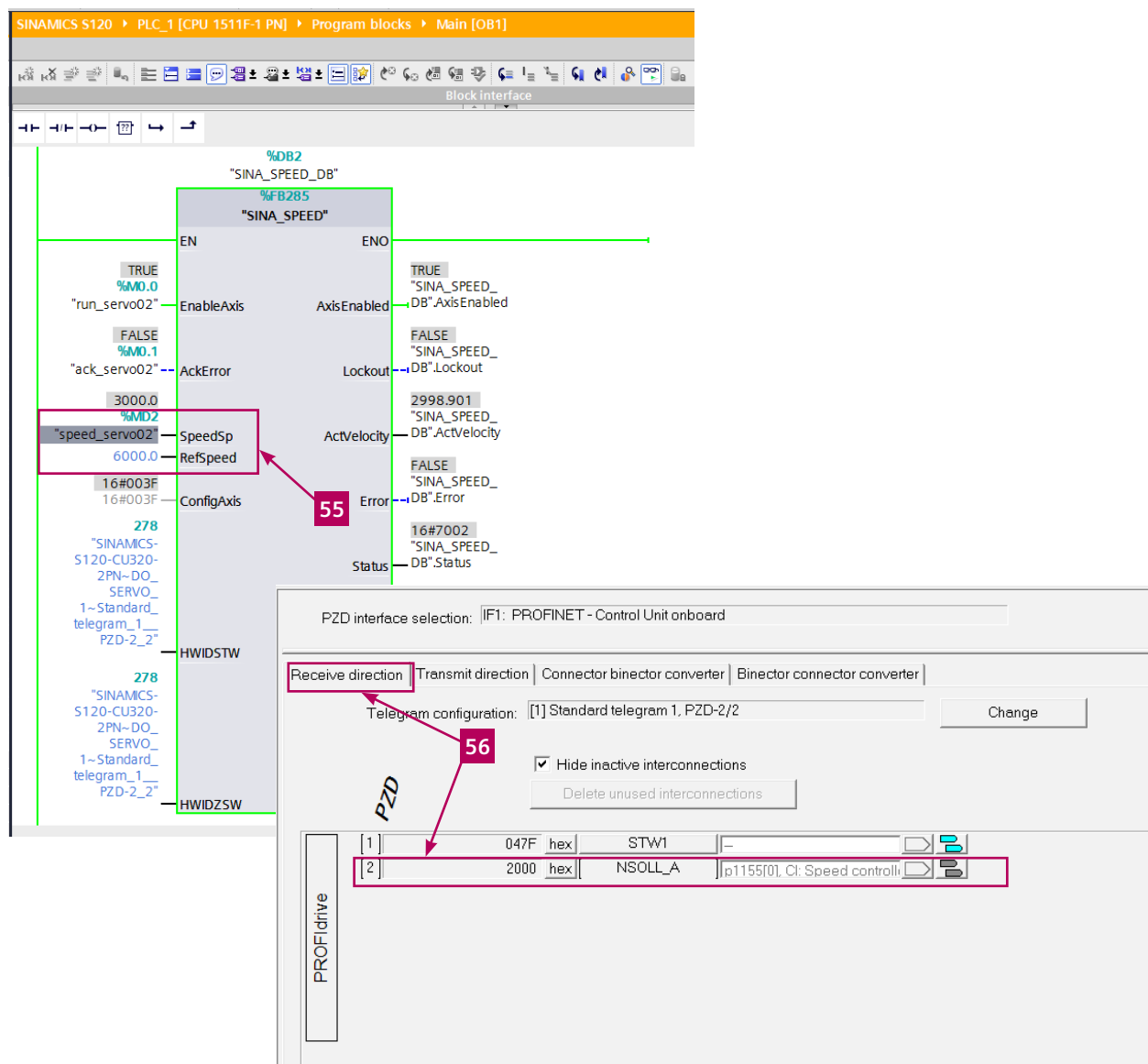
Przechodzimy do widoku **online** – pracując na magistrali sieciowej (wspólnej dla PLC i napędu).



Wartość referencyjną **Ref_speed** którą posługuje się błądzek komunikacyjny **Sina_Speed** dla każdego z napędów należy odczytać indywidualnie. W drzewie projektowym przechodzi do widoku falownika, następnie wybieramy polecenie „**Configuration**” (54). W oknie główny programu zmieniamy widok na „**Reference variables - settings**”, odczytujemy wartość referencyjną prędkości = 6000rpm.

Zadając poprzez magistralę komunikacyjną wartość **4000HEX** układ napędowy otrzymuje 100% sygnału = 6000rpm.

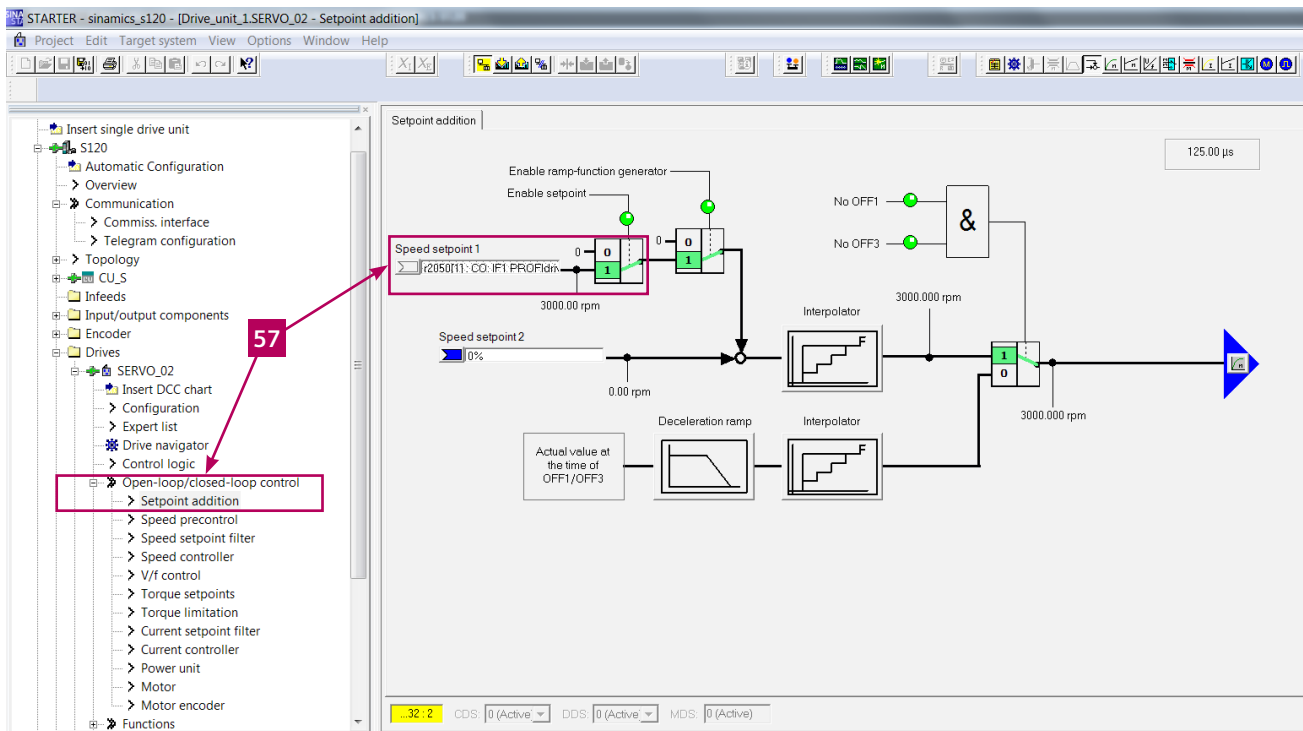
Podobnie w przypadku odczytu wartości aktualnej – przekształtnik do PLC wyśle wartość w zakresie 0 – 4000HEX która odpowiada prędkości z zakresu 0 – 6000rpm.



