

Regulator PID w SIMATIC S7-1200

Projekt w środowisku TIA Portal V11

opr. Włodzimierz Solnik

3. Regulator PID w układach regulacji natężeniem przepływu powietrza, poziomu wody oraz jej natleniania

3.1. Wprowadzenie

Na przykładzie układów regulacji ciągłej natężeniem przepływu powietrza, poziomu wody oraz jej natleniania (rys. 1) przedstawiony zostanie sposób konfiguracji standardowego regulatora PID na bazie sterownika SIMATIC S7-1200 (CPU 1214C AC/DC/RLY Nr katalogowy 6ES7 214-1BE30-0XB0). Obiektami są dmuchawa powietrza z napędem trójfazowym oraz kolumna z wodą wyposażona w siłownik liniowy, regulacyjny z zaworem oraz dmuchawę powietrza. Oba obiekty wyposażono również w odpowiednie czujniki i przetworniki pomiarowe.

Wydajność dmuchawy jest sterowana przekształtnikiem częstotliwości Micromaster 440. Do pomiaru przepływu powietrza wykorzystuje się przetwornik różnicy ciśnień STD 924 z czujnikiem przepływu Diamond II Annubar GNT-10. Czujnik przepływu na wyjściu wytwarza ciśnienie różnicowe, które jest proporcjonalne do kwadratu przepływu objętości powietrza. Producent czujnika podaje firmowy wzór, wg którego oblicza się przepływ objętościowy:

$$Q_a = C' \sqrt{h_w}, \quad (1)$$

gdzie Q_a – wielkość przepływu w zakresie 0-2 (A) m³/min („(A)” oznacza przepływ rzeczywisty), h_w – różnica ciśnień w zakresie 0 – 42,1 mbar, C' – stała obliczeniowa równa 0.308552 (dla różnicy ciśnienia z zakresu 0 - 42.1 mbar). Stała obliczeniowa uwzględnia między innymi średnicę wewnętrzną rury (35.001 mm), gęstość powietrza, współczynnik konwersji jednostek, współczynnik przepływu dla annubara, współczynnik rozszerzalności gazu. Odpowiednie obliczenie może być wykonane przez program aplikacyjny w sterowniku. Zakłócenie stabilnego przepływu symulowane będzie zmianą skokową poboru na wyjściu (przytłacanie wyjścia).

Zbiornik (kolumna) wody wyposażony jest w przetwornik poziomu z czujnikiem ultradźwiękowym (PROSONIC FMU 860 i FDU 80 f-my Endress + Hauser, 0-1,9m – 0-20mA) oraz siłownik liniowy (ESL-10 f-my ZAP, 0-25mm – 4-20mA) z zaworem na doprowadzeniu wody zasilającej. Zakłócenie symulowane będzie ręcznym zaworem odpływowym (niestacjonarny pobór wody). Woda w zbiorniku jest natleniana dmuchawą powietrza (TLD 3 f-my Rietschle, 3,8m³/h) a do pomiaru poziomu zawartości tlenu w wodzie wykorzystany jest tlenomierz z sondą tlenową (LIQUISYS-0 f-my Endress + Hauser i COS 3S f-my Conducta, 0-15mgO₂/l – 4-20mA). Dmuchawa powietrza sterowana będzie z regulatora poprzez przekształtnik częstotliwości SINAMICS G120 f-my Siemens po sieci Profinet IO. Zakłóceniem (powód stosowania układu regulacji) jest naturalna (wolna, chłodna woda z kranu pobierana) redukcja tlenu w wodzie, do badań testowych można wykorzystać Na₂SO₂ – środek chemiczny powodujący redukcję tlenu zawartego w wodzie.

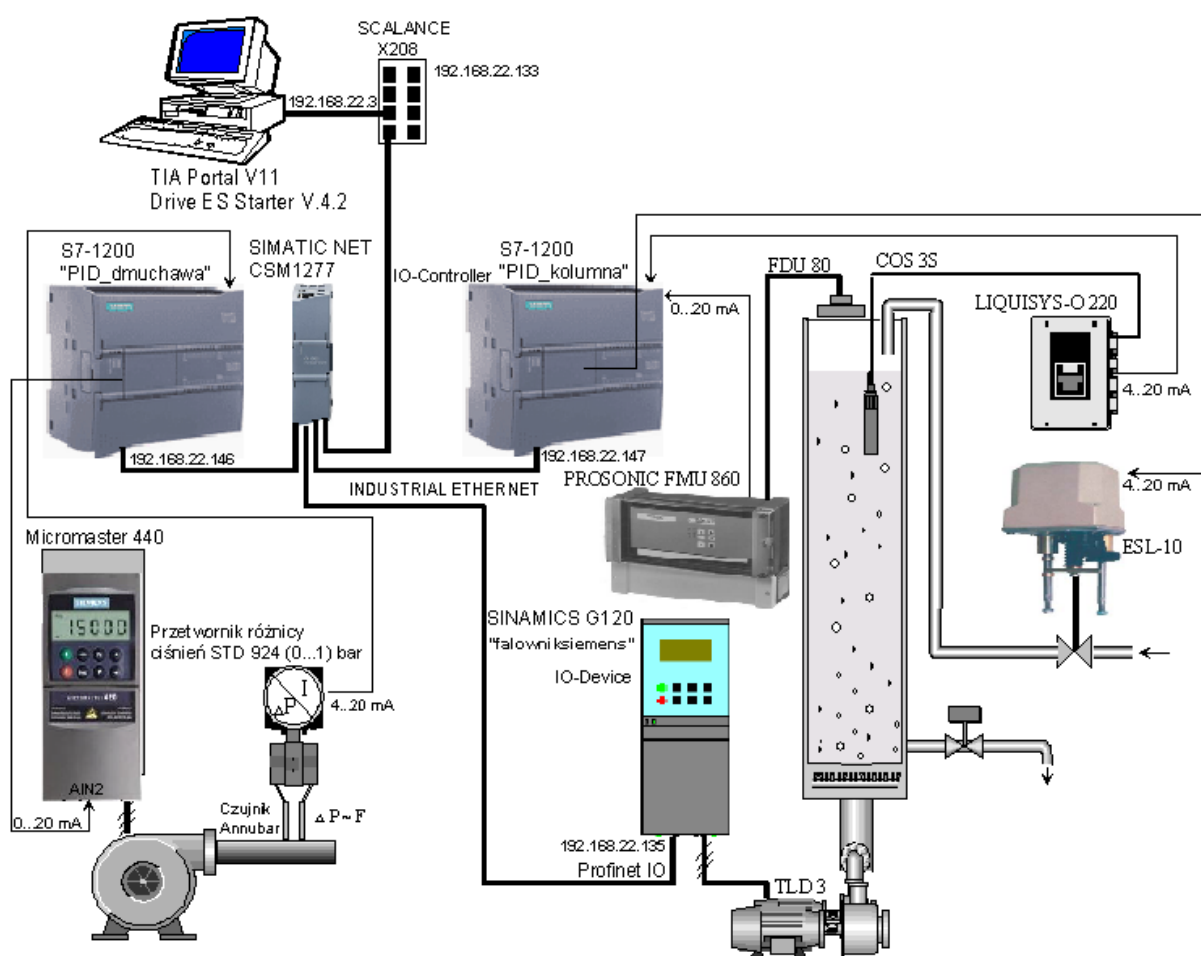
W testowym projekcie wykorzystano dwa sterowniki:

- „PID_dmuchawa” (adres IP 192.168.22.146), w którym skonfigurowano regulator PID do układu regulacji ciągłej natężeniem przepływem powietrza.

- „PID_kolumna” (adres IP 192.168.22.147), w którym skonfigurowano regulatory PID do układów regulacji ciągłej poziomu wody oraz jej natleniania. Przygotowano przekształtnik częstotliwości SINAMICS G120 do działania w układzie regulacji natleniania. Sterownik

„PID_kolumna” działa jako „IO - Controller” w sieci Profinet IO, gdzie przekształtnik częstotliwości jest urządzeniem „IO-Device”. Pojęcia „IO - Controller” i „IO - Device” wykorzystywane w sieci Profinet IO są odpowiednikami „master” i „slave” w sieci Profibus DP. Przedstawiona zostanie również wymiana danych pomiędzy dwoma sterownikami przez sieć Ethernet w komunikacji *Open user communication* z wykorzystaniem *T-blocks* (*TSEND_C* oraz *TRCV_C*) i protokołu *TCP* dla potrzeb układu regulacji natleniania.

Przeprowadzono proces automatycznego doboru nastaw (*Autotuning*) w dwóch trybach (*Pretuning* oraz *Fine tuning*) dla skonfigurowanych regulatorów.



Rys. 1. Układy regulacji ciągłej natężeniem przepływu powietrza, poziomu wody oraz jej natleniania

Do wykorzystania w serii sterowników S7-1200 proponowane są różne rodzaje jednostek centralnych różniących się funkcjonalnością czyli możliwościami wyposażenia sprzętowego, pojemnością pamięci użytkownika i zintegrowanymi, lokalnymi portami wejść/wyjść. Nie wszystkie jednostki są widoczne w katalogu sprzętu podczas tworzenia projektu ponieważ pojawiły się na rynku później niż wersja V11 SP2 programu TIA Portal (wykorzystywana w projekcie). Brakuje na przykład CPU 1215C i CPU 1217C, posiadających oprócz 2 wejść analogowych także 2 zintegrowane wyjścia analogowe. Jednostka CPU 1217C posiada dwa wbudowane porty komunikacyjne Ethernet (typu *switch*, jeden adres IP). Umożliwia to

tworzenie sieci o topologii magistrali czy też pierścienia. Każdy z rodzajów jednostek centralnych (oprócz CPU 1217C) posiada jeszcze odmiany w zależności od rodzaju napięcia zasilającego (AC lub DC) czy też od rodzaju wyjścia zintegrowanego – tranzystorowe (DC) lub przekaźnikowe (Rly). Oprócz standardowego wykonania są również wersje jednostek fail-safe – CPU 1214FC i CPU 1215FC (obsługują aplikacje bezpieczne jak i standardowe, TIA Portal V13).

Oprócz różnic sprzętowych (hardware, „elektronika”) – FS (Functional Stand) jednostki centralne różnią się oprogramowaniem systemowym (software) – FW (FirmWare), od którego zależą możliwości funkcjonalne jednostki. Wersje FW mogą być aktualizowane wyłącznie w ramach grupy. Każda nowa wersja TIA Portal powstaje w związku z wprowadzeniem nowych jednostek CPU o coraz większej funkcjonalności. Istnieje więc powiązanie pomiędzy wersjami TIA Portal oraz FW sterowników. Jednostki z wersjami FW V1.x , V2.x , V3.X nie są już produkowane. FW w wersjach V4.X, obsługiwanych od TIA Portal V13. Jednostka w wersji V4.0 udostępnia wbudowaną funkcjonalność „failsafe”, ale tylko w centralnej kasecie, natomiast V4.2 wyposaża jednostkę w obsługę PROFIsafe, co oznacza, że jest w automatyce maszynowej sterownikiem bezpieczeństwa w pełni.

TIA PORTAL V12, V13, V14 działają pod systemem Windows 7, ze wsparciem przewidzianym do 01/2020 r., natomiast TIA Portal V13 SP2, V14 SP1 i V15 pod Windows 10, ze wsparciem przewidzianym do 10/2025. Jest możliwość aktualizacji oprogramowania z V13 SP2 do V14 SP1 i dalej do V15. Najnowsza wersja pakietu TIA Portal v16 zawiera m.innymi :

- interface OPC UA dla sterowników S7-1200,
- w katalogu sprzętowym nową jednostkę CPU 1513 pro,
- funkcję „ShowBlockInTIAPortal” (w WinCC) umożliwiającą wyświetlanie bloków programu PLC na ekranach HMI,
- wzbogaconą funkcję TMAIL_C umożliwiającą wysyłanie z jednostek CPU 1200/1500 wiadomości email z karty pamięci,
- w ramach tej wersji oprogramowania TIA Portal możliwa współpraca z WinCC Unified v16, udostępnienie nowoczesnych technologii webowych (HTML5, SVG oraz JavaScript) pozwalających na dostęp do *Runtime* WinCC Unified przez przeglądarkę internetową.

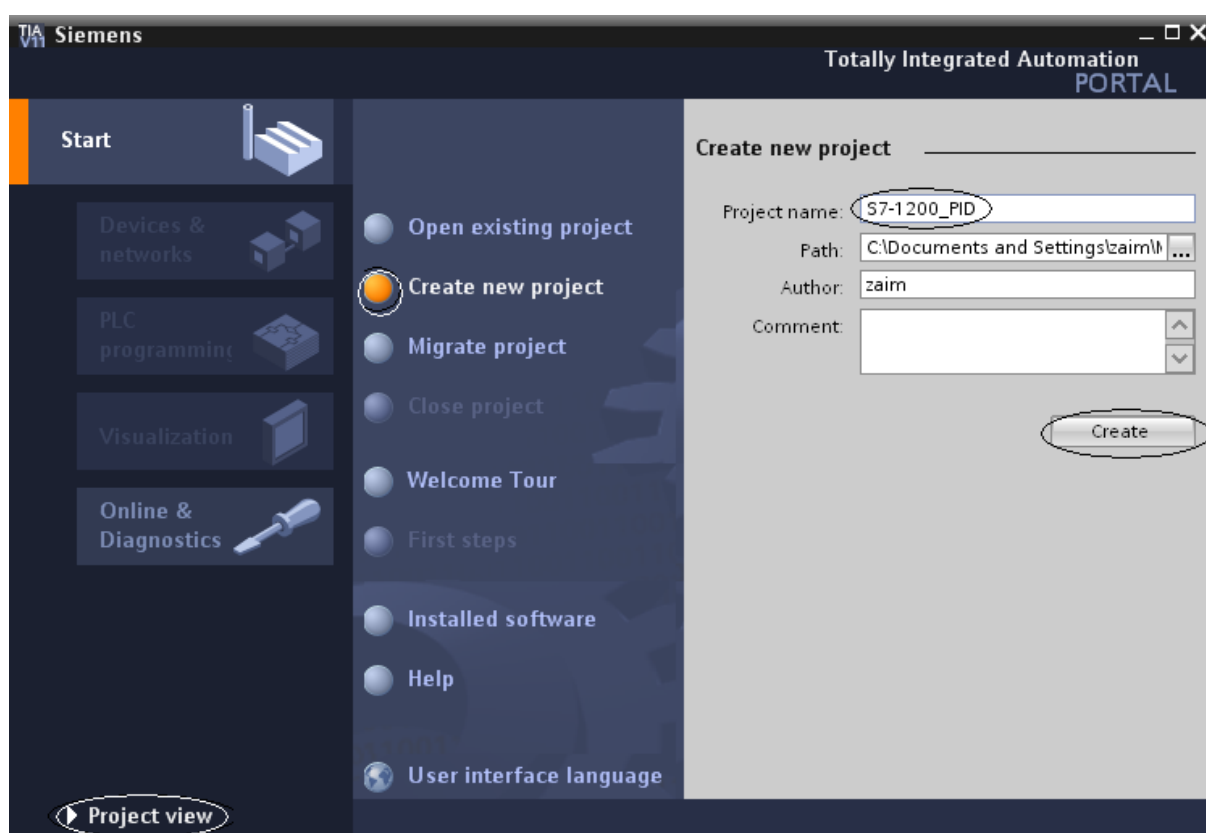
W projekcie wykorzystano jednostki CPU 1214C AC/DC/Rly z FW V2.2:

- 14 wejść i 10 wyjść binarnych,
- 2 wejścia analogowe (0-10VDC) ,
- pamięć robocza – 100kb, ładowania – 4MB i nieulotna – 10kb,
- rozmiar obszaru odwzorowań zmiennych procesu dla I i Q po 1024 bity,
- pamięć bitów obszaru znaczników M – 8192.

Chociaż wykorzystany w projekcie sprzęt i wersja oprogramowania nie jest najnowszy, to proces wykorzystywania instrukcji bibliotecznych PID jest podobny dla różnych wersji FW sterowników oraz TIA Portal.

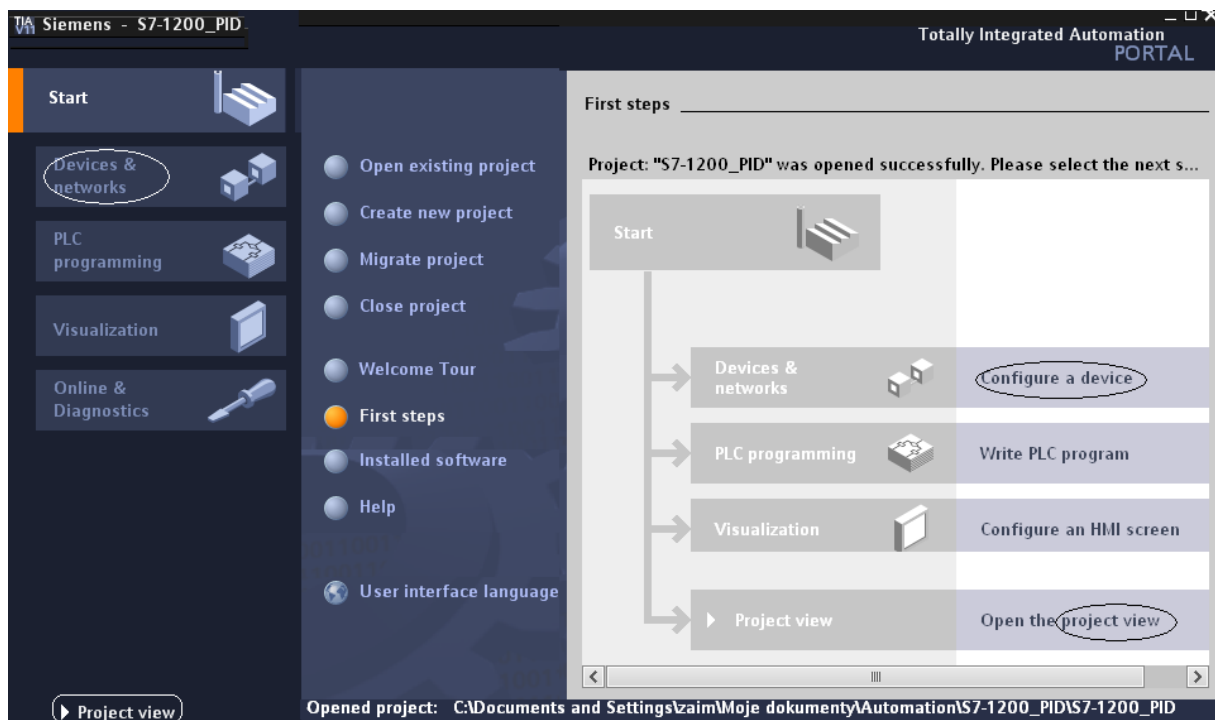
3.2. Otwieranie nowego projektu i konfiguracja sprzętowa sterownika „PID_dmuchawa”

Do opracowania projektu wykorzystano platformę programową TIA Portal V11 SP2, której ekran startowy (*Portal view*) pokazano na rys.2. W celu utworzenia nowego projektu w oknie *Portal view* wybrać *Create new project*. Większość pracy projektowej odbywa się jednak po przejściu do okna *Project view*. Można tu wybrać ekran startowy spośród opcji pojawiających się po wybraniu Options/Settings/General/Start view (*Portal view* lub *Project view*). Zmiana ekranu jest możliwa poprzez odpowiedni przycisk w jego lewym dolnym rogu. Nadano projektowi nazwę „S7-1200_PID” i potwierdzono przyciskiem *Create* jego tworzenie. Projekt będzie zawierał konfigurację sprzętową i oprogramowanie obu sterowników.