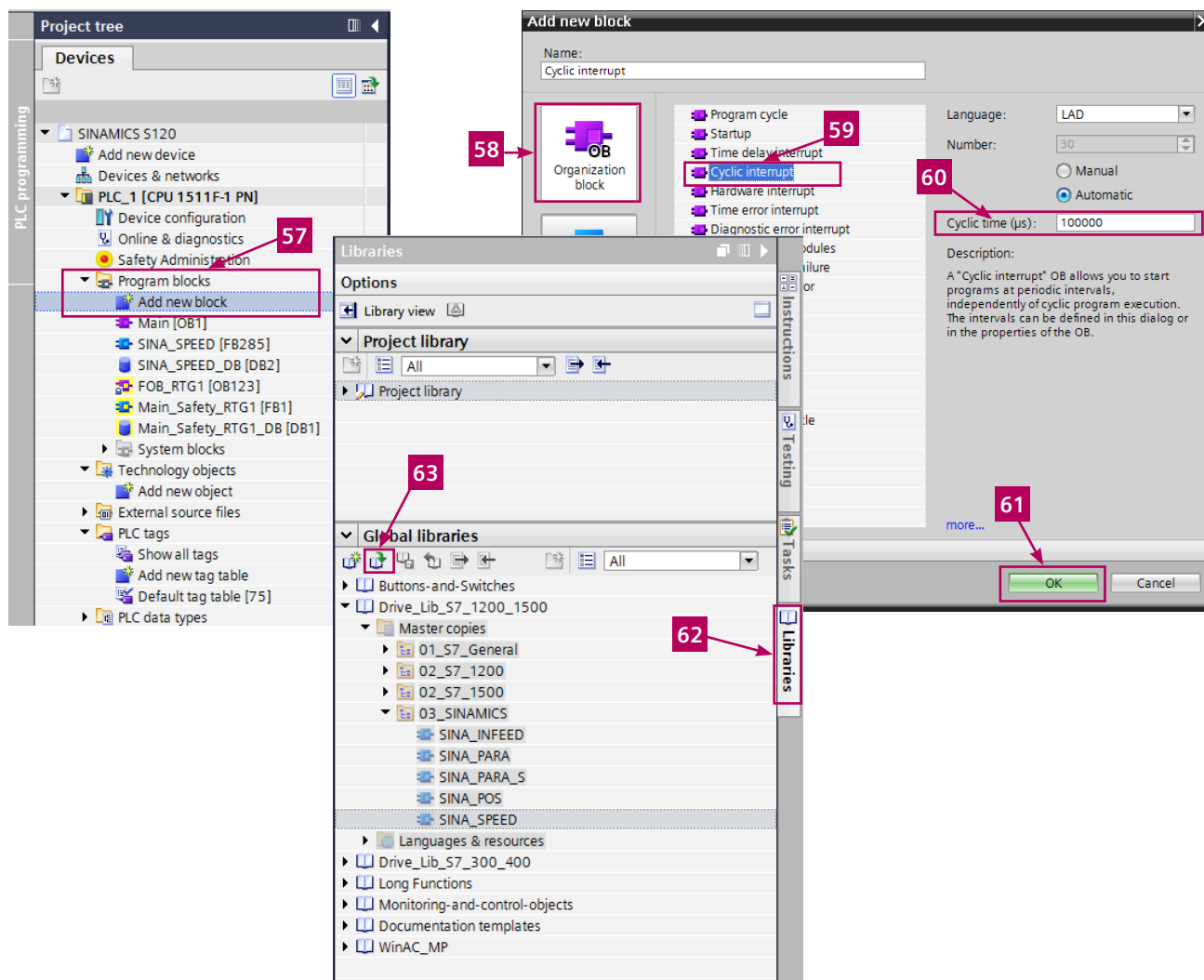
Odczytana wartość referencyjna prędkości wprowadzamy jako RefSpeed (55).

Poprzez SpeedSp możemy zadawać prędkość – prawidłowość wysyłanych sygnałów odczytujemy w programie STARTER, menu Communication zakładka „Recieve direction”(56).

W przykładzie zadajemy wartość $3000\text{rpm} = 2000\text{Hex}$ (50% wartości referencyjnej).



Dalsza diagnostyka sygnału wartości zadanej możliwa jest menu „**Setpoint addition**” w którym wyświetlona zostanie wartość wyskalowana przesyłanego sygnału 3000rpm (57).

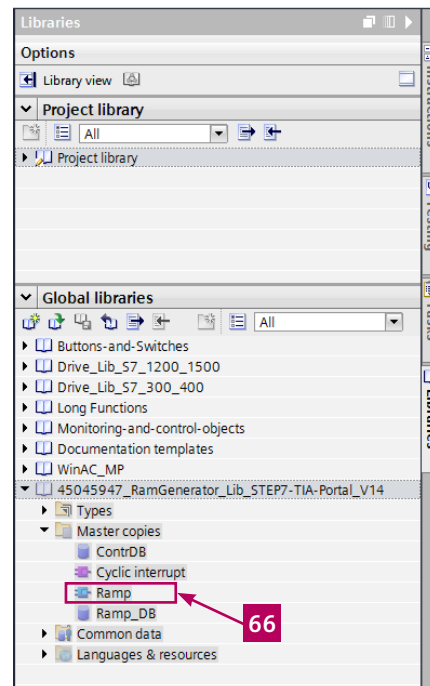
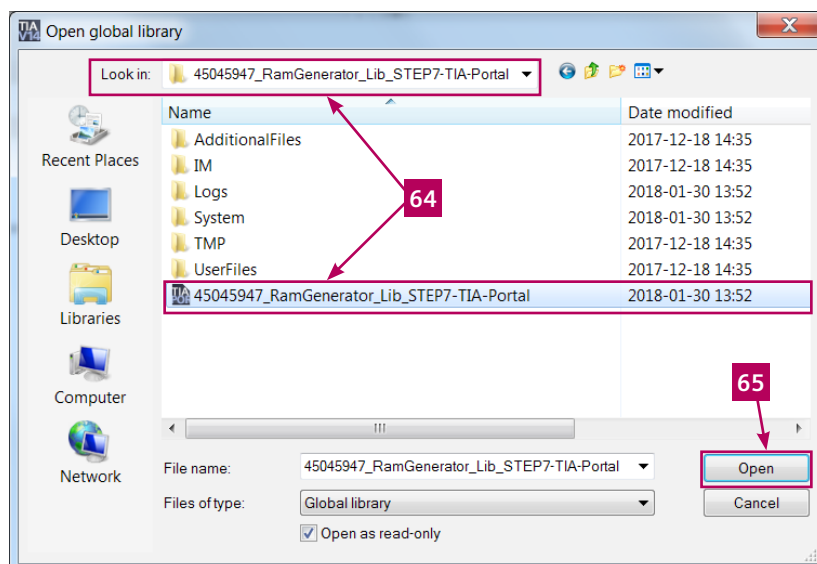


W trybie servo układ napędowy nie posiada generatora rampy. Moduł technologiczny generatora jest nieaktywny – możemy go oczywiście włączyć poprzez parametryzację napędu (ponowne wykonanie procedury szybkiego uruchomienia). W naszym przykładzie zadajnik rampy generowany będzie w PLC. Istnieje gotowy przykład aplikacyjny uwzględniający bibliotekę „[RampGenerator_Lib](#)” – biblioteka oraz opis w załączniku do wersji elektronicznej niniejszej dokumentacji.

Zadajnik generatora rampy umieszczony zostanie w bloku cyklicznych przerwań OB30 w tym celu z drzewa projektu TIA wybieramy polecenie „[Add new block](#)” (57). W

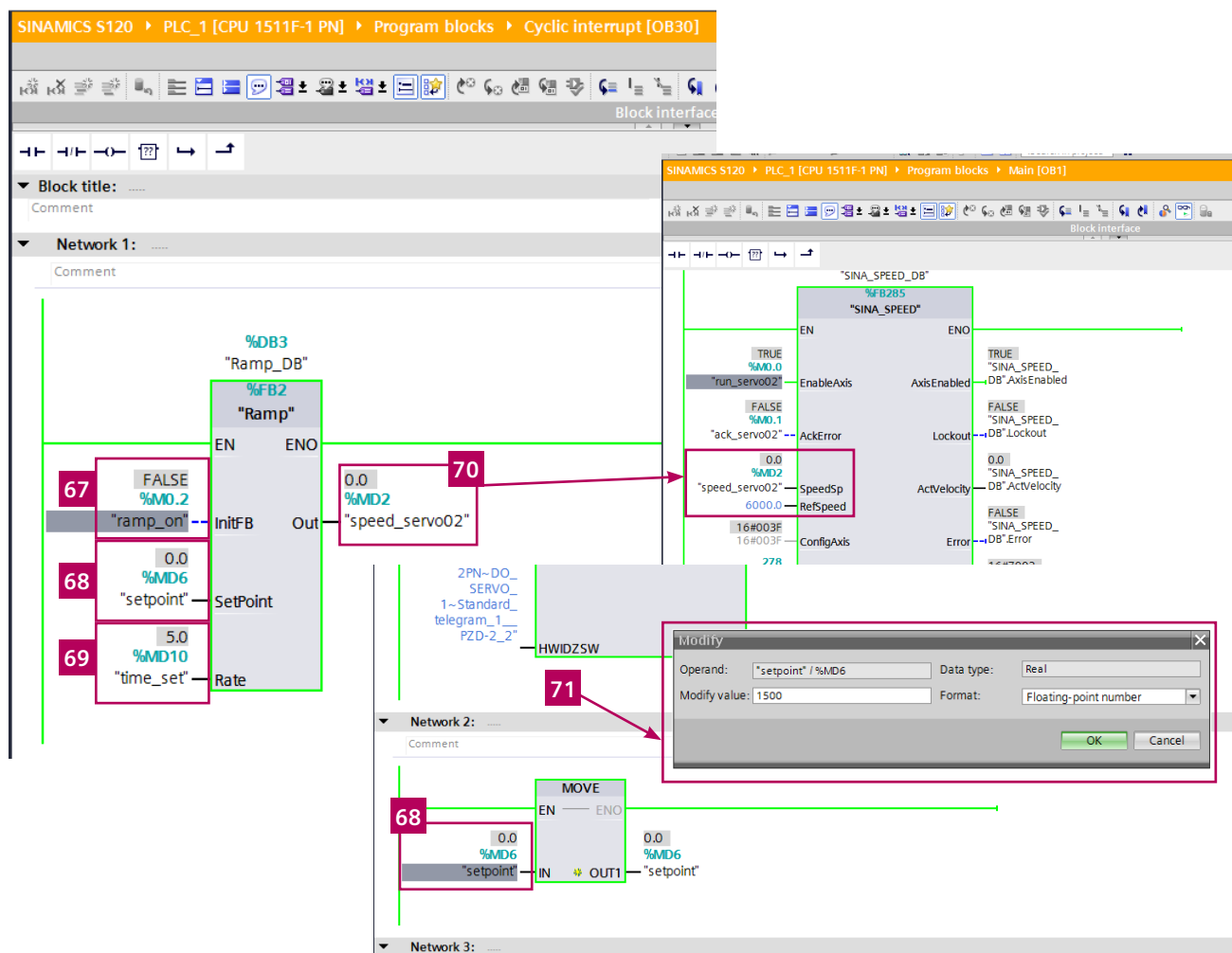
wyświetlonym oknie kontekstowym zaznaczamy typ bloku **OB** (58) oraz jego rodzaj „**Cyclic interrupt**” (59). Następnie wprowadzamy cykl odświeżania procesu – istotny z punktu widzenia przyrostowego zadawania wartości prędkości (60). Na zakończenie wybór oraz konfigurację akceptujemy przyciskiem **OK**. (61)

W kolejnym kroku otwieramy właściwą bibliotekę programu TIA – przechodzimy do zakładki **Libraries** (62) i klikamy w ikonę otwórz bibliotekę (63).



Wskazujemy właściwą lokalizację biblioteki oraz plik **RamGenerator_Lib...** (64) który otwieramy przyciskiem „Open” (65).

W widoku bibliotek TIA przechodzimy do folderu „Master copies” z którego przeciągamy bloczek **RAMP** (66) do widoku sieci bloku **OB30**.



InitFB (67) – inicjalizacja nastaw zadajnika rampy (TRUE), podczas pracy (FALSE)

Setpoint (68) – Wartość zadana prędkości (w naszym projekcie jako Real zmienna od 0 do 6000rpm)

Rate (69) - przykład:

Czas odświeżania bloku OB30 – 0,1s

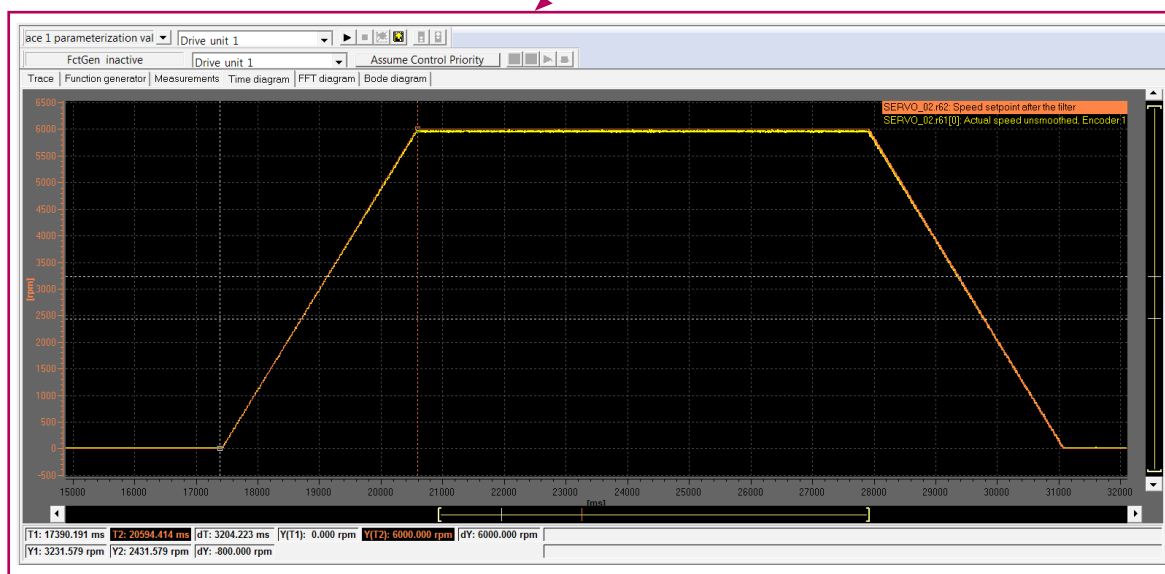
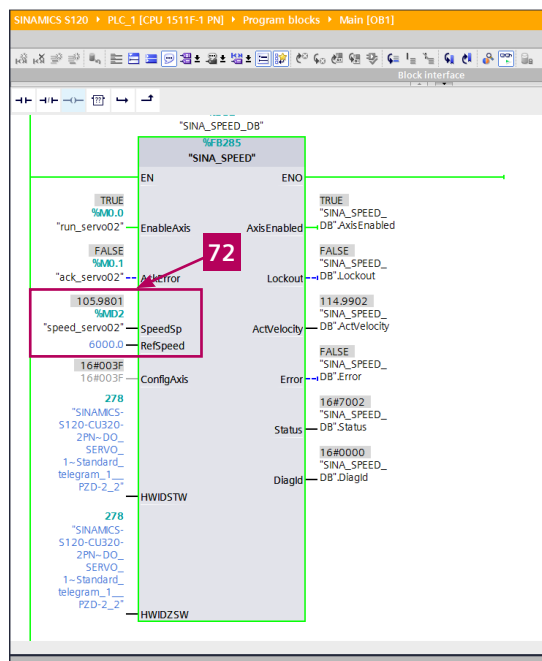
Rate = wartość zadana/s → jeżeli wartość zadana = 1500rpm dla rate = 1/s czas przyspieszania od 0 do 1500rpm = 1500s

Dla **Rate** = 5 czas przyspieszania = 1500/5 = 300s

Out (70) – wartość wyjściowa generatora rampy, podłączona zostaje jako wartość **SpeedSp** bloku komunikacyjnego **SINA_SPEED**.