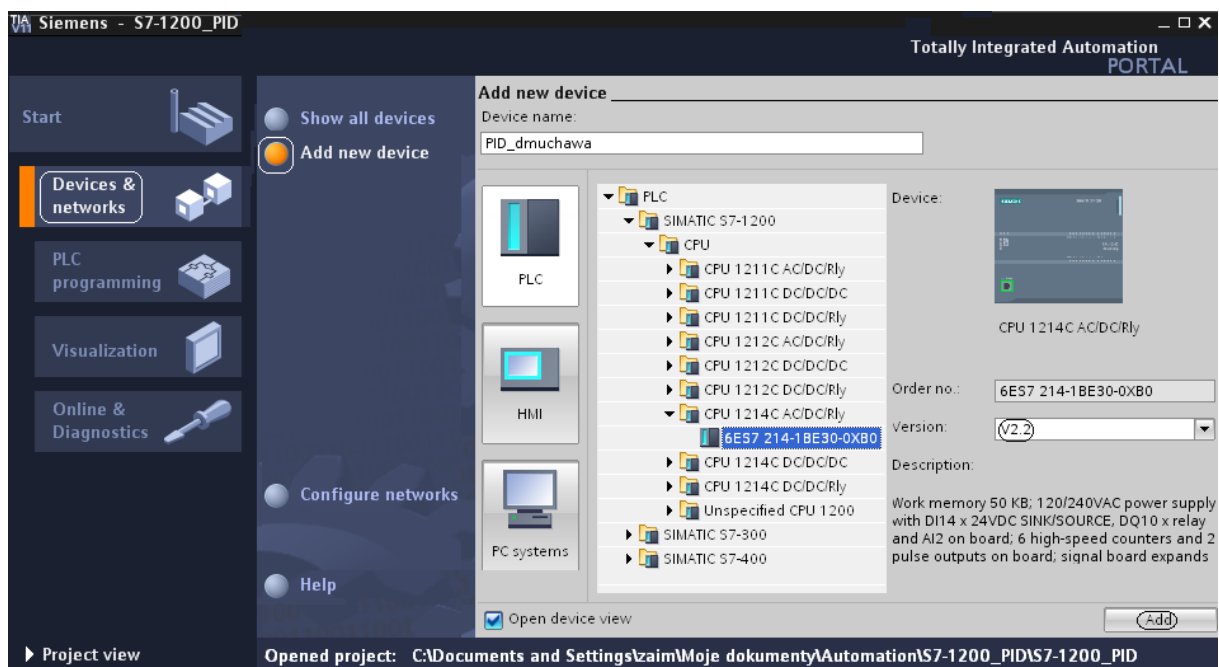
Rys. 2. Rozpoczęcie nowego projektu na ekranie *Portal view*

Następnie przeprowadza się konfigurację sprzętową sterownika , wybierając najpierw zakładkę *Devices & networks /Configure a device* . (rys. 3).



Rys. 3. Wybór etapu konfiguracji sprzętowej sterownika w oknie *Portal view* lub przejście do okna *Project view*.

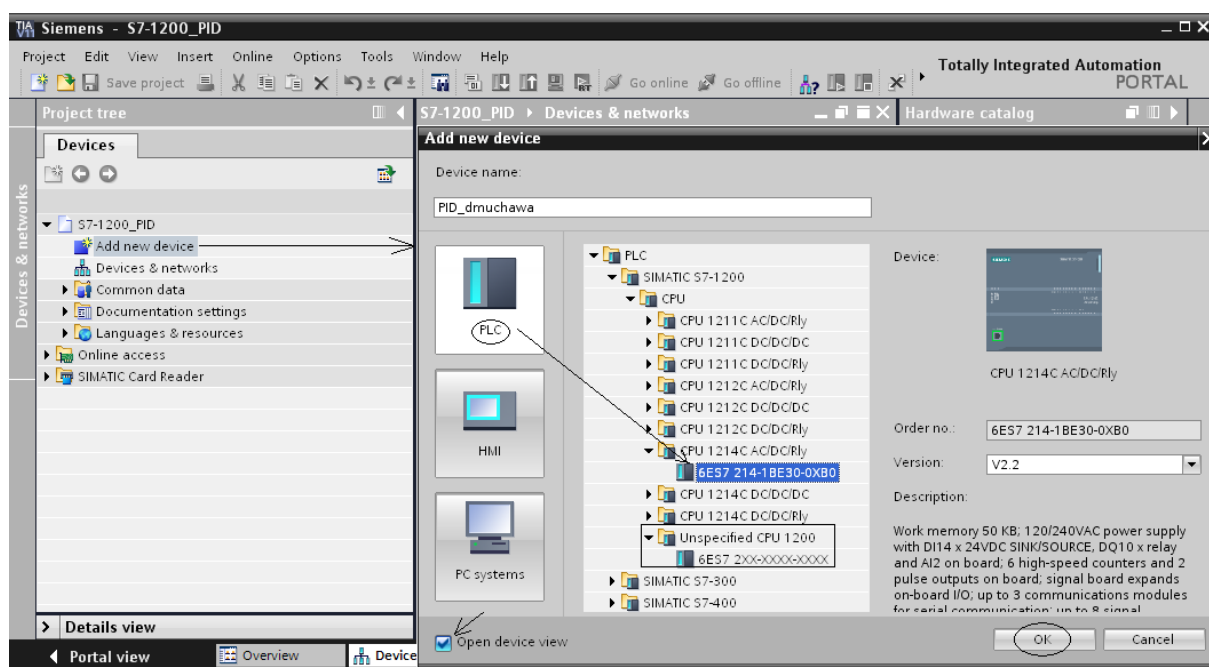
Konfiguracja sprzętowa sterownika przeprowadzona będzie w oknie *Portal view* w sposób standardowy, „katalogowy” (rys.4). Można tu też nadać sterownikowi nazwę.



Rys. 4 Dodawanie do projektu nowego urządzenia na ekranie *Portal view*

Podobnie konfigurację sprzętową sterownika w sposób standardowy, „katalogowy”, przeprowadzić można w oknie *Project view* (rys.5). W drzewie projektu należy kliknąć

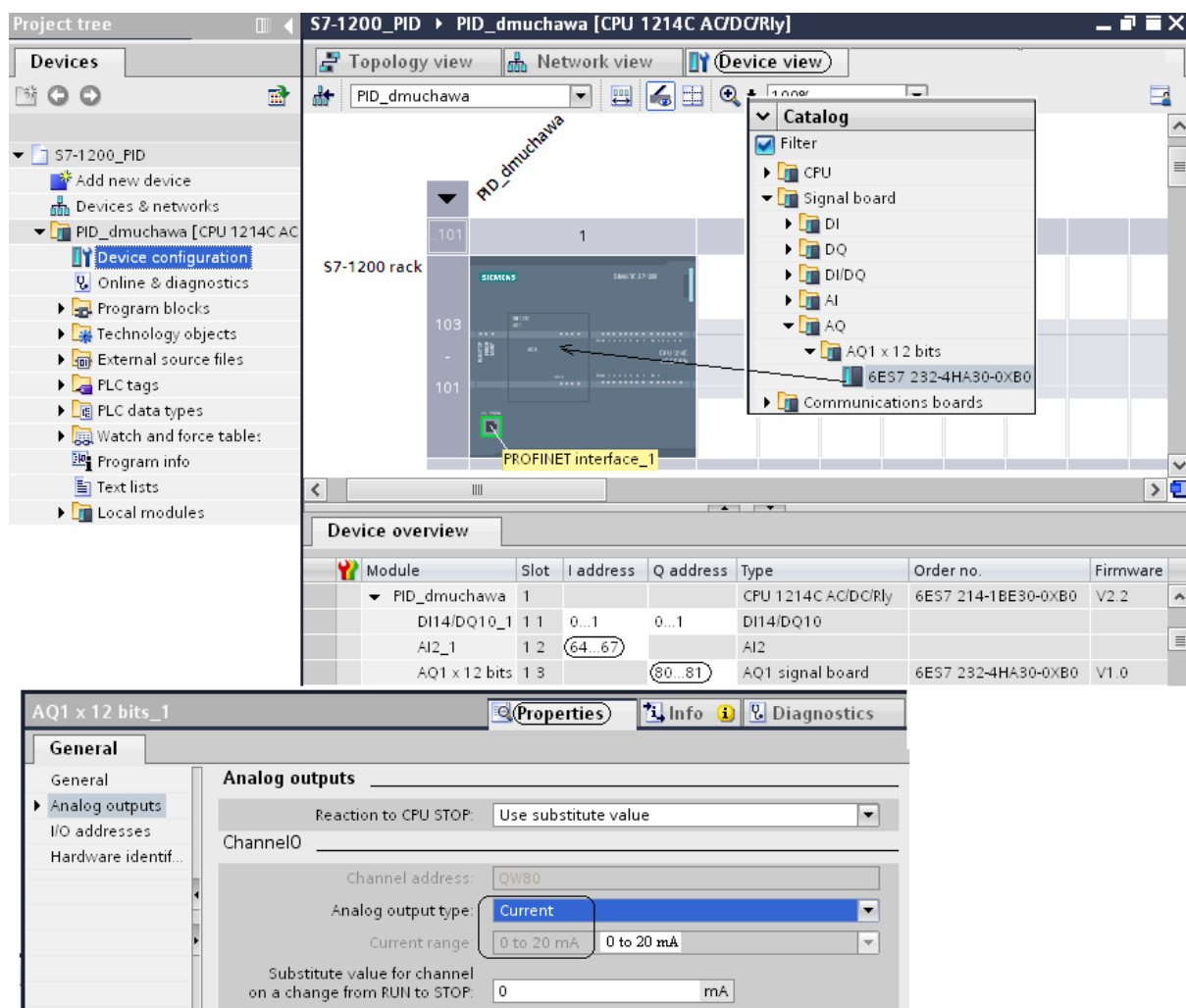
połączenie *Add new device* i w otwierającym się oknie wybrać jednostkę centralną i wersję wykonania (przy aktywnej ikonie „PLC”).



Rys. 5. Wybór w oknie *Project view* jednostki centralnej z wersją wykonania.

Dodawanie nowego urządzenia można przeprowadzić również z wykorzystaniem przeglądu sieci w celu detekcji dostępnego fizycznie odpowiedniego sterownika. W tym celu należy zaznaczyć jednostkę *Unspecified CPU 1200*. Ten sposób zostanie (metoda „detekcji”) wykorzystany podczas konfigurowania sprzętowego sterownika „PID_kolumna” (rys.55).

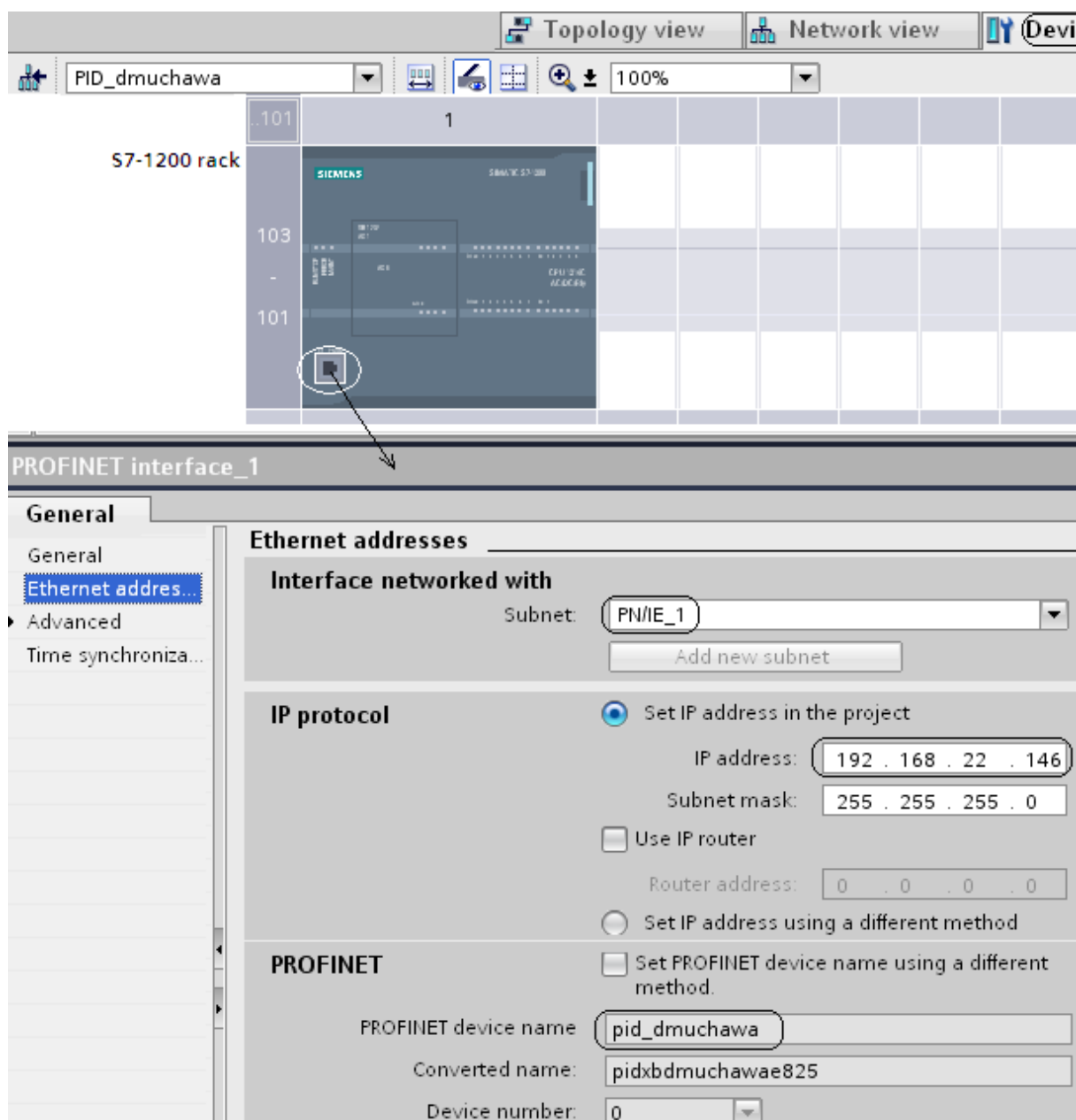
Po dodaniu do projektu nowego urządzenia (CPU sterownika), w automatycznie otwartym oknie *Device view* (w zakładce *Device configuration* okna *Project tree*) należy uzupełnić konfigurację sterownika o kartę sygnałową *AQ1x12 bit* (rys. 6). Ustawić zakres 0 – 20mA. Na rysunku przedstawiono również adresy wejść/wyjść w obszarze odwzorowań I/O skonfigurowanego sterownika S7-1200 („PID_dmuchawa”).



Rys.6. Uzupełnienie konfiguracji sprzętowej, adresy wejść/wyjść w obszarze odwzorowań I/O sterownika „PID_dmuchawa” i ustawienie zakresu karty sygnałowej *AQ1x12 bit* w oknie *Device view*.