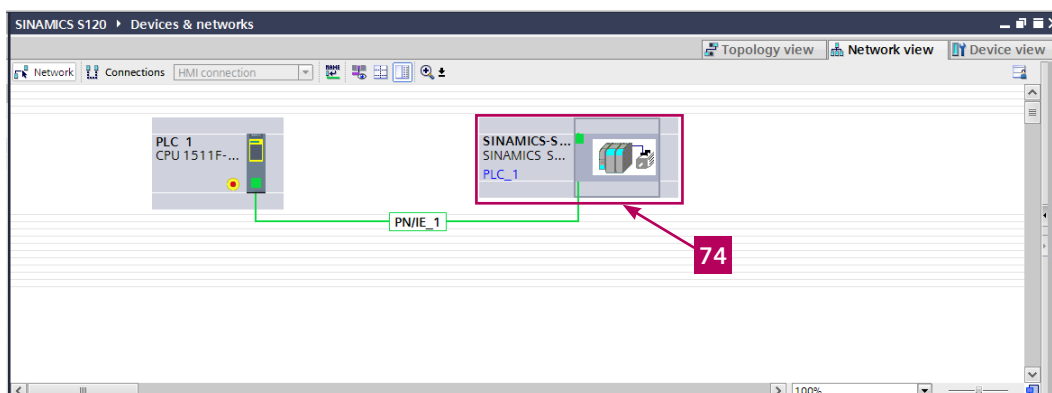
W celu ułatwienia pracy zmienna setpoint zostaje umieszczona w bloku OB1 – bloczek MOVE pozwoli na sterowanie wartością zadaną z jednego miejsca programu.

Poprzez modyfikację zmiennej „setpoint” (71) wpływamy na wartość zadaną prędkości przetwarzaną przez zewnętrzny generator rampy.



Po odpowiednim dopasowaniu wartości Rate możemy przystąpić do sprawdzenia pracy układu napędowego. Wartość zadaną wprowadzamy poprzez zmienną „setpoint” – wyjście bloku Ramp obserwujemy w polu 72.

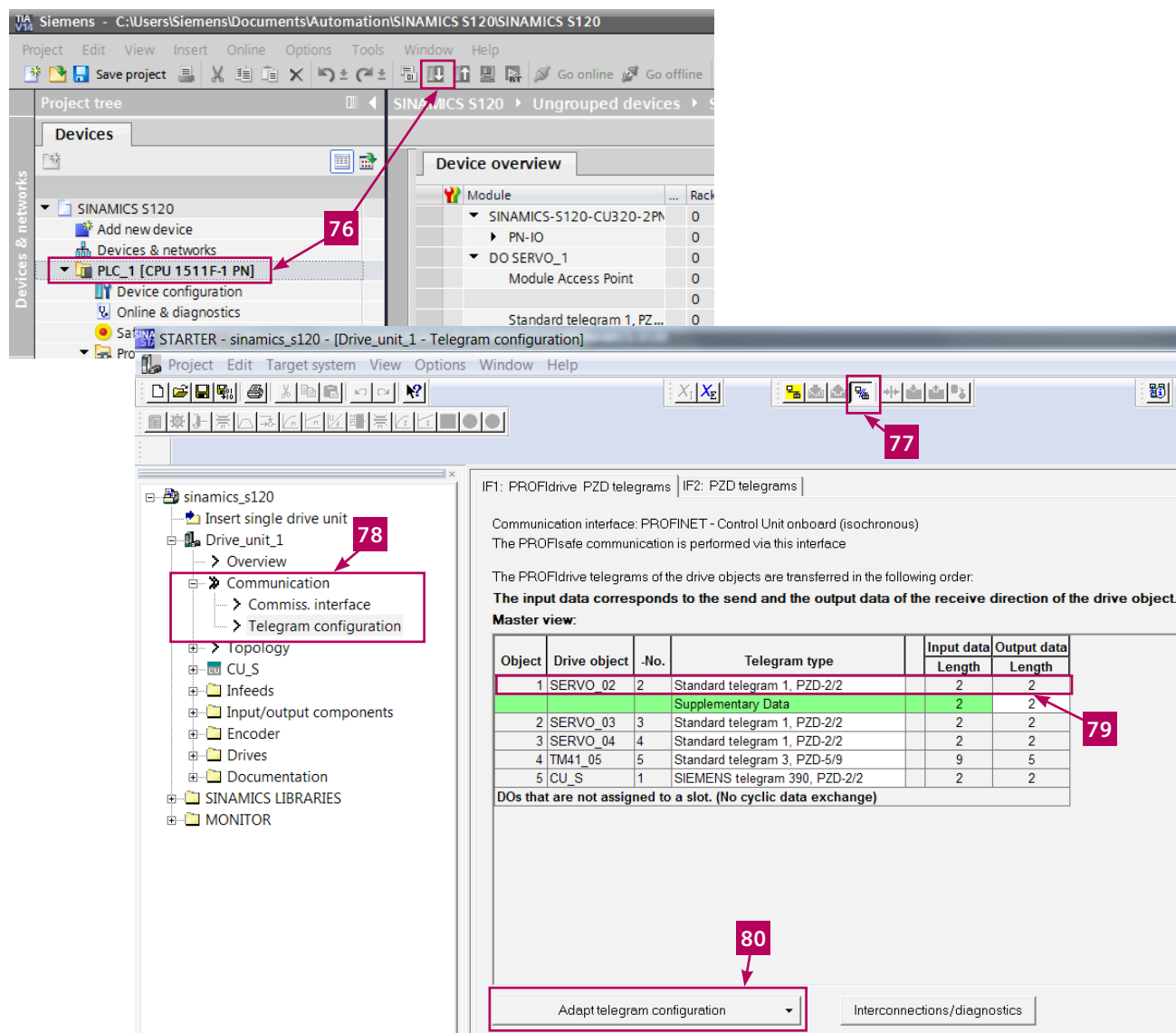
Przebiegi wartości zadanej oraz wyjściowej możemy zarejestrować poprzez funkcję **Trace** programu **STARTER** (73).