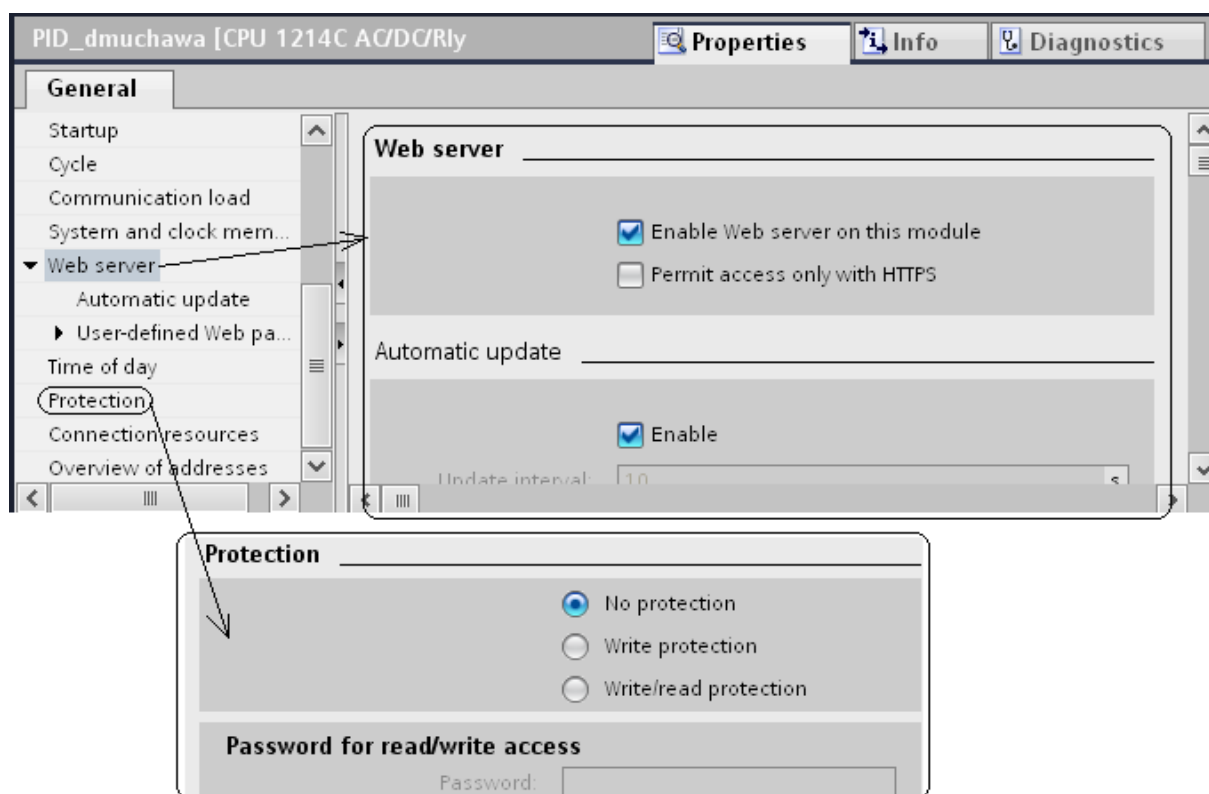
Klikając w symbol złącza portu ethernetowego (nazywane „RJ45”, „IE Profinet”) wywołuje się okno własności *PROFINET interface_1* (rys. 7), gdzie można nadać adres IP, nazwę *PROFINET device name* sterownikowi „PID_dmuchawa” jako *IO-Device* (pozostanie domyślna) oraz przypisać go do wybranej podsiatki PN/IE_1.

Nazwa «RJ45» (RJ - Registered Jack) błędnie wykorzystywana jest do złącza *8P8C* (8 Position 8 Contact), stosowanego w sieciach ethernetowych. W rzeczywistości RJ45 (RJ45S) jest całkowicie niekompatybilny fizycznie z złączem *8P8C* (jest standardem złącza *8P4C* i wyposażona w dodatkową wypustkę, klucz uniemożliwiający włożenie jej do typowego gniazda *8P8C*). Zewnętrzne podobieństwo, a także brak szerokiego zastosowania „prawdziwego” RJ45 chyba spowodowało powszechne wykorzystanie tego żargonu.



Rys. 7. Nadawanie adresu IP oraz nazwy *PROFINET device name* sterownikowi „PID_dmuchawa”

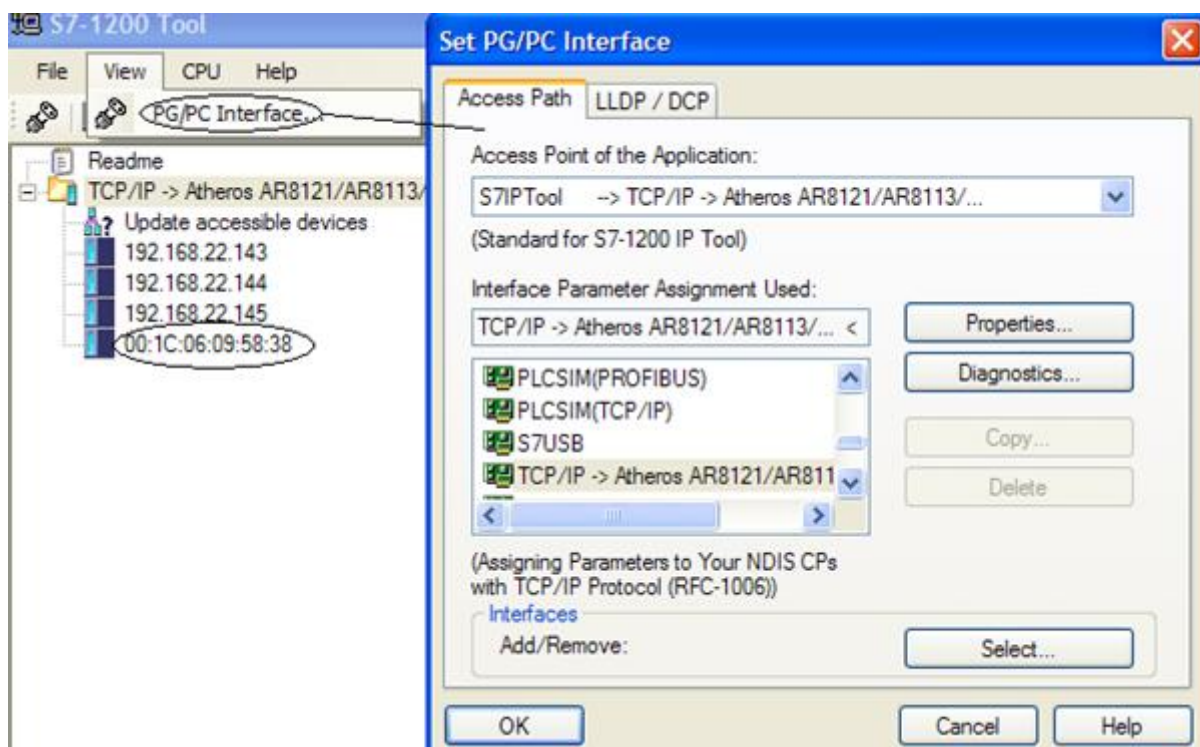
Ustawić także (uaktywnić) w oknie *Properties* (rys. 8) dla sterownika „PID_dmuchawa” możliwość korzystania z wbudowanego web serwera poprzez przeglądarkę internetową i ewentualnie ustawić hasło dostępu dla opcji *pisz* czy też *pisz/czytaj*.



Rys. 8. Okno uaktywniania wbudowanej opcji *Web server* oraz *Protection*

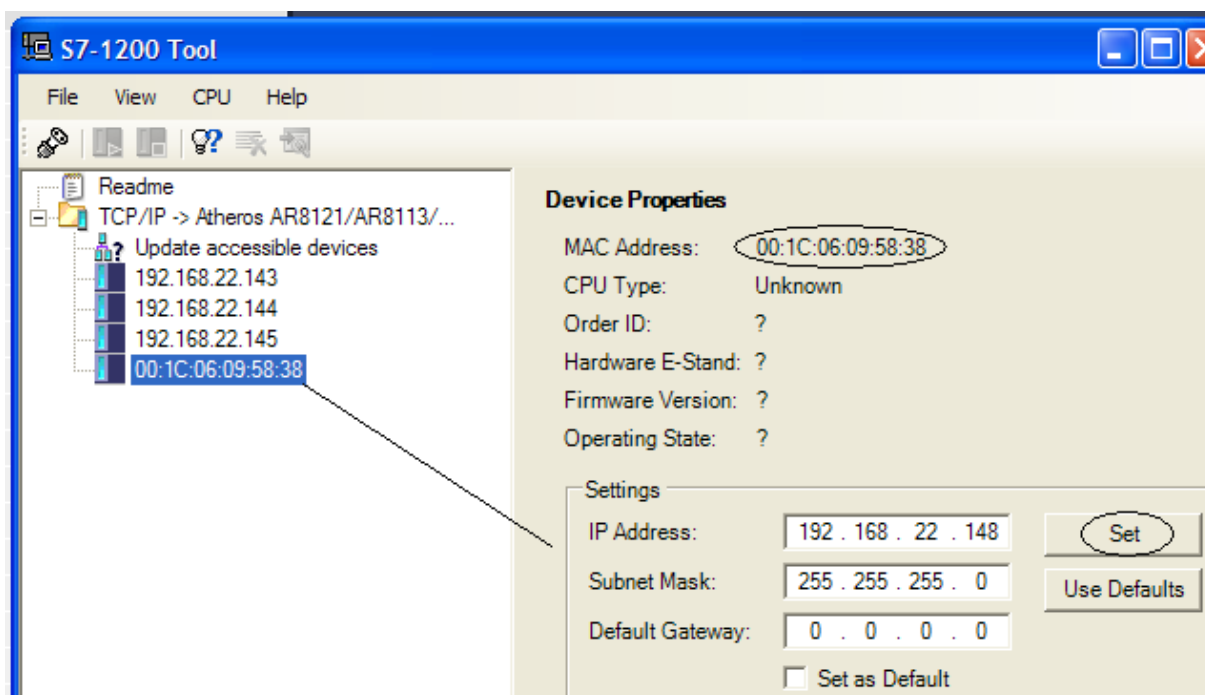
Adres IP 192.168.22.146 sterownika „PID_dmuchawa” w projekcie jest aktualnym adresem istniejącego w sieci lokalnej sterownika S7-1200 przeznaczonego do testowania tego projektu.

Ustawianie adresu posiadanego sterownika można przeprowadzić wykorzystując program użytkowy *S7-1200 Tool* (rys. 9). W oknie *Set PG/PC Interface* wybiera się rodzaj połączenia komputera (programator) ze sterownikiem -TCP/IP i typ zainstalowanej karty komunikacyjnej.



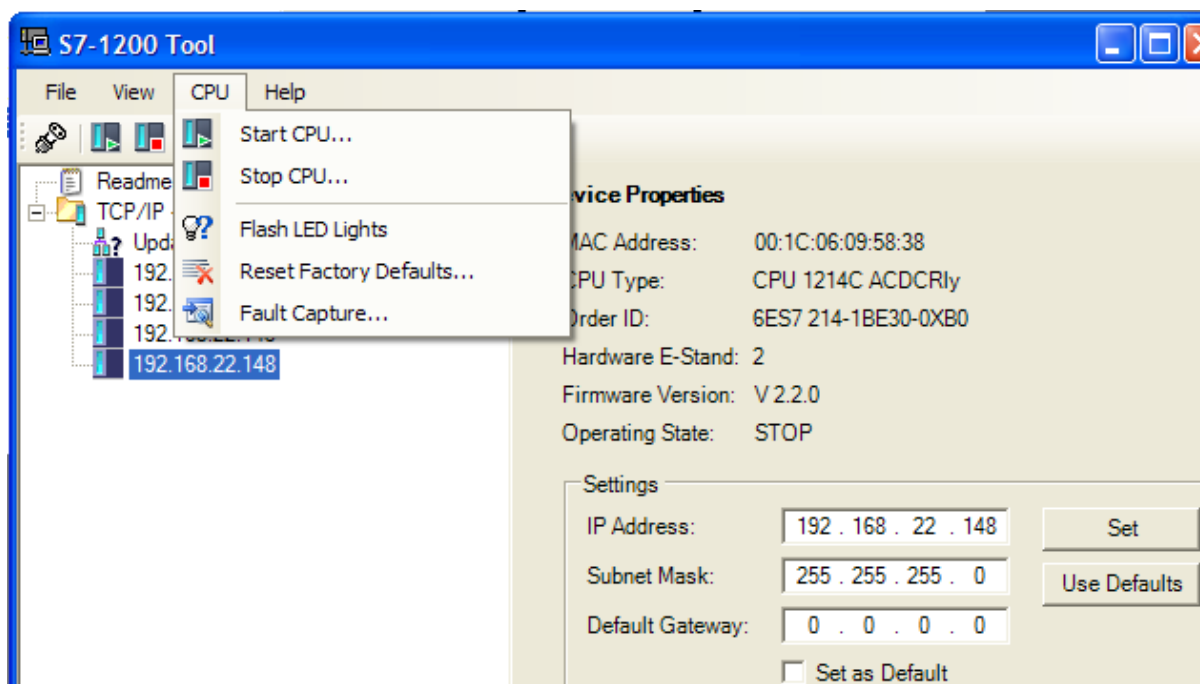
Rys. 9. Wybór rodzaju połączenia między sterownikiem a komputerem w *S7-1200 Tool*.

Sterownikowi posiadającemu tylko adres MAC można przypisać zaplanowany adres IP (na przykład IP 192.168.22.148) w podsieci miejscowej (rys. 10).



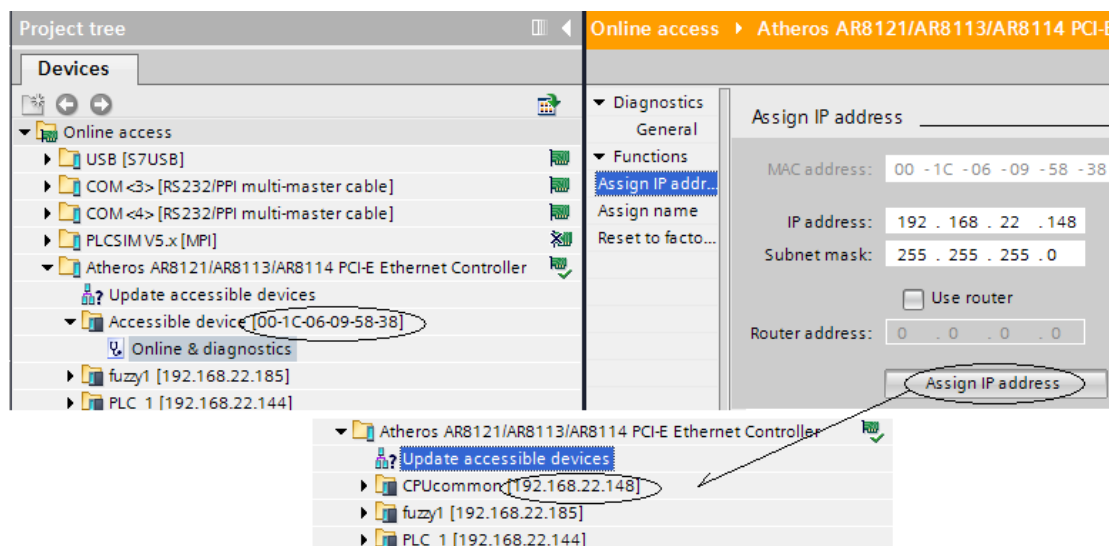
Rys. 10. Ustawianie adresu IP w sterowniku o wybranym adresie MAC.

Program *S7-1200 Tool* pozwala także skorzystać z funkcji dodatkowych: start/stop jednostki centralnej, wskazanie wybranego węzła poprzez migotanie diody LED, powrót do ustawień fabrycznych, rejestracja błędów (rys. 11).



Rys. 11. Dodatkowe funkcje programu *S7-1200 Tool*

Innym sposobem przypisywania adresu IP jest wybranie w drzewie projektu (program TIA Portal) zakładki *Online acces/nazwa ethernetowej karty komunikacyjnej* w komputerze, a następnie uaktywnić *Update accessible devices*. Wyświetlony zostanie wykaz wszystkich węzłów w podsięci wraz z ich adresami IP lub MAC. Na rys. 12 pokazano przykład przypisania węzłowi z zaznaczonym adresem MAC określonego adresu IP oraz maski podsięci. Po wydaniu polecenia *Assign IP address* i odświeżeniu listy węzłów podsięci (*Update accessible device*) na nowej liście zaobserwować można dany węzeł z przypisanym mu adresem IP.

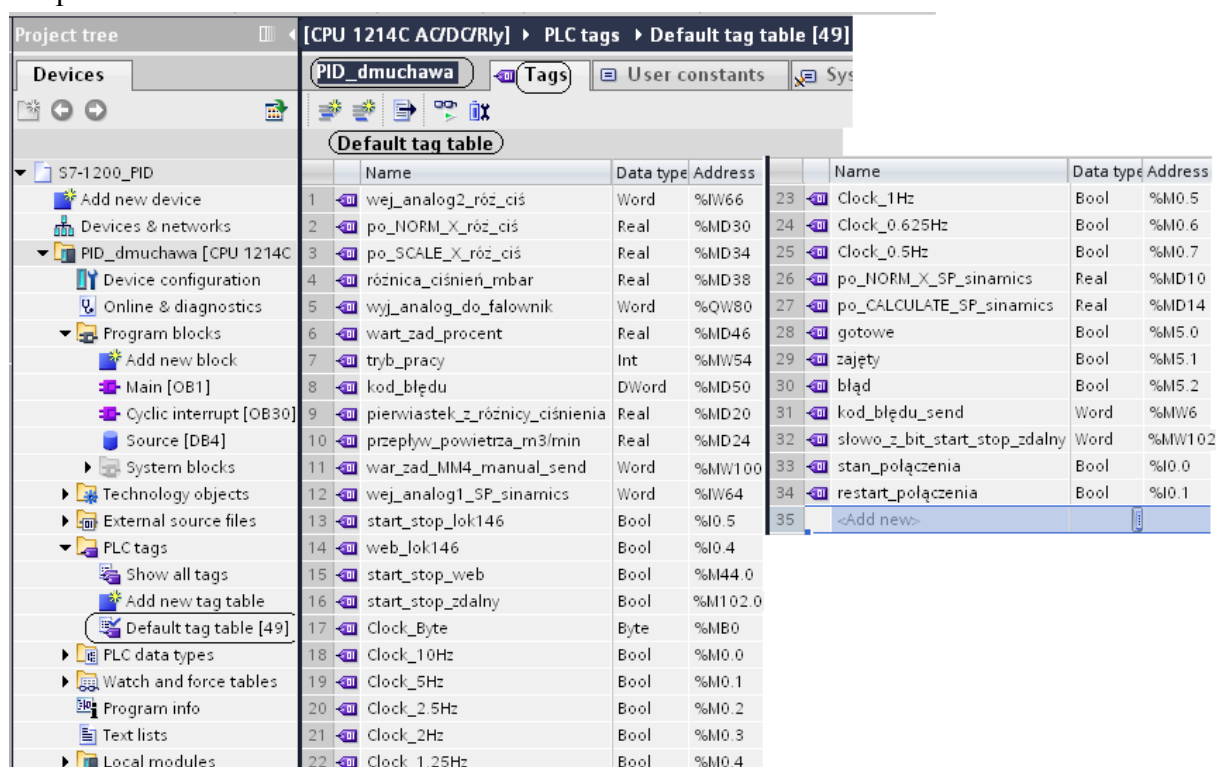


Rys. 12. Przypisanie adresu IP w zakładce *Online access* w narzędziu TIA Portal

Do zmiany adresu IP węzła oraz parametrów sieci Ethernet można też wykorzystać niezależne narzędzie *SIMATIC NET Primary Setup Tool*. Obecnie zamiast *S7-1200 Tool i Primary Setup Tool* udostępniane jest narzędzie programowe *SIMATIC Automation Tool*.

3.3 Oprogramowanie sterownika „PID_dmuchawa”

Przed opracowywaniem oprogramowania aplikacyjnego dla sterownika „PID_dmuchawa” można w oknie przeglądarki projektu (*Project tree*) rozwinąć folder *PLC tags*, kliknąć na zakładkę *Default tag table* i wpisać parametry zmiennych wykorzystywanych w ramach projektu : nazwę, typ zmiennej oraz adres. Rys.13 przedstawia zawartość tablicy zmiennych już na końcowym etapie projektu, która podczas projektowania może być aktualizowana i uzupełniana .



Name	Data type	Address	Name	Data type	Address
1 wej_analog2_róż_cis	Word	%IW66	23 Clock_1Hz	Bool	%M0.5
2 po_NORM_X_róż_cis	Real	%MD30	24 Clock_0.625Hz	Bool	%M0.6
3 po_SCALE_X_róż_cis	Real	%MD34	25 Clock_0.5Hz	Bool	%M0.7
4 różnica_cisnień_mbar	Real	%MD38	26 po_NORM_X_SP_sinamics	Real	%MD10
5 wyj_analog_do_falownik	Word	%QW80	27 po_CALCULATE_SP_sinamics	Real	%MD14
6 wart_zad_procent	Real	%MD46	28 gotowe	Bool	%M5.0
7 tryb_pracy	Int	%MW54	29 zajęty	Bool	%M5.1
8 kod_błędu	DWord	%MD50	30 błąd	Bool	%M5.2
9 pierwiastek_z_różnicy_cisnienia	Real	%MD20	31 kod_błędu_send	Word	%MW6
10 przepływ_powietrza_m3/min	Real	%MD24	32 słowo_z_bit_start_stop_zdalny	Word	%MW102
11 war_zad_MM4_manual_send	Word	%MW100	33 stan_połączenia	Bool	%I0.0
12 wej_analog1_SP_sinamics	Word	%IW64	34 restart_połączenia	Bool	%I0.1
13 start_stop_lok146	Bool	%I0.5	35 <Add new>		
14 web_lok146	Bool	%I0.4			
15 start_stop_web	Bool	%M44.0			
16 start_stop_zdalny	Bool	%M102.0			
17 Clock_Byte	Byte	%MB0			
18 Clock_10Hz	Bool	%M0.0			
19 Clock_5Hz	Bool	%M0.1			
20 Clock_2.5Hz	Bool	%M0.2			
21 Clock_2Hz	Bool	%M0.3			
22 Clock_1.25Hz	Bool	%M0.4			

Rys. 13. Tablica *Default tag table* (PID_dmuchawa / PLC tags)

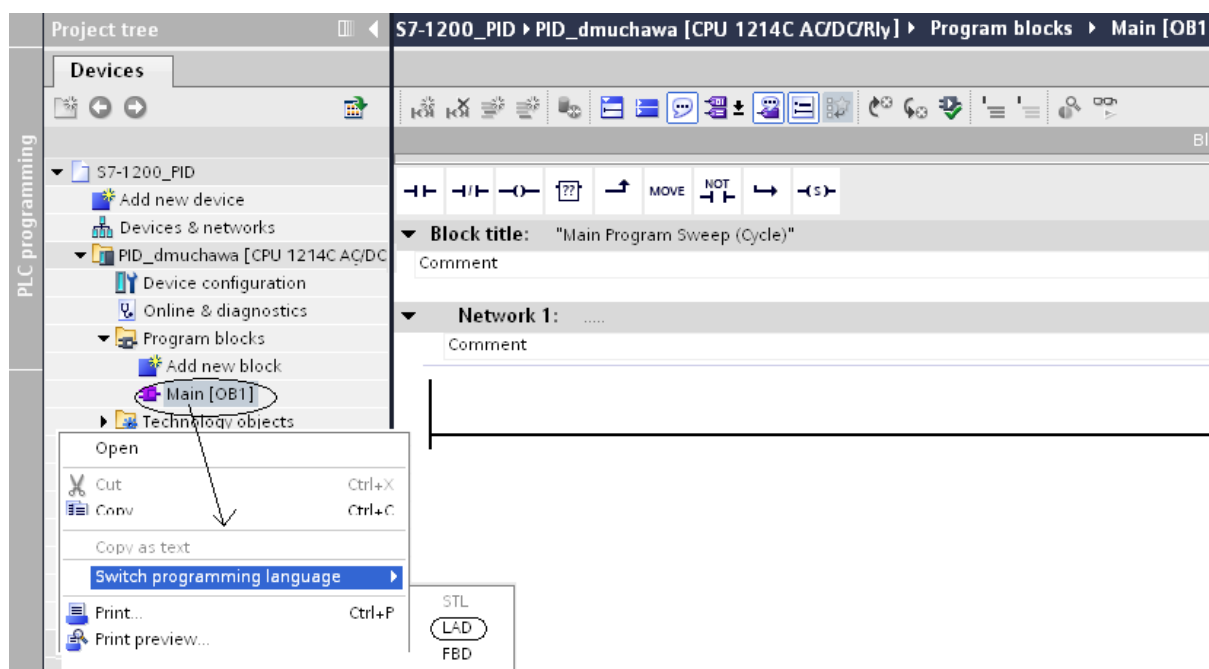
3.3.1 Realizacja softwarowego regulatora przepływu powietrza PID o wyjściu ciągłym (*reg_przepływu*)

Celem układu regulacji jest stabilizacja natężenia przepływu powietrza przez rurę niezależnie od działającego zakłócenia (regulacja natężenia przepływu powietrza pośrednio poprzez regulację innej, dostępnej bezpośrednio zmiennej procesowej – różnicy ciśnień na rurze wylotowej). Zakłócenia przepływu powietrza symulowane będą w układzie regulacji poprzez ręczne przytkanie wylotu lub wlotu dmuchawy. Z przetwornika pomiarowego różnicy ciśnień przychodzi sygnał 4-20mA (zmiana wielkości pomiarowej w granicach 0 – 30mbar), który jest podłączony pod drugie wejście analogowe AI1 (pierwsze oznaczone AI0) sterownika o adresie %IW66 (na wejściu podpięty jest opornik pomiarowy 500 om z którego pobierany jest sygnał , spadek napięcia, od 2 do 10 V). Regulator PID skonfigurowany w sterowniku S7-1200 będzie korzystał z tego sygnału jako wartości aktualnej zmiennej procesowej PV.

Natomiast wejście przekształtnika częstotliwości jest podłączone do wyjścia analogowego AQ0 sterownika o adresie %QW80 (zmienna sterująca CV – wyjście regulatora) i sterowany sygnałem i do jego wejścia. Zmiana sygnału (0-20mA) na wejściu przekształtnika powoduje zmianę częstotliwości zasilania trójfazowego silnika dmuchawy a tym samym jego prędkości obrotowej. Adresy wejść/wyjść binarnych i analogowych w obszarze odwzorowań I/O sterownika jakie zostały automatycznie przydzielone podczas konfiguracji sprzętowej sterownika można odczytać w oknie *Device overview* (rys.6).

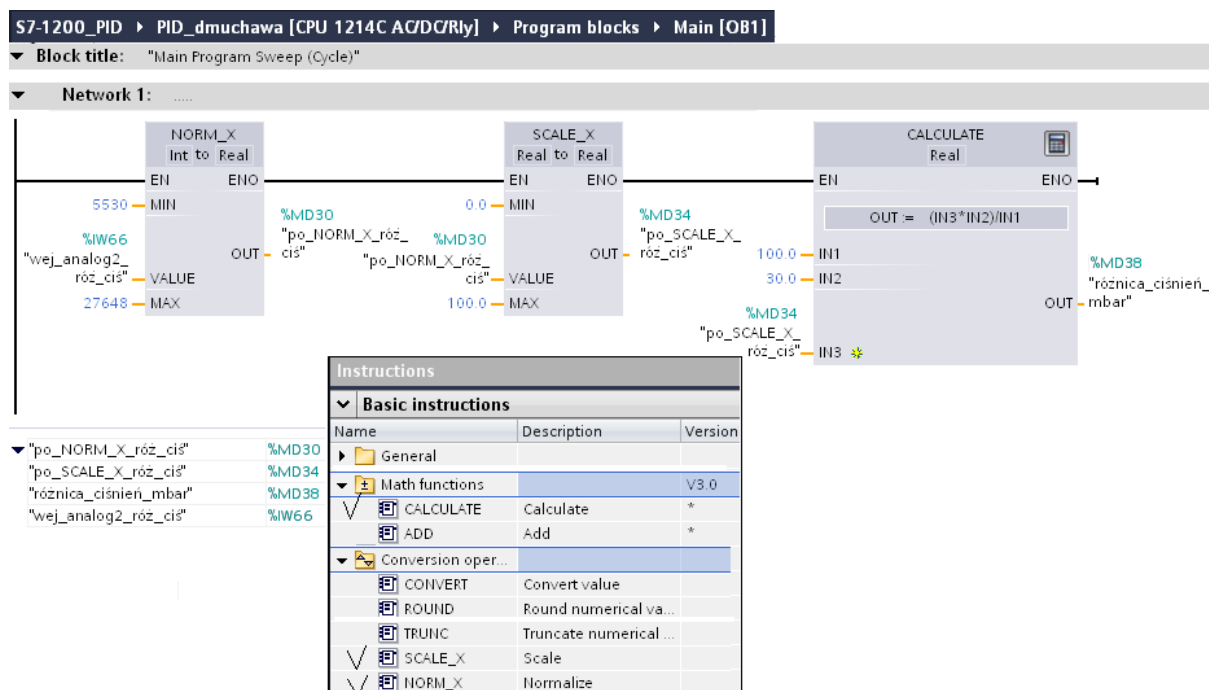
Tworzenie programu aplikacyjnego zaczyna się od wejścia w zakładkę *Program blocks* okna *Project tree* i dwukrotnym kliknięciu w *Main [OB1]*. Otworzy się okno (rys. 14) do pisania programu w wybranym języku LD (LAD) – edytor szczebli programowych tzw. *Networks*.

Język programowania wybrać można klikając ppm element *Main [OB1]* w drzewie projektu w zakładce *Switch programming language* lub w *własnościach* *Main [OB1]* (ppm i zakładka *Properties*).



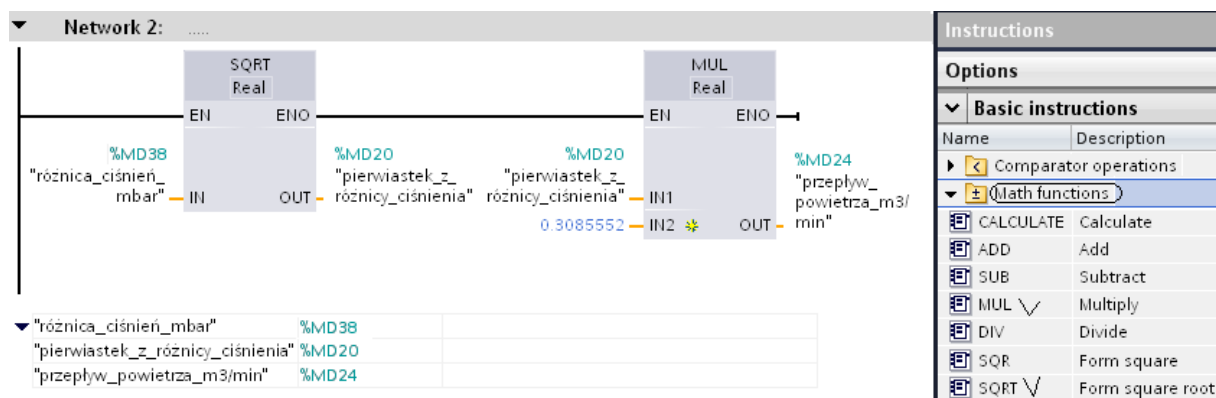
Rys. 14. Zakładka *Main [OB1]* i wybór języka programowania

Pierwszym krokiem (Network 1) będzie odczyt i przeskalowanie wartości pomiarowej, odczytanej z przetwornika pomiarowego na drugim wejściu analogowym (AI1) sterownika (%IW66). W tym celu należy z biblioteki *Basic instruction – Conversion operations* wybrać bloki *NORM_X* i *SCALE_X*. Instrukcja *NORM_X* służy do zamiany przedziału wartości wejścia analogowego z 5530 – 27648 (4 – 20 ma) na przedział od 0.0 do 1.0. Kolejna instrukcja *SCALE_X* służy do zamiany przedziału z 0.0 – 1.0 na przedział od 0 do 100 [%]. W celu przeliczenia procentowej wartości na wartość rzeczywistą należy dodać instrukcję *CALCULATE*, która będzie przeliczała wartość procentową od 0 do 100% na wartość rzeczywistą od 0 do 30 mbar (rys.15).



Rys. 15. Skalowanie wartości różnicy ciśnień oraz jej przeliczanie na wartość w jednostkach fizycznych [mbar]

W sterowniku na podstawie wzoru (1) będzie obliczona (posiadając pomiar różnicy ciśnień) rzeczywista wartość objętościowego przepływu powietrza (Network 2, rys. 16).



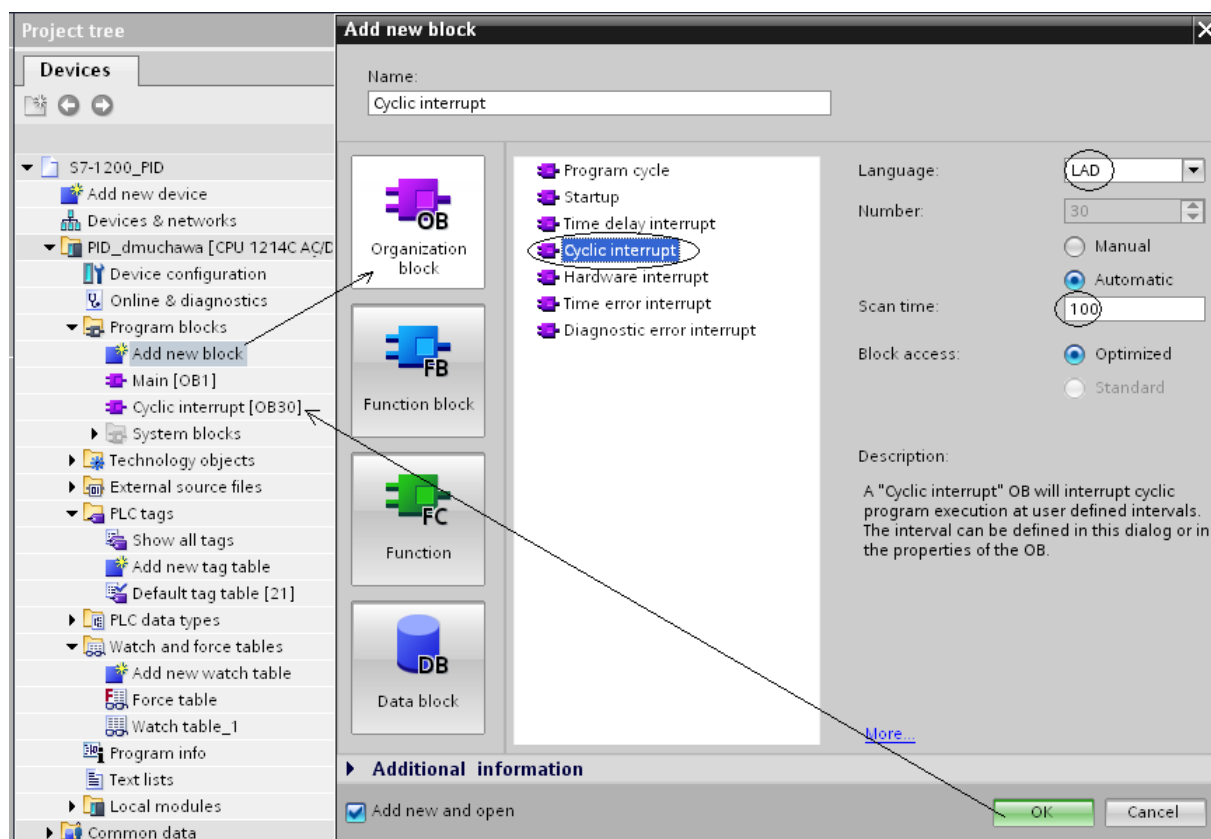
Rys. 16. Obliczenie rzeczywistej wartości objętościowego przepływu powietrza

Szczeble (Networks) 1-2 stanowią część „pomocniczą oprogramowania” do wykorzystania np. przy wizualizacji procesu i niekonieczną do realizacji regulatora programowego.

Następnym krokiem w realizacji projektu jest dodanie i skonfigurowanie regulatora PID o wyjściu ciągłym.