Module	Rack	Slot	I address	Q address	Type	Article number	Firmware	Comment	Access
SINAMICS-S120-CU320-2PN	0	0			SINAMICS S120/S1...	6SL3 040-1MA01-0AAx	V4.70		PLC_1
PN-IO	0	0	X150		SINAMICS-S120-CU...				PLC_1
DO SERVO_1	0	1			DO SERVO				PLC_1
Module Access Point	0	1.1			Module Access Point				PLC_1
Module Access Point	0	1.2			Module Access Point				PLC_1
Standard telegram 1, PZ...	0	1.3	15...18	5...8	Standard telegram ...				PLC_1
Supplementary data, PZ...	0	1.4	49...52	37...40	Supplementary dat...				PLC_1
DO SERVO_2	0	2			DO SERVO				PLC_1
Module Access Point	0	2.1			Module Access Point				PLC_1
Module Access Point	0	2.2			Module Access Point				PLC_1
Standard telegram 1, PZ...	0	2.3	19...22	15...18	Standard telegram ...				PLC_1
Module Access Point	0	2.4			Module Access Point				PLC_1
DO SERVO_3	0	3			DO SERVO				PLC_1
Module Access Point	0	3.1			Module Access Point				PLC_1
Module Access Point	0	3.2			Module Access Point				PLC_1
Standard telegram 1, PZ...	0	3.3	23...26	19...22	Standard telegram ...				PLC_1
Module Access Point	0	3.4			Module Access Point				PLC_1
DO TM41_1	0	4			DO TM41				PLC_1
Module Access Point	0	4.1			Module Access Point				PLC_1
without PROFIsafe	0	4.2			without PROFIsafe				PLC_1
Standard telegram 3, PZ...	0	4.3	27...44	23...32	Standard telegram ...				PLC_1
DO Control Unit_1	0	5			DO Control Unit				PLC_1
Module Access Point	0	5.1			Module Access Point				PLC_1
without PROFIsafe	0	5.2			without PROFIsafe				PLC_1
SIEMENS telegram 390, P...	0	5.3	45...48	33...36	SIEMENS telegram ...				PLC_1
	0	6							
	0	7							
	0	8							
	0	9							
	0	10							
	0	11							

Jeżeli telegram **komunikacyjny nr 1** jest dla nas niewystarczający ze względu na ilość danych przesyłanych pomiędzy przekształtnikiem a PLC, możemy go rozbudować o kolejne słowa wysyłane poprzez proces komunikacji cyklicznej. W tym celu w trybie **offline TIA** przechodzimy do widoku sieci i urządzeń. Następnie poprzez dwukrotne kliknięcie w układ napędowy (74) przechodzimy do widoku urządzenia w którym dokonamy modyfikacji struktury telegramu komunikacyjnego.

Dla wybranego falownika (w naszym przykładzie **servo_1**) ramka komunikacyjna zawierająca **standardowy telegram nr1** została uzupełniona **dodatkowym telegramem** o strukturze 2/2 (75 dwa słowa nadawane oraz odbierane).



Wprowadzone zmiany wgrywamy do **PLC (76)**.

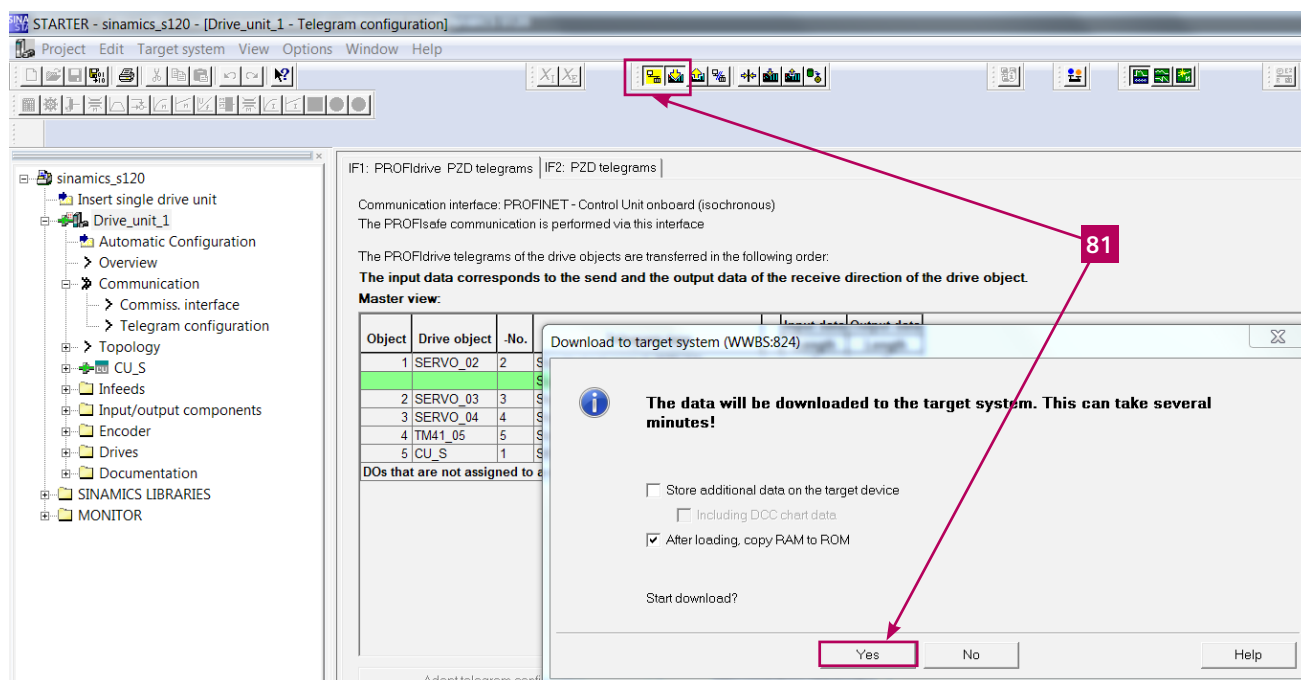
Modyfikacja struktury telegramu komunikacyjnego dodatkowo musi zostać odwzorowana po stronie układu napędowego. W tym celu w programie **STARTER** przechodzimy do pracy w trybie **offline (77)** – pamiętamy o wszystkich niezapisanych zmianach jeśli występują (kopiujemy pamięć ulotną do trwałej przekształtnika częstotliwości a następnie zapisujemy bieżącą wersję projektu z napędu na dysku twardym PC).

W kolejnym kroku przechodzimy w drzewie projektowym do widoku struktury telegramu komunikacyjnego **(78)**.

W oknie głównym programu **STARTER** zaznaczamy obiekt napędowy zdefiniowany poprzez właściwy telegram komunikacyjny **(79)**. Rozbudowa telegramu będzie możliwa poprzez kliknięcie w przycisk „**Adapt telegram configuration**” **(80)**.

Z wyświetlonego menu wybieramy właściwą opcję (w identyczny sposób do poszczególnych obiektów napędowych dodawane są telegramy bezpieczne).