Do utworzenia softwarowego regulatora PID o wyjściu ciągłym w sterowniku S7-1200 potrzebny jest blok funkcjonalny (FB) PID_Compact (instrukcja z algorytmem sterowania) oraz odpowiedni dla tej instrukcji blok danych (DB) PID_Compact_Nr (blok danych *single instance*) jako „obiekt technologiczny”. Konfiguracja regulatora zapisana zostanie w tym „obiekcie technologicznym”, a przechowywany on będzie w specjalnej zakładce *Technology objects* okna *Project tree*.

Instrukcja „PID” wymaga wywoływania cyklicznego, aby tego dokonać należy dodać nowy blok OB przerwania cyklicznego (OB30). W oknie konfiguracji projektu *Project tree* (rys. 17) należy kliknąć dwukrotnie zakładkę *Add new block*. Następnie wybrać *Organization block (OB)* i zaznaczyć *Cyclic interrupt*. Nadać nazwę, wybrać język programowania *LAD* (LD), wprowadzić wartość częstotliwości wywoływania instrukcji (np. *co100 ms*), zaznaczyć opcję *Add new and open* i potwierdzić OK.



Rys. 17. Wybór bloku organizacyjnego OB30 przerwania cyklicznych

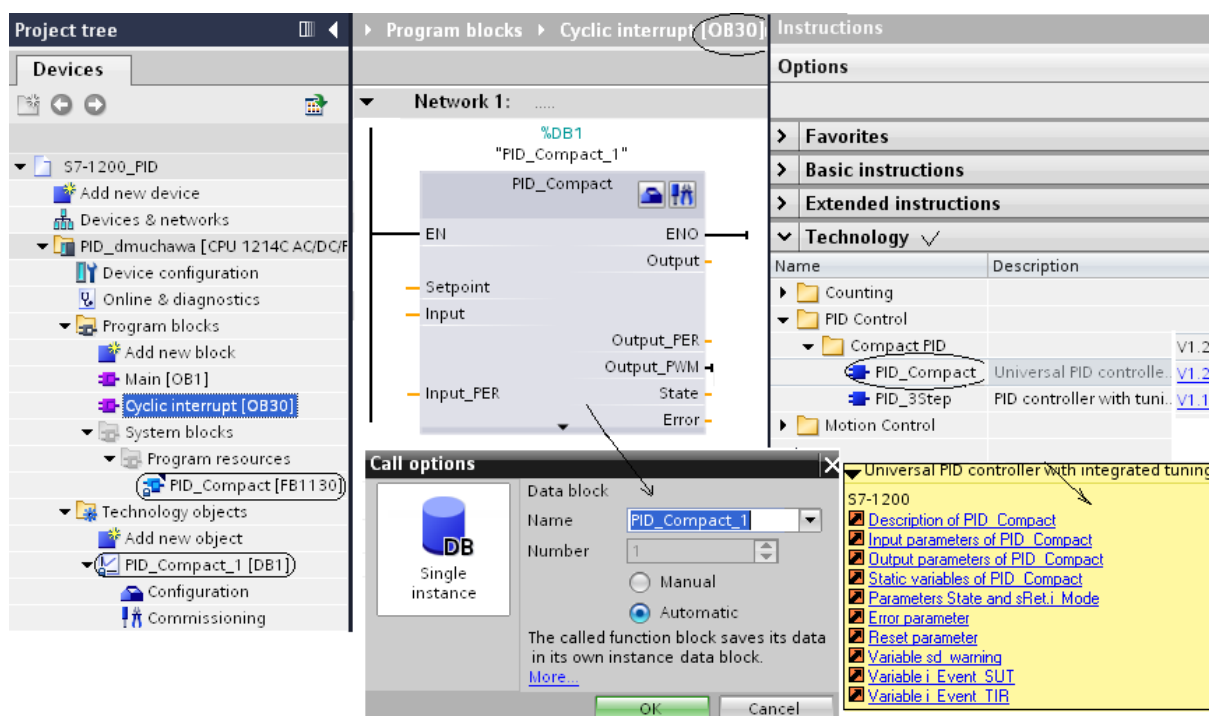
Po wprowadzonych ustawieniach, automatycznie pojawi się okno edytora oprogramowania w dodanym bloku *Cyclic interrupt* (rys. 18). Po prawej stronie okna projektu znajdują się biblioteki instrukcji. W zależności od wykorzystywanej jednostki centralnej z jej oprogramowaniem systemowym (*software*) – FW (FirmWare), instrukcje *PID_Compact* z biblioteki udostępniane są w wersjach V1 lub V2. Jednostka centralna w projekcie (FW 2.2) może korzystać tylko z instrukcji w wersji V1. Wersja V2 (S7-1200 – V2.2, S7-1500 – V2.0 i V2.1) udostępnia rozszerzoną funkcjonalność dostępną od FW 4.0 (oraz większej).

W bibliotece *Technology* należy rozwinąć folder *PID Control /Compact PID* i przeciągając myszką dodać instrukcję *PID_Compact* do Network 1. Automatycznie pojawi się okno *Call options* (przypisywanie do bloku funkcyjnego FB odpowiedniego bloku danych DB – *single instance*) gdzie należy wpisać nazwę np. *PID_Compact_1*, pozostawić *Automatic* i kliknąć OK. Następnie przypisać zmienne do bloku regulatora *PID_Compact* zdefiniowane wcześniej (rys.13, tablica zmiennych)

Setpoint– wartość zadana (SP) (0-100%), adres %MD46.

Input_PER– wejście peryferyjne, wartość aktualnej zmiennej procesowej (PV) (5530-27648), adres %IW66.

Output_PER– wyjście peryferyjne, zmienna sterująca (CV) (0-27648), adres %QW80.



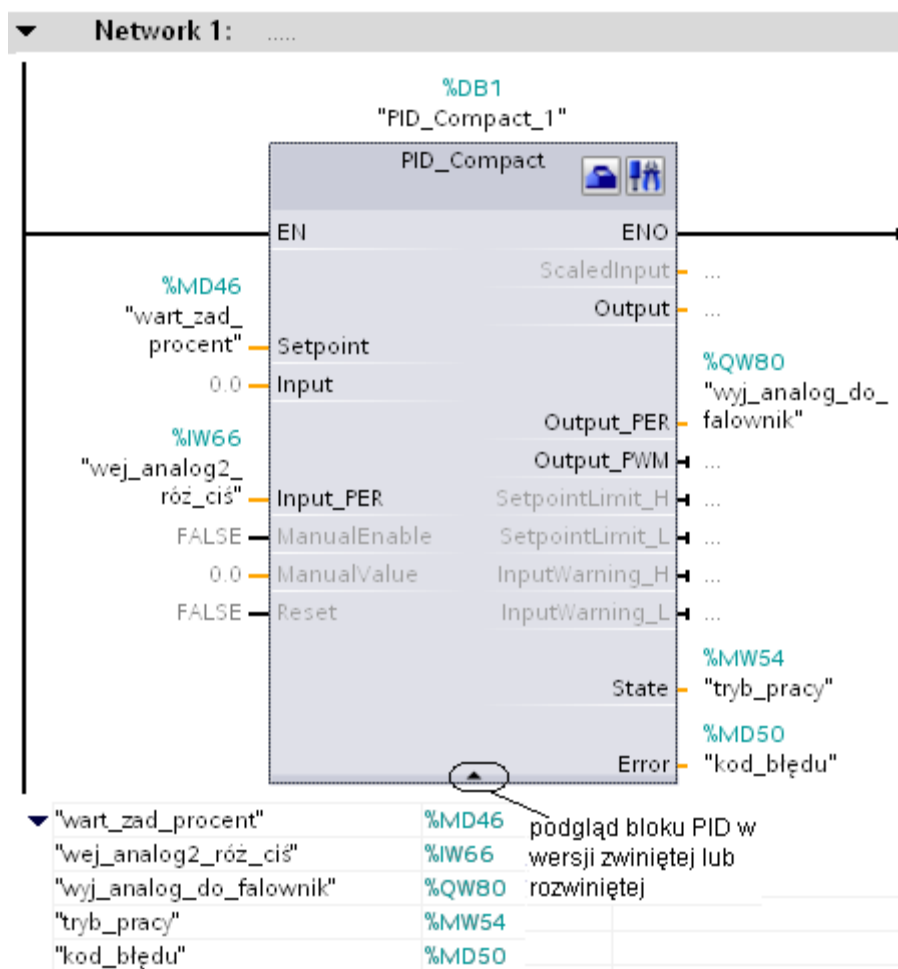
Rys. 18. Dodanie bloku funkcyjnego *PID_Compact*

W zakładce *Technology objects* pojawi się utworzony blok danych *PID_Compact_1[DB1]*, natomiast w *Program blocks/System blocks/Program resources* blok funkcyjny *PID_Compact [FB1130]*.

Na rys. 19 przedstawiono blok regulatora *PID_Compact* w wersji rozwiniętej. Wykorzystano wyjście *State* (%MW54) do informowania o bieżącym trybie pracy regulatora (0- *inactive*, 1- *pretuning*, 2- *manual fine tuning*, 3- *automatic mode*, 4- *manual mode*) i wyjście *Error* (%MD50) do powiadamiania o występującym błędzie (wartość domyślna: DW#16#0000- brak błędu).

Widoczne nieaktywne wejścia i wyjścia bloku regulatora *PID_Compact V1.2* są również przewidziane do wykorzystania. Na przykład, wejście *ManualEnable* – przejście na pracę ręczną z ustawianą wartością wyjścia regulatora (*ManualValue*) w tym trybie (wykorzystane podczas realizacji softwarowego regulatora poziomu tlenu PID, rys 98).

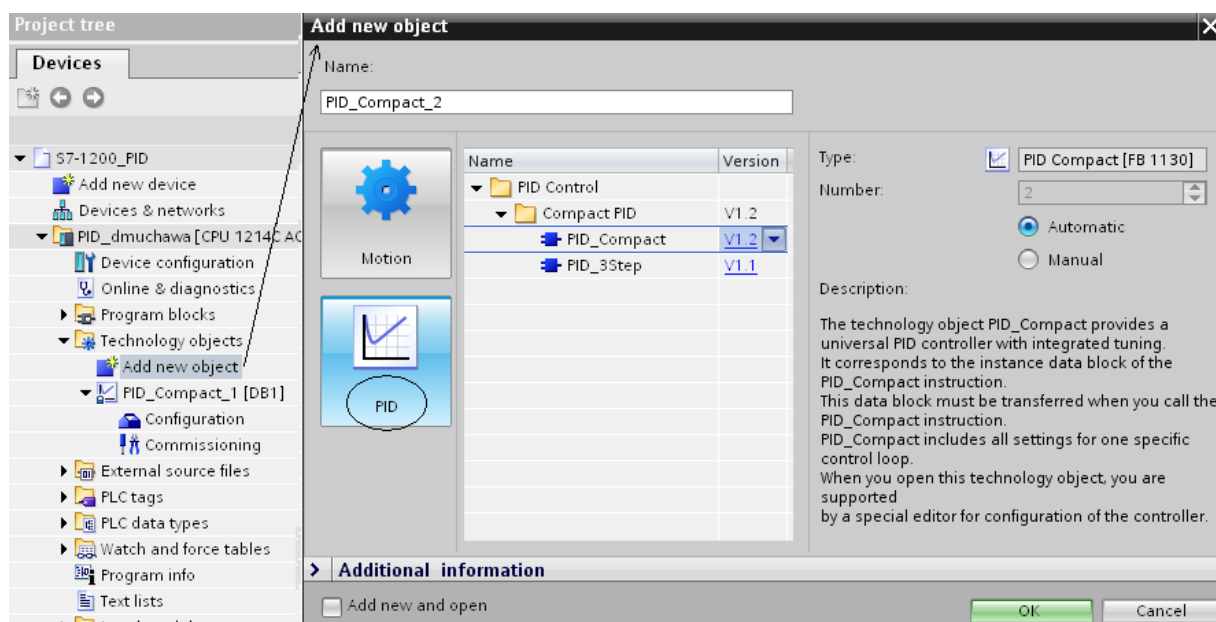
W instrukcjach *PID_Compact V2.2* (nowa wersja instrukcji, z dodatkową funkcjonalnością - *PID_Compact V2*) dla sterowników S7-1200 z FW (FirmWare) począwszy od V4.0 wykorzystana jest większa ilość sygnałów wejściowych/wyjściowych (zawiera, na przykład, wejście *ModeActivate* do ustawienia domyślnego stanu regulatora z pozycji programu aplikacyjnego). Należy wspomnieć, że jednostki z wersjami FW V1.X, V2.X i V3.X nie są już produkowane.



Rys. 19. Widok bloku *PID_Compact* w wersji rozwiniętej z przypisanymi zmiennymi.

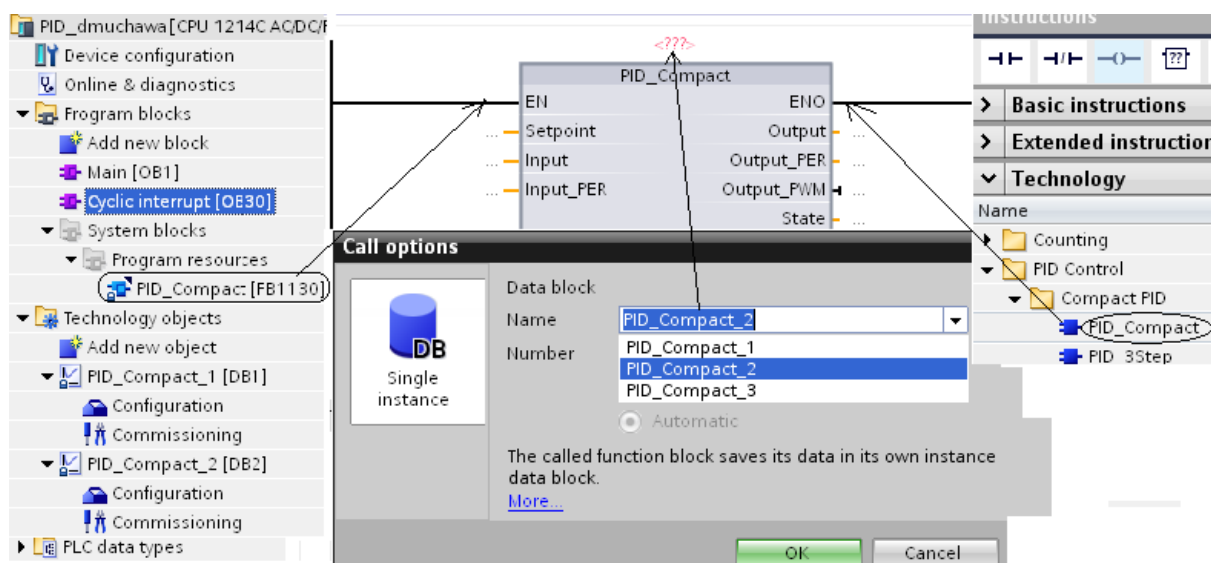
Innym sposobem dodania regulatora *PID_Compact* do projektu (przedstawiono tylko jako przykład) jest wybranie w oknie *Project tree* w zakładce *Technology objects* polecenia *Add new object* (rys. 20). Na ekranie *Add new object* wybrać „PID” i rodzaj regulatora – *PID_Compact*. Sugerowany numer przypisywanego bloku danych– *single instance* (*PID_Compact_2*) w trybie *Automatic* można zmienić na inny żądany po zaznaczeniu *Manual*. Zaznaczenie pola wyboru „Add new and open” u dołu okna dialogowego *Add new object* spowoduje otwarcie okna konfiguracji obiektu technologicznego po zakończeniu jego dodawania.

W zakładce *Technology objects* pojawi się nowoutworzony blok danych *PID_Compact_2[DB2]* z możliwością konfiguracji (*Configuration*) i uruchomienia (*Commissioning*).



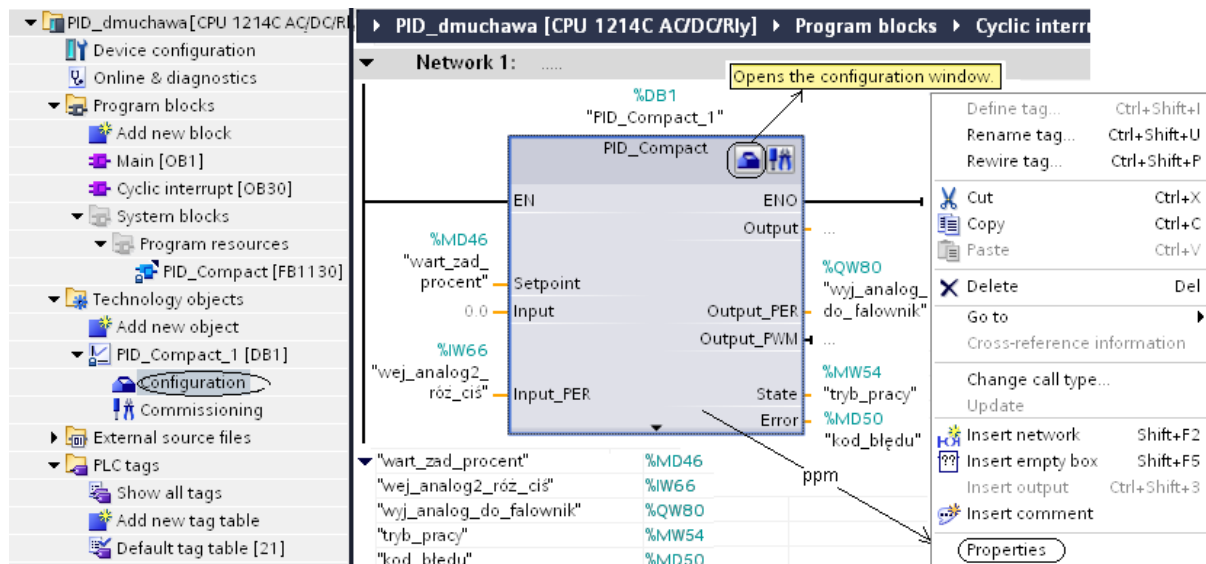
Rys. 20. Dodawanie nowego obiektu technologicznego – *PID_Compact_2* [DB2]

Aby dodać do aplikacji instrukcję regulatora PID_Compact, należy umieścić ją w otwartym bloku organizacyjnym przerwania cyklicznego OB30 przeciągając z biblioteki instrukcji *Technology/PID Control* lub z *Project tree/System blocks/Program resources* (PID_Compact [FB1130]). Następnie wymagane jest przypisanie do bloku funkcyjnego odpowiedniego bloku danych (*single instance*) *PID_Compact_2* (rys. 21).



Rys. 21. Dodanie instrukcji regulatora PID z przypisaniem bloku danych *Single instance*

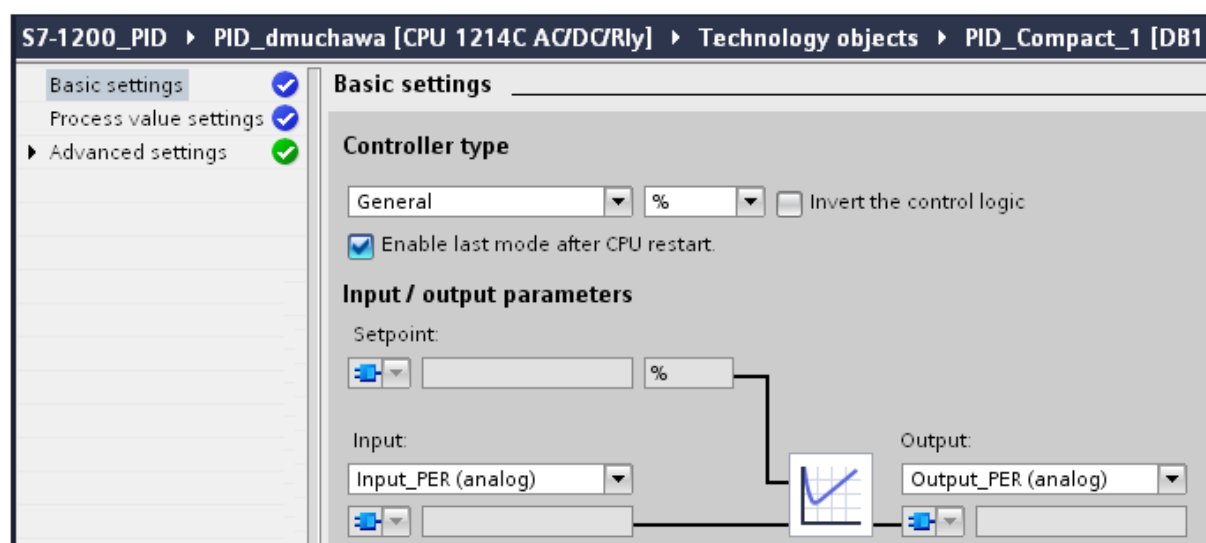
Okno konfiguracji odpowiedniego obiektu technologicznego (bloku danych *Single instance* dla bloku funkcyjnego *PID_Compact* [FB1130]) można wywołać na dwa sposoby: w prawym górnym rogu bloku *PID_Compact* należy kliknąć ikonę „Configuration” lub wybrać w oknie *Project tree* zakładkę *Technology objects/PID_Compact_1 [DB1]/Configuration* (rys. 24).



Rys. 24. Wywołanie okna konfiguracji regulatora *PID_Compact*

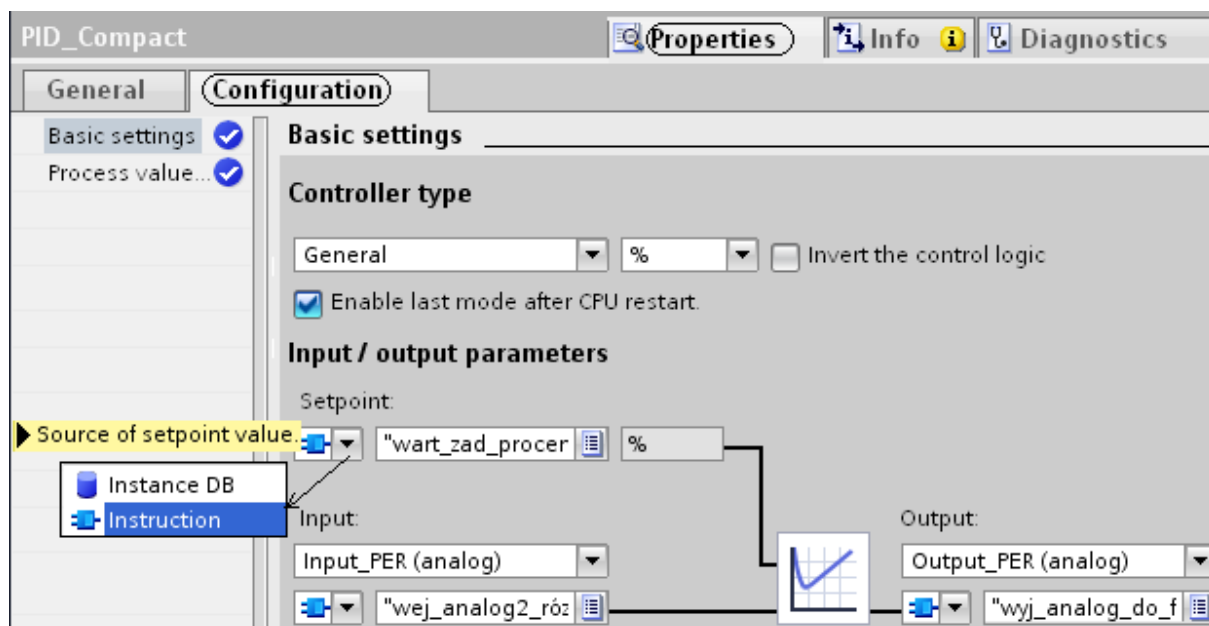
W oknie konfiguracji bloku funkcyjnego *PID_Compact* (rys.25) ikony w obszarze nawigacji podczas konfiguracji pokazują dodatkowe szczegóły dotyczące kompletności ustawień : niebieskie kółko z znakiem akceptacji oznacza domyślne wartości ustawień i ich kompletność, zielone kółko z znakiem akceptacji oznacza definiowanie wartości ustawień przez użytkownika i ich kompletność, czerwone kółko z znakiem „x” oznacza błędne definiowanie wartości ustawień przez użytkownika lub ich niekompletność.

W oknie konfiguracji dla zakładki *Basic settings* w *Controller type* wybrano *General*, nie zaznaczono opcji *Invert the control logic* oraz wybrano w polu *Input/output parameters*: *Input_PER (analog)* i *Output_PER (analog)*.



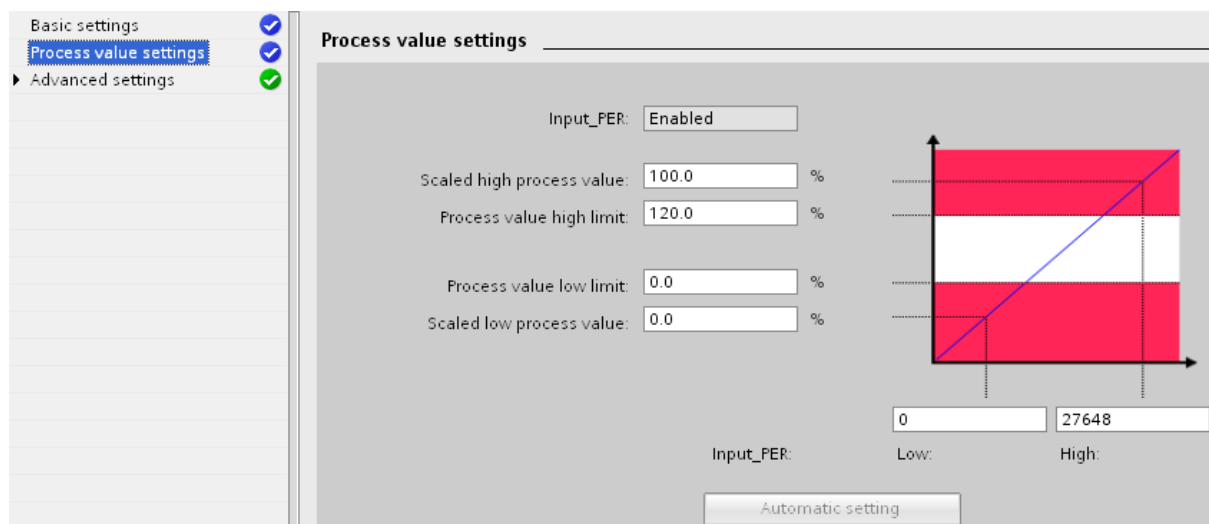
Rys. 25. *PID_Compact* konfiguracja z *Basic settings*

Można też dojść do parametrów konfiguracyjnych (jednak bez zakładki *Advanced settings*) poprzez kliknięcie (ppm) bloku funkcyjnego *PID_Compact* w OB30 i wyboru okna *Properties* a w nim zakładki *Configuration* (rys.26). Poprzez wypełnienie pól *Input/output parameters* opisywany jest (opcjonalnie) blok *PID_Compact* (rys.24).



Rys. 26. PID_Compact konfiguracja z *Basic settings* poprzez wybór okna *Properties* bloku

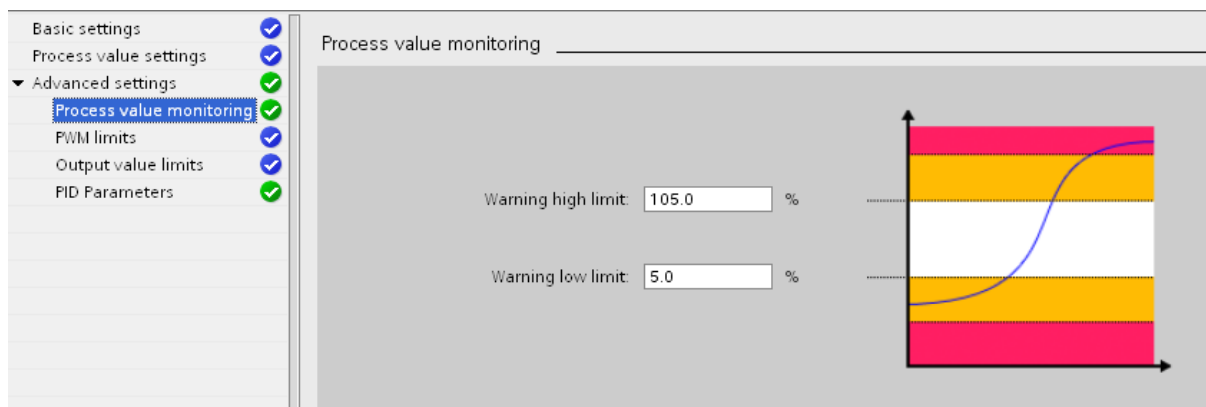
W kolejnej zakładce *Process value settings* (rys. 27) ustawiane są specyfikacje wartości wejściowej, zmiennej procesowej. W polu *Input_PER* ustawiono wartość *Low* na **0** a wartość *High* na **27648** (wbudowane w CPU wejście analogowe : 0 – 10V, rozdzielczość – 10bit, zakres pomiarowy 0 – 27648) . min – 5530???



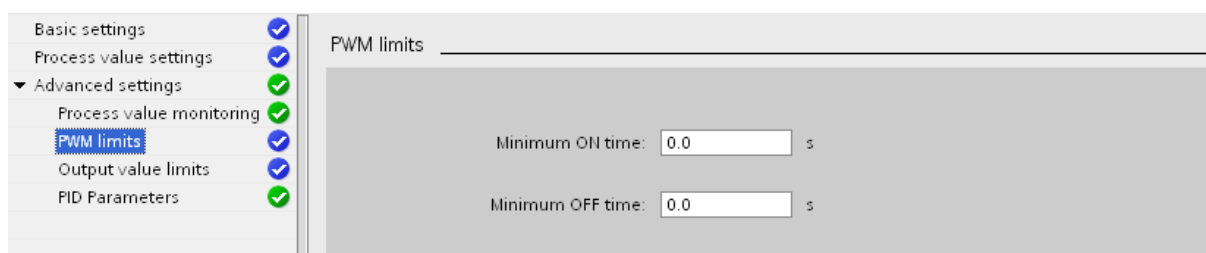
Rys. 27. PID_Compact konfiguracja z *Process value settings*

Zakładka *Advanced settings* zawiera strony:

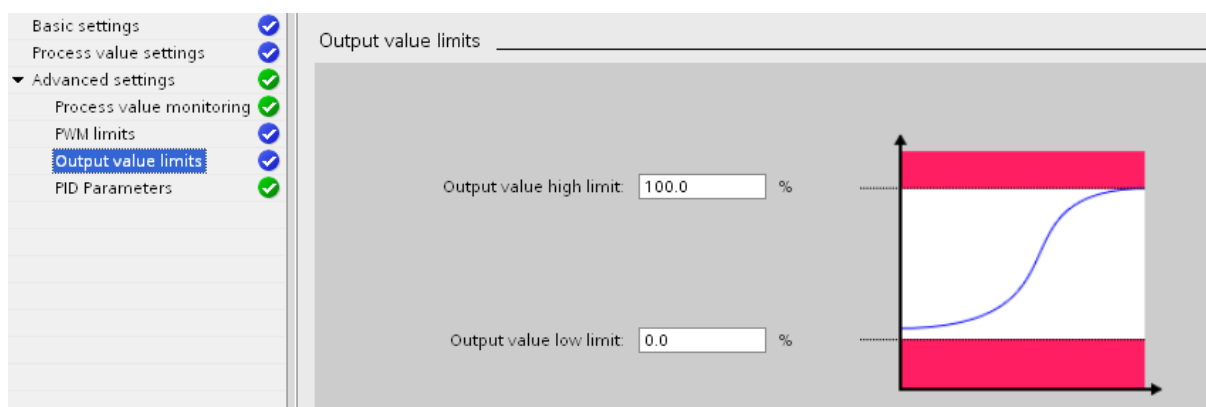
Process value monitoring (rys. 28), *PWM limits* (rys. 29), *Output value limits* (rys. 30) umożliwiające wprowadzenie różnych opcji charakteryzujących wartość wejściową i wyjściową.



Rys. 28. PID_Compact konfiguracja z *Advanced settings- Proces value monitoring*

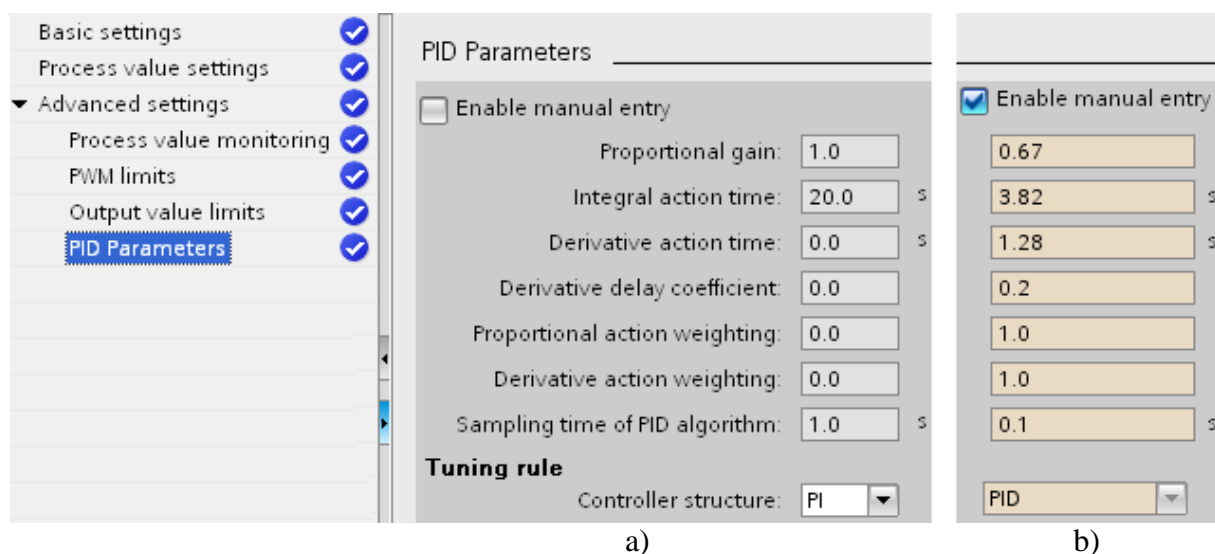


Rys. 29. PID_Compact konfiguracja z *Advanced settings- PWM limits*



Rys. 30. PID_Compact konfiguracja z *Advanced settings – Output value limits*

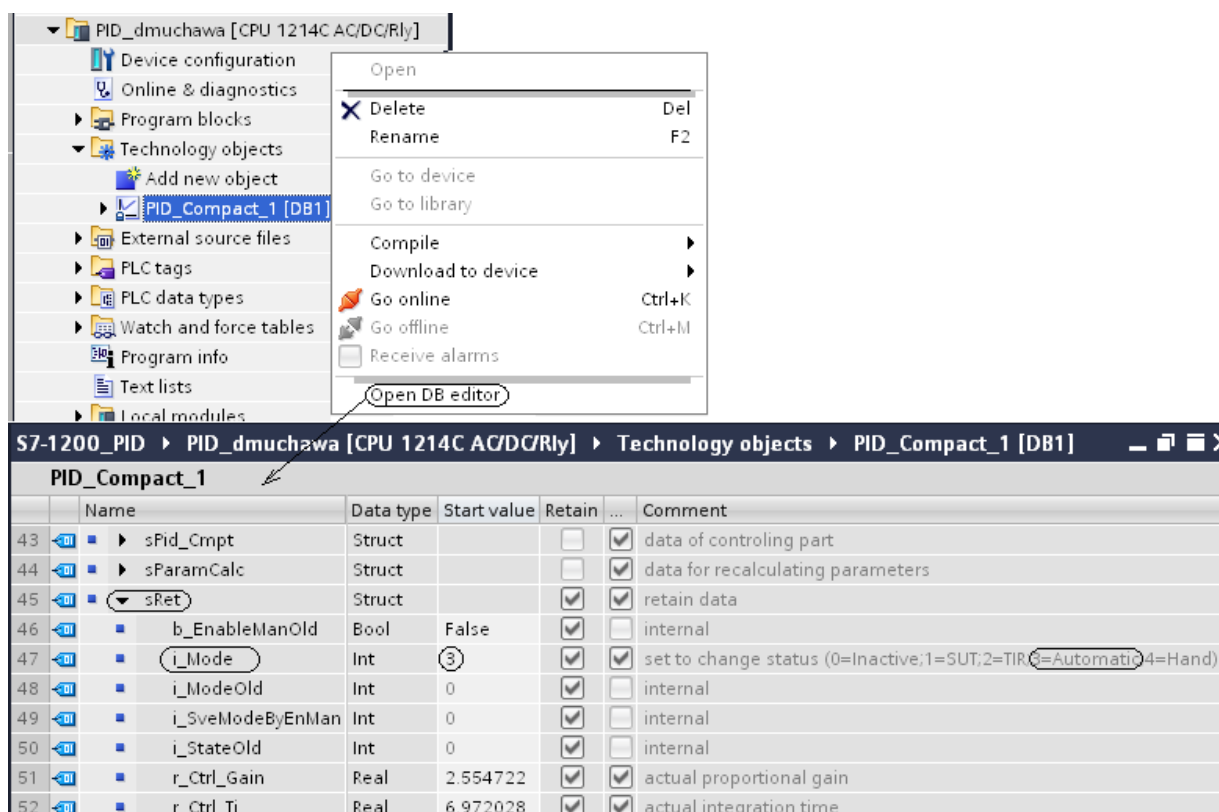
Strona *PID Parameters* zakładki *Advanced settings* służy do wprowadzenia własnych, obliczonych nastaw regulatora PID (po zaznaczeniu pola *Enable manual entry*). W przypadku wykorzystania opcji automatycznego dobierania nastaw regulatora nie zaznacza się pola *Enable manual entry* (praktycznie, pozostawienie zaznaczonego pola nie przeszkadza w przeprowadzeniu *Autotuning*’u), natomiast w polu *Controller structure* wybrany zostaje algorytm – „PID” lub „PI” (w zależności od potrzeb). Domyślnym ustawieniem w zakładce jest niezaznaczone pole *Enable manual entry* oraz struktura „PI”(rys.31a).