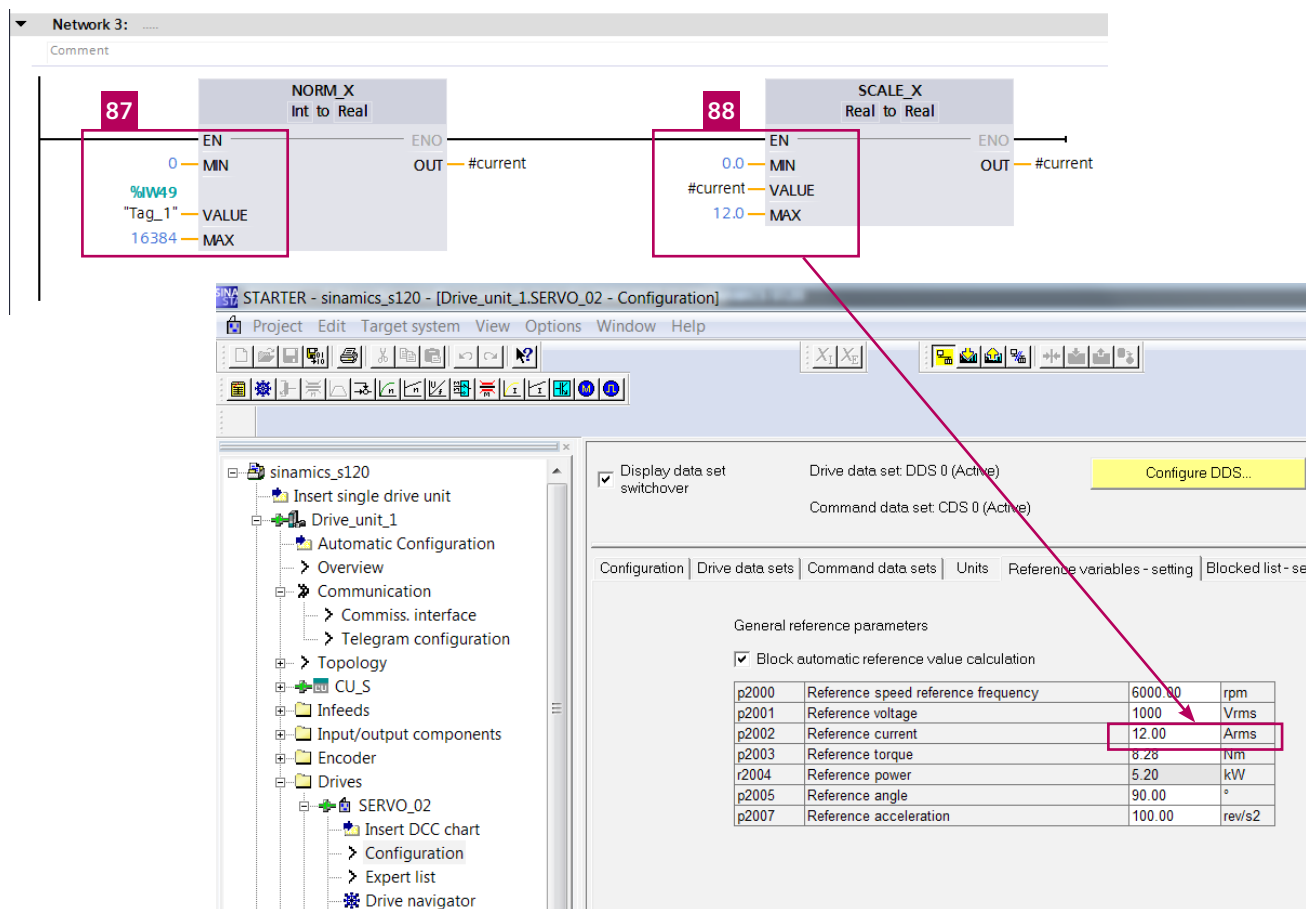
Na zakończenie definiujemy długość telegramu – kolumny **Length** w kanałach **Input** oraz **Output** data.



Zmiany związane z konfiguracją telegramu po przejściu do trybu **online** wgrzywamy do napędu (**81**) – polecenie **Download to target system**.

W wyświetlonym oknie kontekstowym możemy również zaznaczyć konieczność kopiowania pamięci ulotnej do trwałej po zakończeniu procesu wgrzywania nowych nastaw pracy przekształtnika częstotliwości.





Wartości odczytywane poprzez magistralę komunikacyjną obsługującą proces cyklicznej wymiany danych muszą być skalowane po stronie napędu oraz po stronie PLC.

Sama wymiana danych bazuje na zasadzie przesyłania wartości w formie int32 – przesyłamy wartość zmienną w zakresie 0 – 4000Hex co odpowiada wartości procesowej zmiennej od 0 do 100%.

Skalowanie po stronie PLC może być realizowane poprzez bloki **NORM_X** oraz **SCALE_X**.

NORM_X – normalizuje sygnał odczytywany poprzez magistralę. Wartość prądu w telegramach komunikacyjnym podłączono została do trzeciego słowa telegramu komunikacyjnego czyli pierwszego słowa rozszerzenia telegramu komunikacyjnego SERVO_1, kolumna wejść PLC i odpowiadająca adresowi IW49 (87).

Wartość minimalna = 0, maksymalna 4000HEX → 16384DEC.

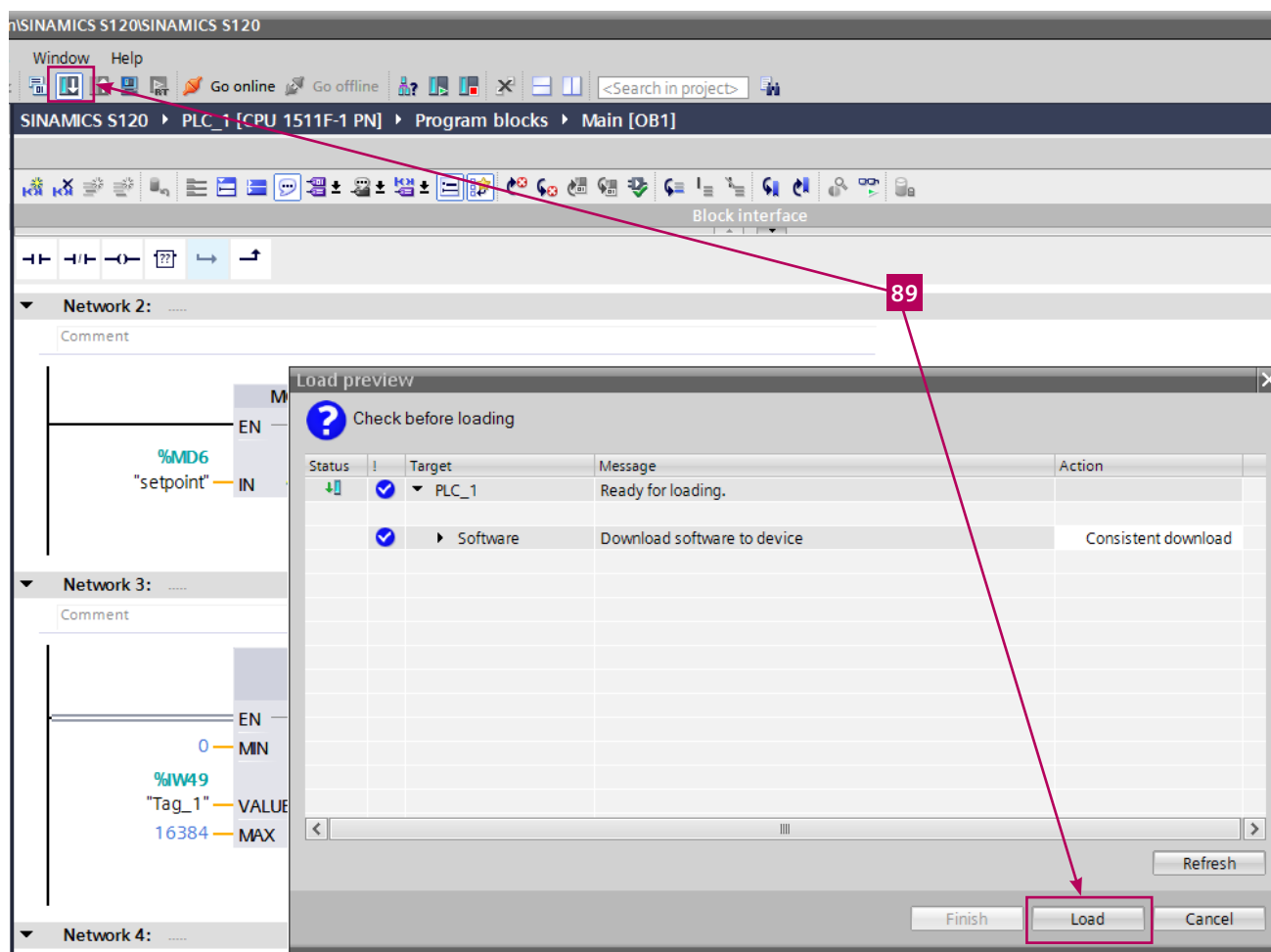
Wyjście bloku stanowi zmienna lokalna o nazwie „**current**”, która przyjmować będzie wartość z zakresu 0 - 1.