Rys. 31. PID_Compact konfiguracja z *Advanced settings* – *PID Parameters* a) ustawienia domyślne b) ustawienia obliczone przez jednostkę wielofunkcyjną SIPART DR24

UWAGA. W załączonym *Dodatku* (na końcu danej instrukcji laboratoryjnej) podano schemat blokowy struktury regulatora *PID_Compact*, jego transmitancję wraz z opisem parametrów.

Przed wgraniem projektu do sterownika, ręcznie wprowadzono (przy zaznaczonym w tym czasie polu *Enable manual entry*) obliczone przez jednostkę wielofunkcyjną SIPART DR24 [Solnik W., Zajda Z. :Sieć Profibus DP w praktyce przemysłowej. Przykłady zastosowań. Wydawnictwo BTC, Legionowo 2013, rozdz.8.4.2] nastawy regulatora PID (rys. 31b) . Aby domyślnie (po załadowaniu programu do sterownika) regulator PID włączał się w trybie *Automatic* a nie *Inactive* , należy w oknie konfiguracji projektu *Project tree* rozwinąć zakładkę *Technology objects*. Następnie kliknąć prawym przyciskiem myszy zakładkę *PID_Compact_1[DB1]* oraz wybrać *Open DB editor* (rys. 32).

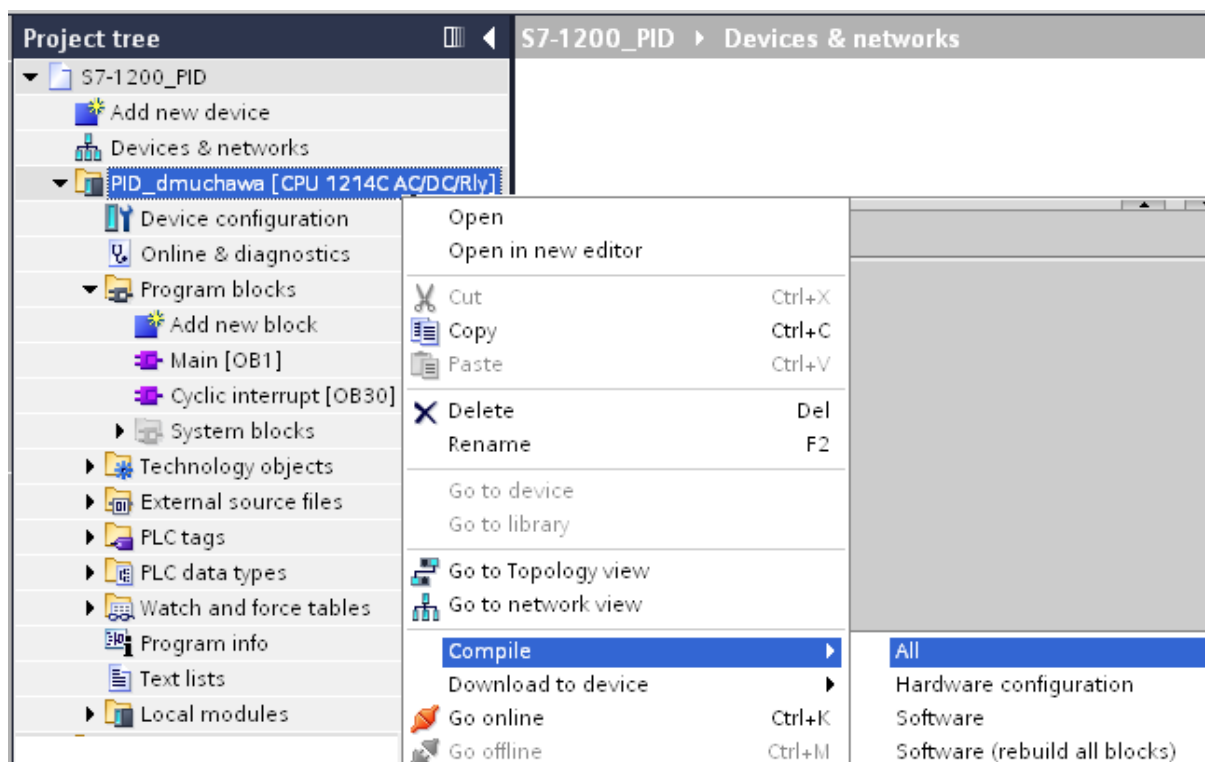


Rys. 32. Wybór w oknie *Project tree* edytora bloku danych DB1 - *Open DB editor*

W otwartym edytorze bloku danych DB1 - *Open DB editor* rozwinąć zakładkę *sRet* i następnie w wierszu o nazwie *i_Mode* w polu *Start value* wpisać wartość „3” (*Automatic*) (struktura okna edytora DB dla instrukcji wersji **V2** jest w znacznym stopniu inna, np nazwy zmiennych)

Wgranie projektu do PLC

Należy kliknąć prawym przyciskiem myszy w oknie *Project tree* na *PID_dmuchawa [CPU 1214C AC/DC/Rly]*. Wybrać zakładkę *Compile* a następnie wybrać *All* (rys. 33).



Rys. 33. Kompilacja projektu (dotyczy *PID_dmuchawa [CPU 1214C AC/DC/Rly]*)

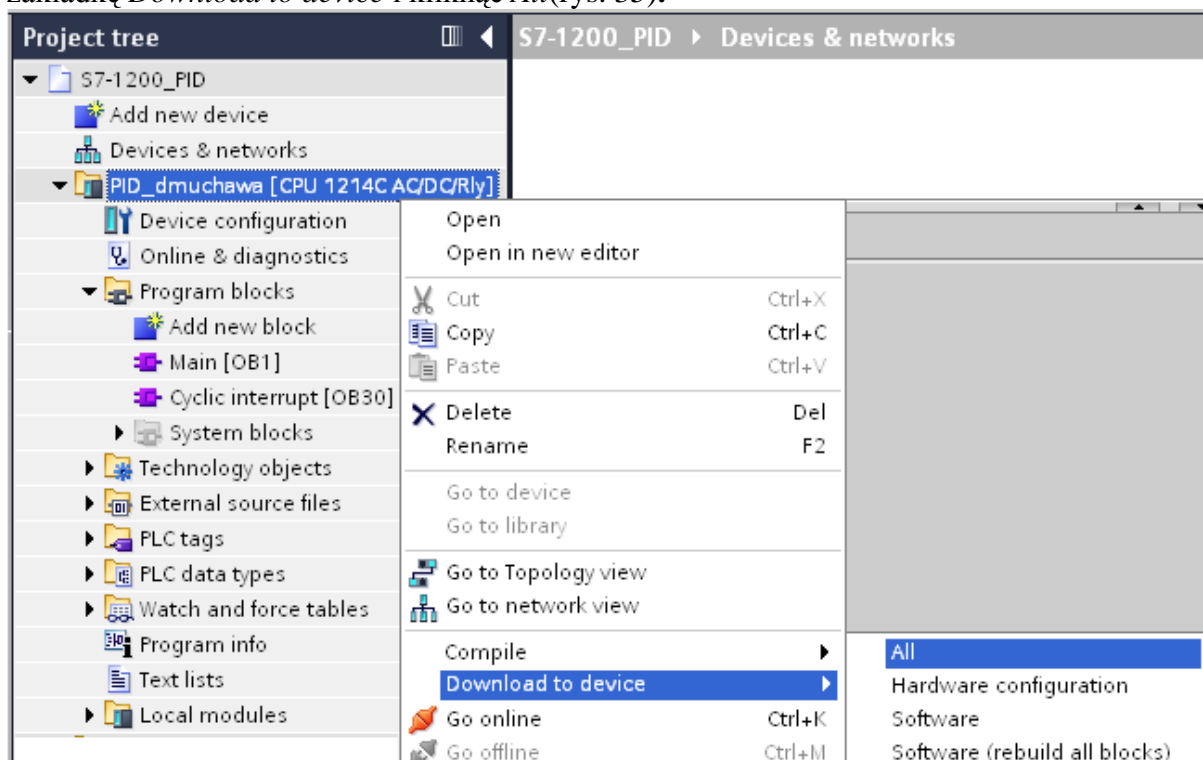
Na rys. 34 przedstawiono wynik kompilacji dla dwóch przypadków – jeden z ostrzeżeniem że „strojenie jeszcze się nie rozpoczęło” (*Tuning has not been started yet*), tylko gdy pozostawiono domyślne ustawienia parametrów regulatora (rys. 31a). W drugim przypadku kompilacja kompletna bez ostrzeżenia – **wprowadzono w domyślnych ustawieniach zmianę parametru *Proportional gain* z 1.0 na 2.0.**

General ⓘ Cross-references Compile					
Compiling completed (errors: 0; warnings: 1)					
!	Path	Description	Errors	Warnings	Time
⚠	▼ PID_dmuchawa		0	1	1:43:40 PM
✓	Hardware configuration		0	0	1:43:40 PM
⚠	▼ Program blocks		0	1	1:43:43 PM
⚠	▼ PID_Compact_1 (DB1)		0	1	1:43:44 PM
⚠	Tuning	Tuning has not been started yet.	0	1	1:43:44 PM
ℹ		Block was successfully compiled.	0	0	1:43:45 PM
✓	Cyclic interrupt (OB30)	Block was successfully compiled.	0	0	1:43:45 PM
⚠	Compiling completed (errors: 0; warnings: 1)		0	1	1:43:47 PM

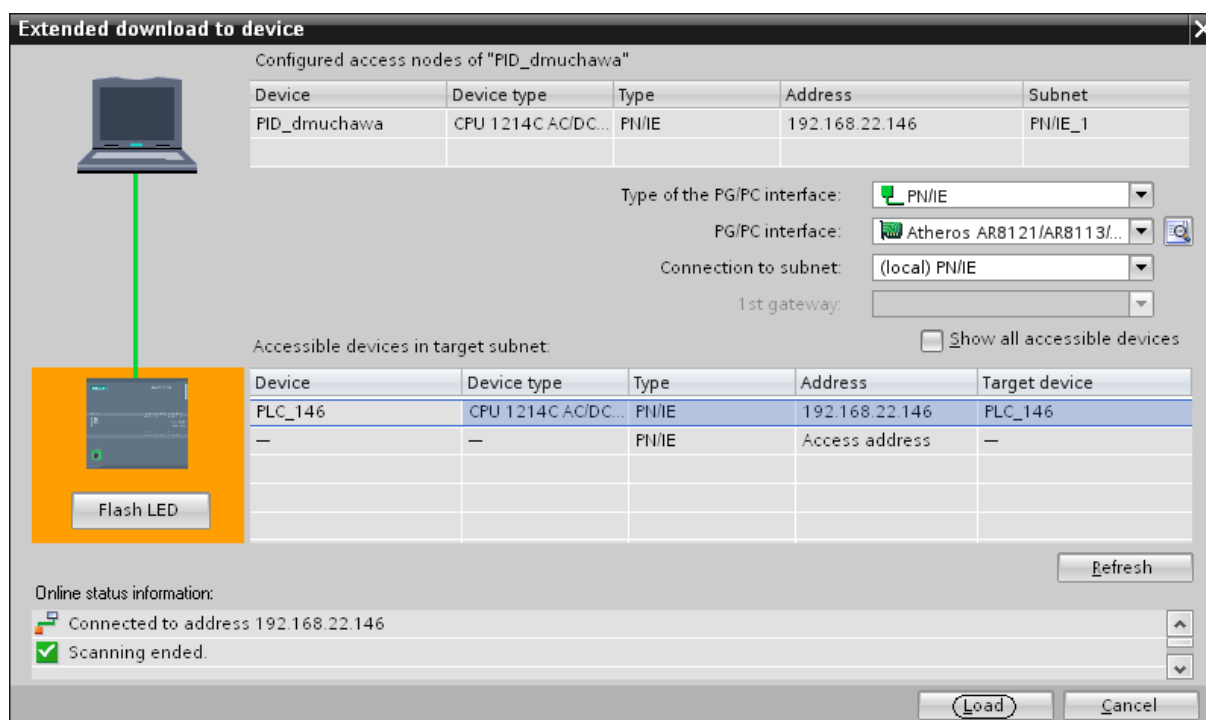
General ⓘ Cross-references Compile					
Compiling completed (errors: 0; warnings: 0)					
!	Path	Description	Errors	Warnings	Time
✓	▼ PID_dmuchawa		0	0	6:53:03 PM
✓	Hardware configuration		0	0	6:53:03 PM
✓	▼ Program blocks		0	0	6:53:12 PM
✓	Cyclic interrupt (OB30)	Block was successfully compiled.	0	0	6:53:17 PM
✓	Main (OB1)	Block was successfully compiled.	0	0	6:53:21 PM
✓	Compiling completed (errors: 0; warnings: 0)		0	0	6:53:22 PM

Rys. 34. Wyniki kompilacji projektu (*Hardware configuration* oraz *Software*)

Po udanej kompilacji można przesłać program do sterownika. W tym celu należy wybrać zakładkę *Download to device* i kliknąć *All*(rys. 35).

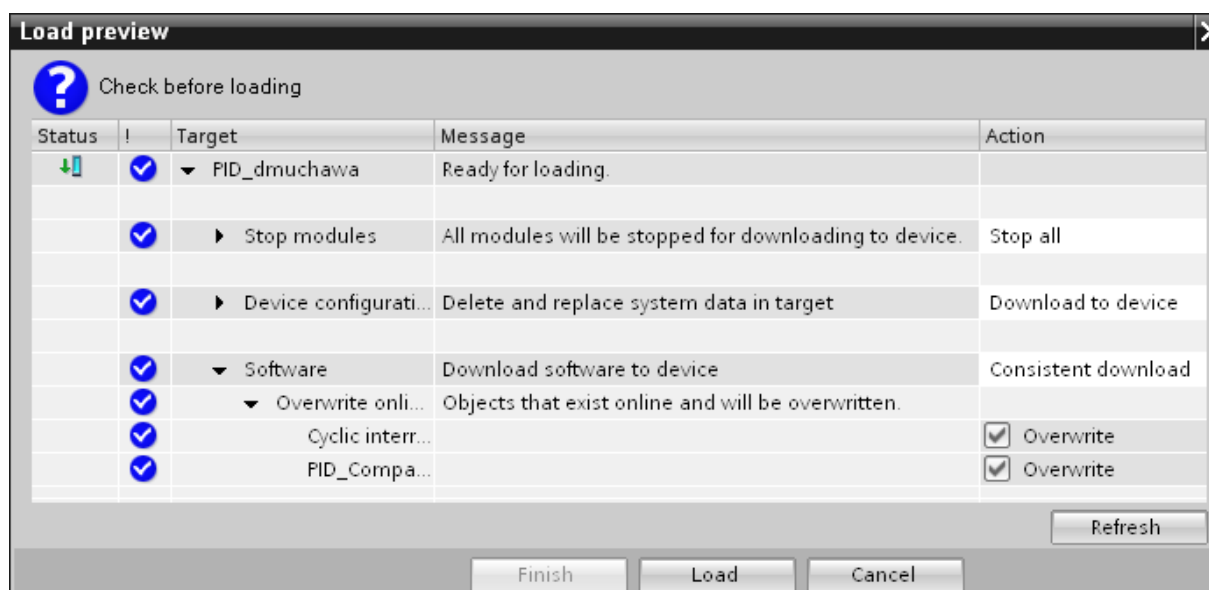


Rys. 35. Przesłanie projektu do sterownika *PID_dmuchawa [CPU 1214C AC/DC/Rly]*
Proces przesyłania rozpocznie się od wyznaczenia ścieżki do sterownika. Na rys.36 pokazano wynik wyznaczenia sterownika o konkretnym adresie IP w którym znajduje się aplikacja z aktualną nazwą sterownika (*PLC_146*).