W przypadku bloku **SCALE_X** dokonamy właściwego skalowania otrzymywanego sygnału „**current**”.

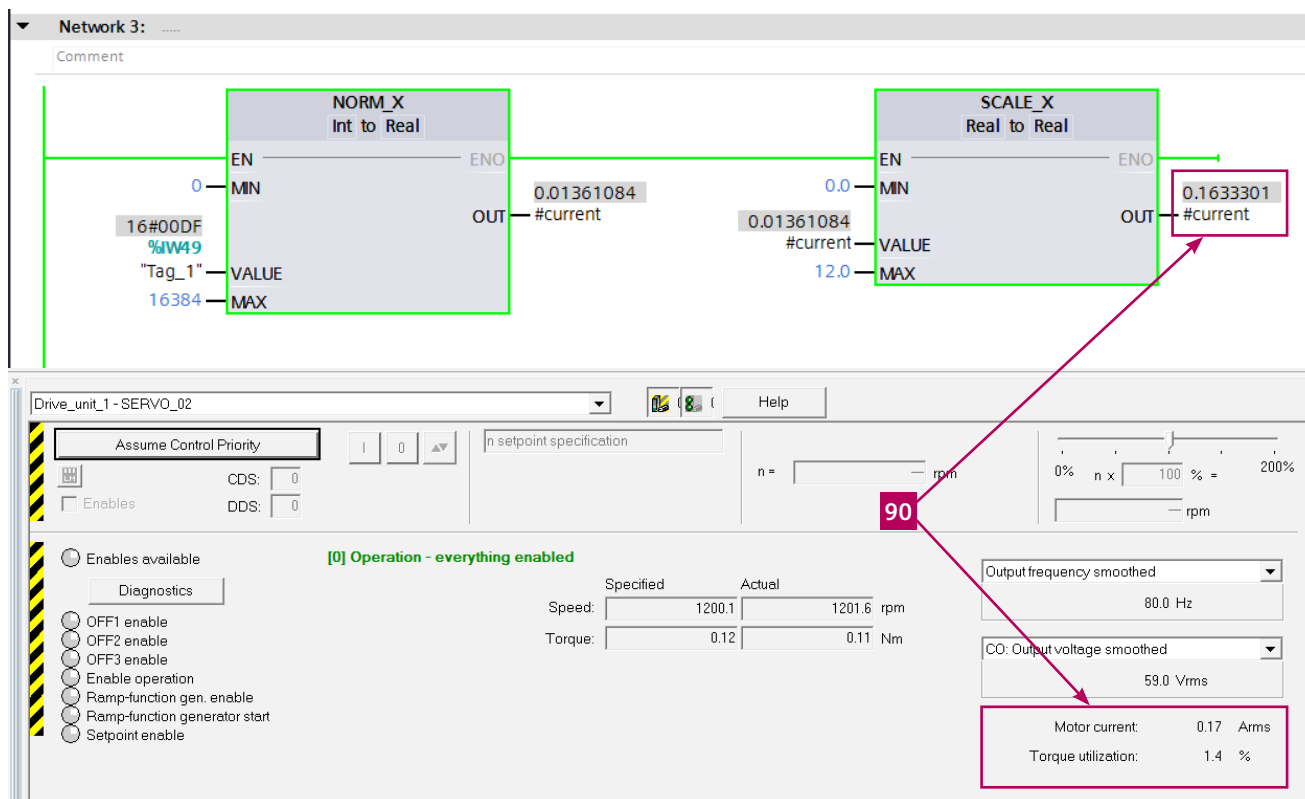
Odczytanie 0 odpowiadać będzie wartości 0A.

Natomiast odczytanie 1 stanowi wartość referencyjną dla odczytywanego sygnału – w naszym przypadku **12A** (88) (Wartości referencyjne znajdziemy w programie STARTER menu „Configuration” a następnie zakładka okna głównego programu „Reference variables - settings”).

Na wyjściu bloku **SCALE_X** otrzymywać będziemy aktualną wartość prądu wyjściowego wyrażoną w A.

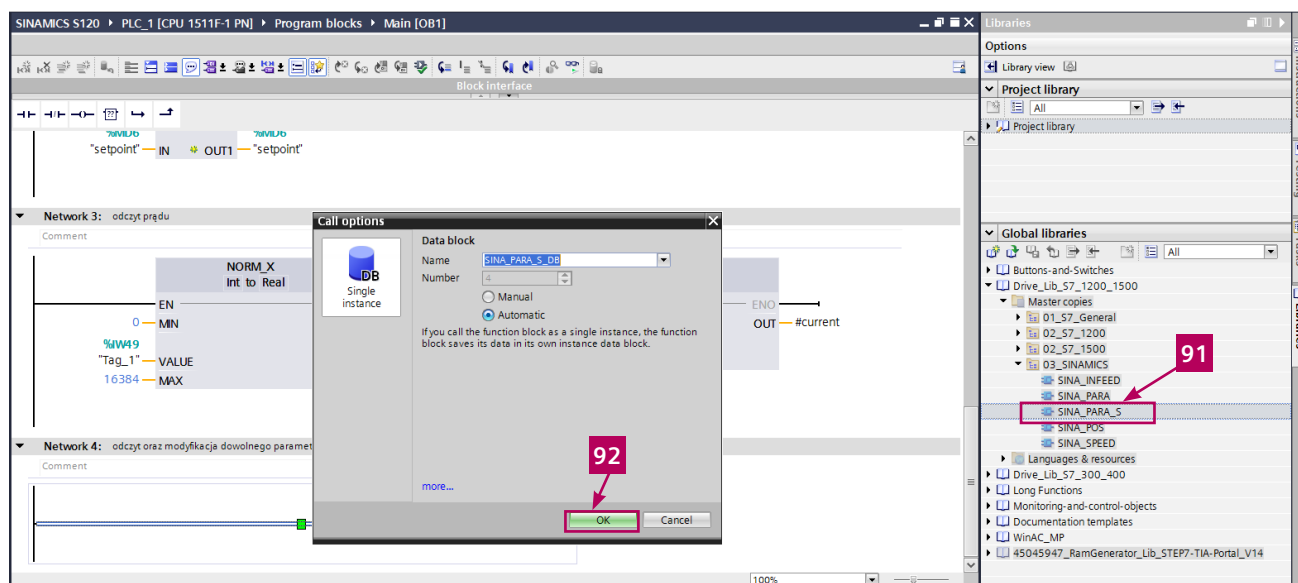


Zmodyfikowany projekt zapisujemy, kompilujemy i wgrywamy do PLC (89).



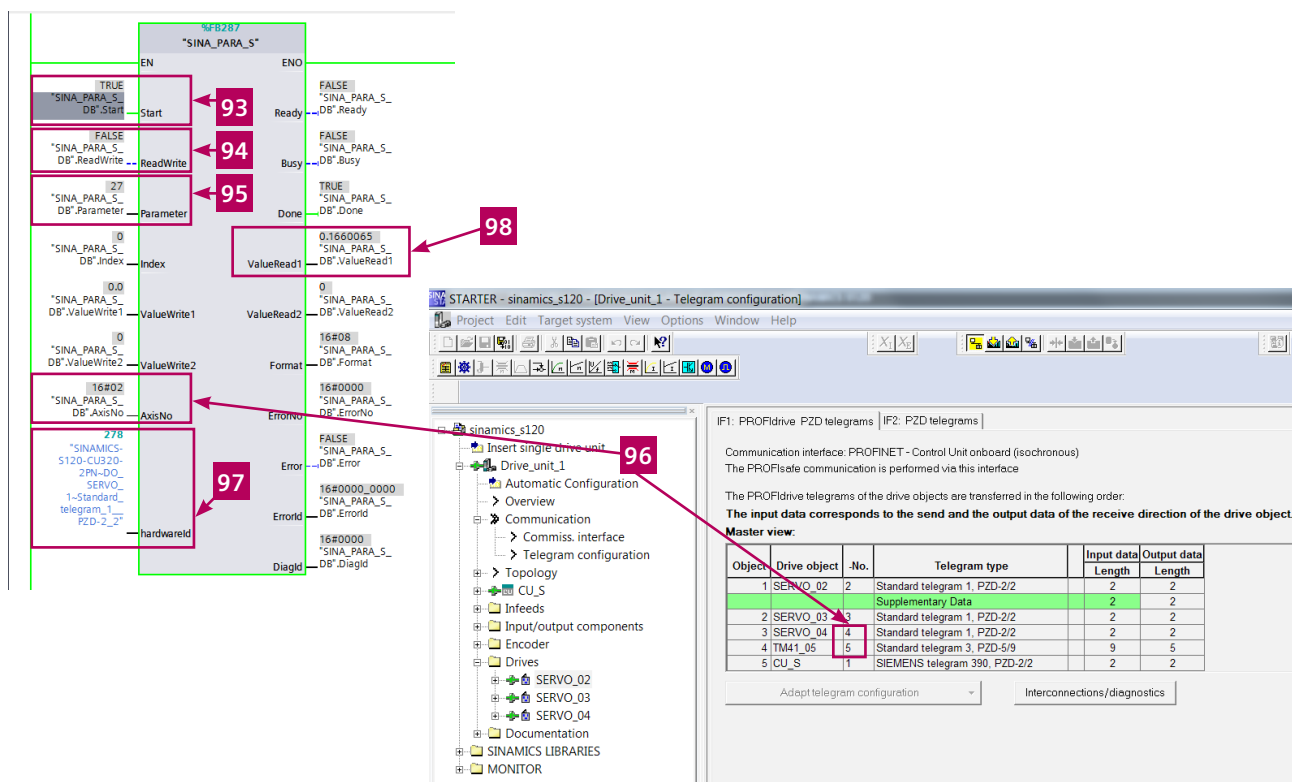
Następnie przechodzimy do podglądu bloku **OB1**,
załączamy przekształtnik do pracy, zadajemy prędkość.

W takim stanie możemy porównać wartości odczytywane
(90) – w programie STARTER oraz poprzez magistralę
komunikacyjną.



Alternatywną metodą odczytania oraz modyfikacji stanów poszczególnych parametrów przekształtnika częstotliwości jest biblioteka komunikacyjna **Drive_Lib**, w której znajdziemy bloki „**SINA_PARA_S**” oraz „**SINA_PARA**” (91).

W projekcie możemy umieścić blok o nazwie „**SINA_PARA_S**” dla którego tworzony jest również blok danych – numerację oraz nazwę akceptujemy przyciskiem „**OK**” (92).



Poszczególne wejścia oraz wyjścia bloku **SINA_PARA_S** podłączone zostały do zmiennych bloku danych.

Przypisanie bloku do współpracy z odpowiednim układem napędowym odbywa się poprzez wprowadzenie adresu HW telegramu komunikacyjnego.

W naszym przykładzie jest to oś o nazwie **SERVO_1** – definiowana poprzez telegram standardowy 1 oraz rozszerzenie telegramu (**97**).

Dla systemów wieloosiowych – **SINAMICS S120** istotnym parametrem definiującym określony obiekt napędowy jest również **AxisNo** – wprowadzamy tu wartość odczytaną w kolumnie No programu STARTER definiującą numer obiektu (**96**).

Pozostałe sygnały:

Start – wykonaj operację (odczyt/zapis) na zboczce narastające,

ReadWrite – 0 odczyt stanu parametru, 1 zmiana wartości parametru

Parameter – numer parametru odczytywanego lub modyfikowanego, w naszym przykładzie 27

Index – index parametru = 0

ValueWrite1 – wartość modyfikowana 1

ValueWrite2 – wartość modyfikowana 2 (inny format)

Odczytana wartość prądu widoczna będzie w polu (**98**).

Odczytywane w ten sposób wartości nie podlegają skalowaniu!

Posługujemy się komunikacją acykliczną – odczyt lub modyfikacja na żądanie (zmienna Start).

The screenshot displays the SINAMICS S120 parameter editor interface. The top part shows a ladder logic diagram with the following steps:

- Start (99):** EN (Enable) is TRUE, ENO (Enable Out) is FALSE. The output is "SINA_PARA_S_DB".Start.
- ReadWrite (100):** EN is TRUE, ENO is FALSE. The output is "SINA_PARA_S_DB".ReadWrite.
- Parameter (1121):** EN is TRUE, ENO is FALSE. The output is "SINA_PARA_S_DB".Parameter.
- ValueWrite1 (101):** EN is TRUE, ENO is FALSE. The output is "SINA_PARA_S_DB".ValueWrite1.
- Done (102):** EN is TRUE, ENO is FALSE. The output is "SINA_PARA_S_DB".Done.

The bottom part shows the "Expert list" table with the following columns: Param..., Dat..., Parameter text, Online value SERVO_02, Unit, and Modifiable to. The table lists various parameters, and the row for P1121 is highlighted in green, indicating it is the current parameter being modified.

Param...	Dat...	Parameter text	Online value SERVO_02	Unit	Modifiable to
361 p071	A	Save drive object parameters	[0] Inactive		Operation
362 p975[0]	A	Drive object identification, Company (Siemens = 42)	42		
363 p979[0]	A	PROFIdrive encoder format, Header	2266		
364 p1000[0]	C	Macro Connector inputs (CI) for speed setpoints	0		Ready to run
365 p1001	C	Maximum speed scaling	100.00	%	Operation
366 p1002[0]	D	Maximum speed	9000.000	rpm	Ready to run
367 p1003[0]	D	CO: Speed limit in positive direction of rotation	210000.000	rpm	Operation
368 p1004	D	CO: Speed limit positive effective	8250.000	rpm	Operation
369 p1006[0]	D	CO: Speed limit in negative direction of rotation	-210000.000	rpm	Operation
370 r1007	D	CO: Speed limit negative effective	-8250.000	rpm	Operation
371 p1121[0]	D	Ramp-function generator ramp-down time	4.500	s	Operation
372 p1135[0]	D	OFF3 ramp-down time	0.000		Operation
373 p1140[0]	C	BI: Enable ramp-function generator/inhibit ramp-function generator	SERVO_02 : r2090.4		Ready to run
374 p1141[0]	C	BI: Continue ramp-function generator/freeze ramp-function generator	SERVO_02 : r2090.5		Ready to run
375 p1142[0]	C	BI: Enable setpoint/inhibit setpoint	SERVO_02 : r2090.6		Ready to run
376 p1155[0]	C	CI: Speed controller speed setpoint 1	SERVO_02 : r2050[1]		Ready to run
377 p1160[0]	C	CI: Speed controller speed setpoint 2	0		Ready to run
378 r1169	CO	Speed controller speed setpoints 1 and 2	1200.073	rpm	
379 r1170	CO	Speed controller speed setpoint sum	1200.073	rpm	
380 p1199[0]	D	Speed setpoint configuration	3H		Operation
381 p1190	CI	DSC position deviation XERR	0		Ready to run

Jeżeli chcielibyśmy dokonać modyfikacji wartości parametru – przykładowo rampa hamowania P1121:

ReadWrite zmieniamy na **TRUE (99)**

Wprowadzamy nowy numer parametru 1121 index 0 (**100**)

W polu **ValueWrite1 (101)** wprowadzamy nową wartość – przykładowa **4,5s**

Operację modyfikacji możemy śledzić poprzez listę ekspercką programu **STARTER** – zmiana wartości następuje po ustawieniu zbrocza narastającego w poleceniu **START**.

Status operacji zakończonych prawidłowo reprezentuje wyjście bloku o nazwie **Done (102)**.