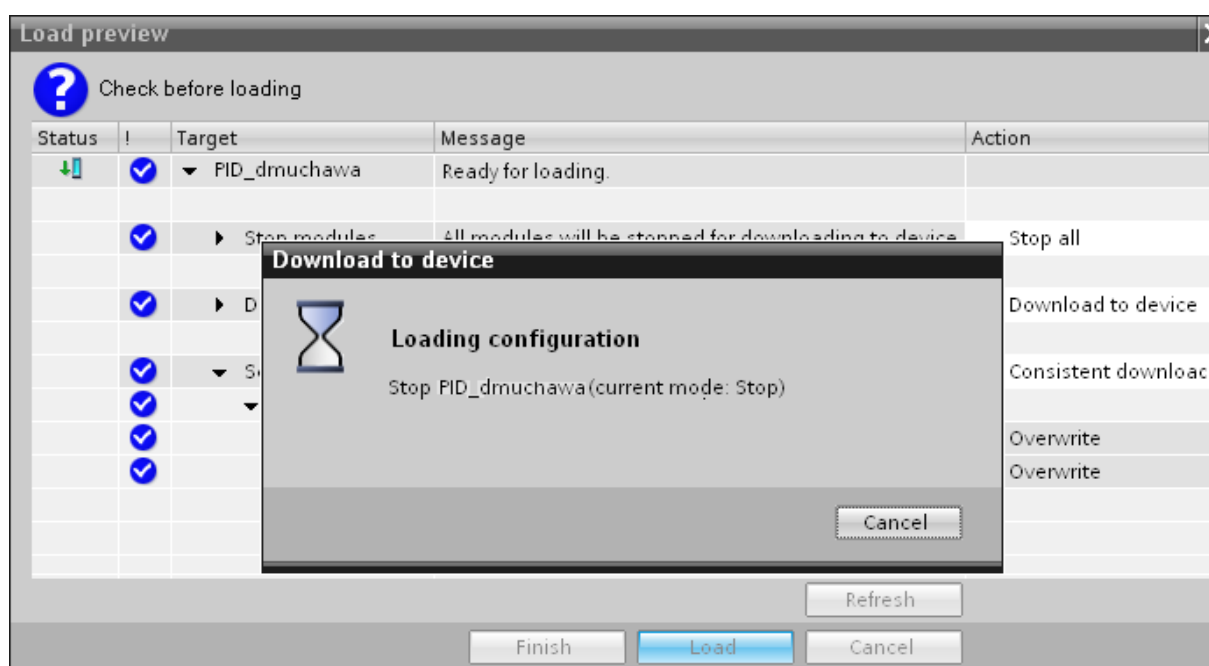
Rys. 36. Wyznaczenie ścieżki do sterownika podczas operacji przesyłania aplikacji

Nastąpi na początku również kompilacja projektu przed przesłaniem do sterownika, której wynik przedstawia rys. 37 - jest możliwość rozpoczęcia przesyłania (*Load*).



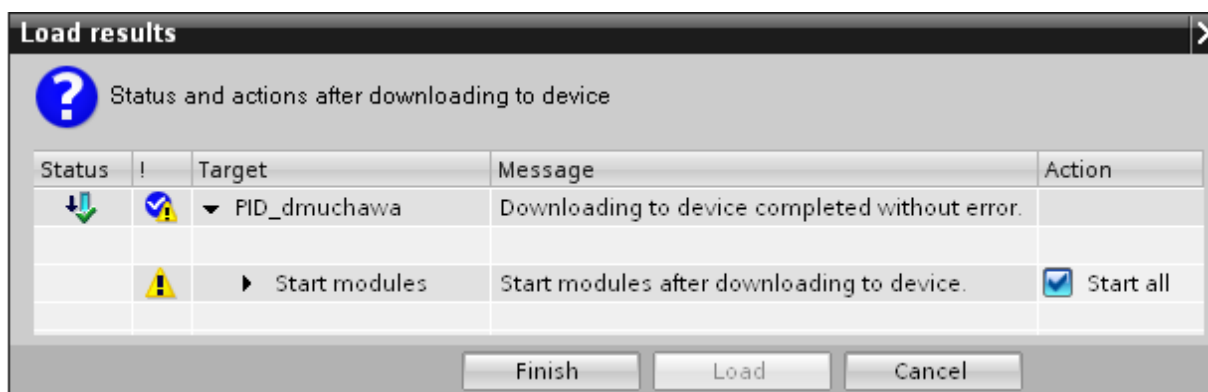
Rys. 37. Wynik kompilacji projektu z możliwością rozpoczęcia przesyłania do sterownika.

Rys. 38 przedstawia proces ładowania do sterownika kolejno konfiguracji sprzętowej i aplikacji. Sterownik powinien być podczas ładowania programu aplikacyjnego z nowymi nastawami regulatora w trybie *STOP*. Jeżeli sterownik był w trybie *RUN*, to nowe wartości nastaw będą przyjęte przez regulator po wyłączeniu i ponownym włączeniu zasilania sterownika. Po przesłaniu programu do sterownika rozpocznie on działanie z nowymi nastawami regulatora PID.



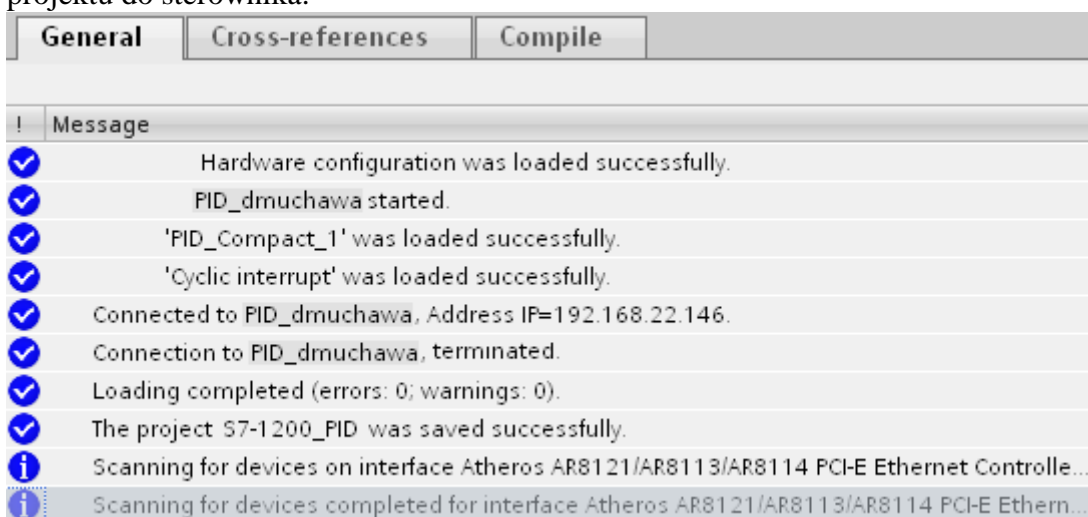
Rys. 38. Przebieg przesyłania projektu do sterownika

Rys. 39 przedstawia wynik przesłania projektu do sterownika, można tu po zaznaczeniu *Start All* kliknąć *Finish*



Rys. 39. Wynik przesłania projektu do sterownika

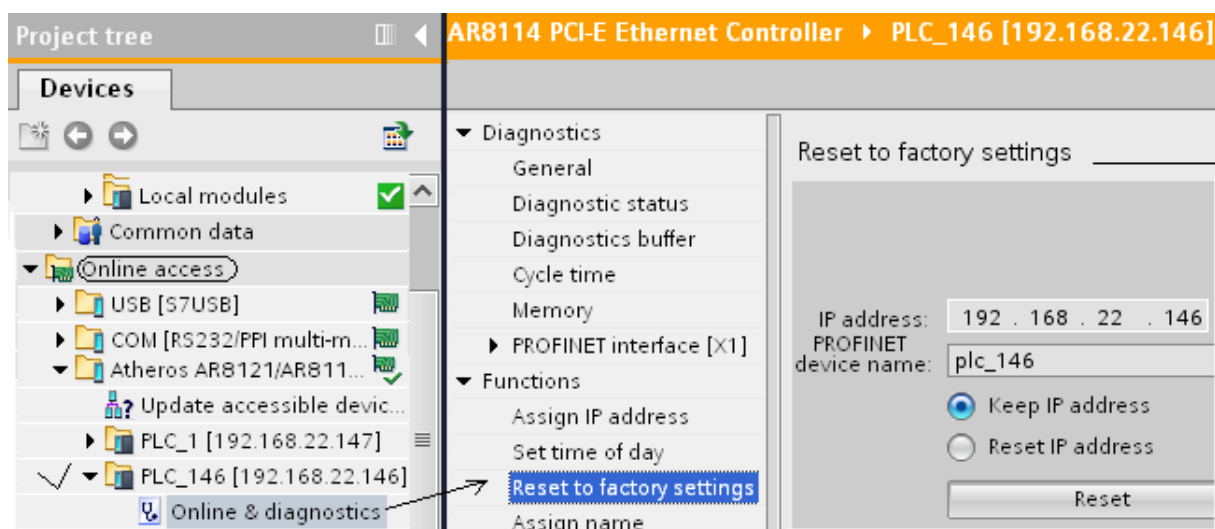
Na rys. 40 przedstawiono ogólne wyświetlane komunikaty związane z procesem ładowania projektu do sterownika.



Rys. 40. Komunikaty ogólne wyświetlane po zakończeniu procesu ładowania projektu do sterownika.

Jeżeli w sterowniku S7-1200 znajduje się projekt realizowany przy pomocy innej wersji TIA Portal (STEP 7 Vxx), to wymagane jest wstępnie zresetowanie do ustawień fabrycznych (inaczej podczas przesyłania projektu wykonanego w środowisku TIA Portal V11 pojawi się informacja o błędzie wewnętrznym).

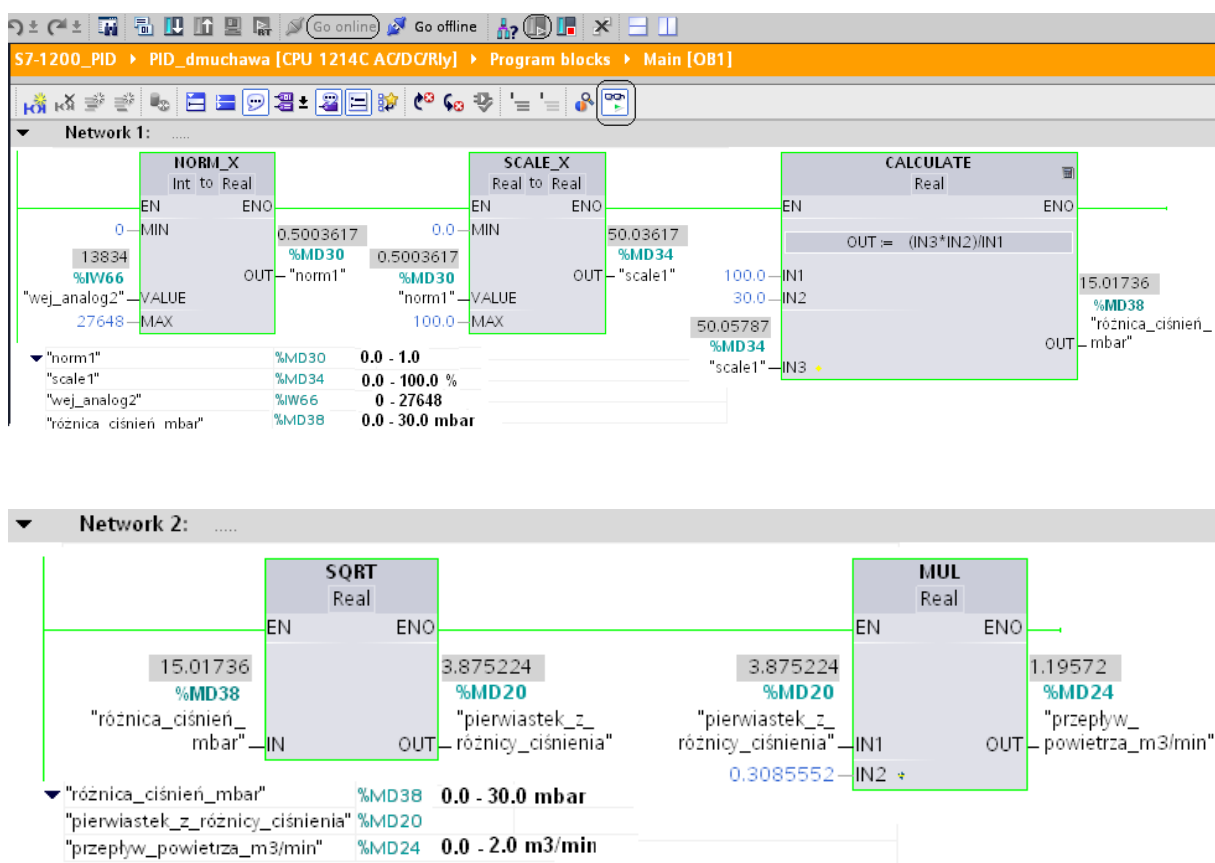
Przywracanie ustawień fabrycznych (w trybie *STOP* oraz z wyjętą kartą pamięci SD) można przeprowadzić za pomocą narzędzia programistycznego np. S7-1200 tool, czystej karty SD lub wykorzystywanego STEP 7 V11. Na rys. 41 przedstawiono okno z funkcją resetu sterownika do ustawień fabrycznych w TIA Portal V11. Po kliknięciu „Online & diagnostics” w oknie nawigacyjnym *Project tree* wybrać polecenie „Reset to factory settings” w zakładce *Functions*. Przed akceptacją wykonania polecenia („Reset”) zwrócić uwagę na możliwość zachowania dotychczasowego adresu IP sterownika („Keep IP address”).



Rys. 41. Okno z funkcją resetu sterownika do ustawień fabrycznych

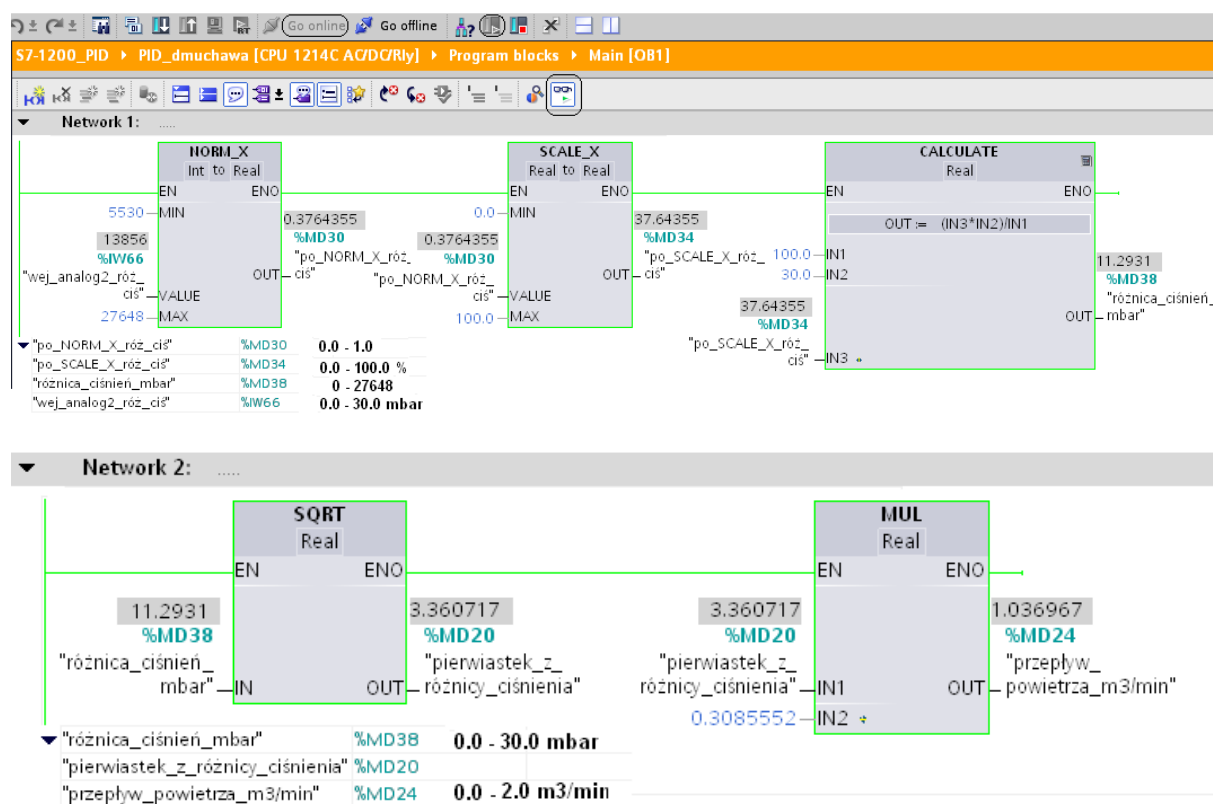
Po poprawnym wgraniu projektu za pomocą STEP 7 V11 (TIA Portal) do sterownika, podłączeniu się „Go online”, w trybie *RUN* można sprawdzić działanie części aplikacji w bloku organizacyjnym OB1. W tym celu włączyć opcję *Monitoring* w górnym pasku narzędzi (ikona „okulary”). Jeżeli sterownik nie był w trybie *Online*, to wybór opcji monitorowania może spowodować najpierw próbę podłączenia się do sterownika. Pojawi się wtedy okno wyboru sterownika z lokalnej sieci ethernetowej do pracy w trybie *Go online*.

Na rys. 42 przedstawiono monitorowanie części aplikacji w bloku organizacyjnym OB1 w trybie *Online*.



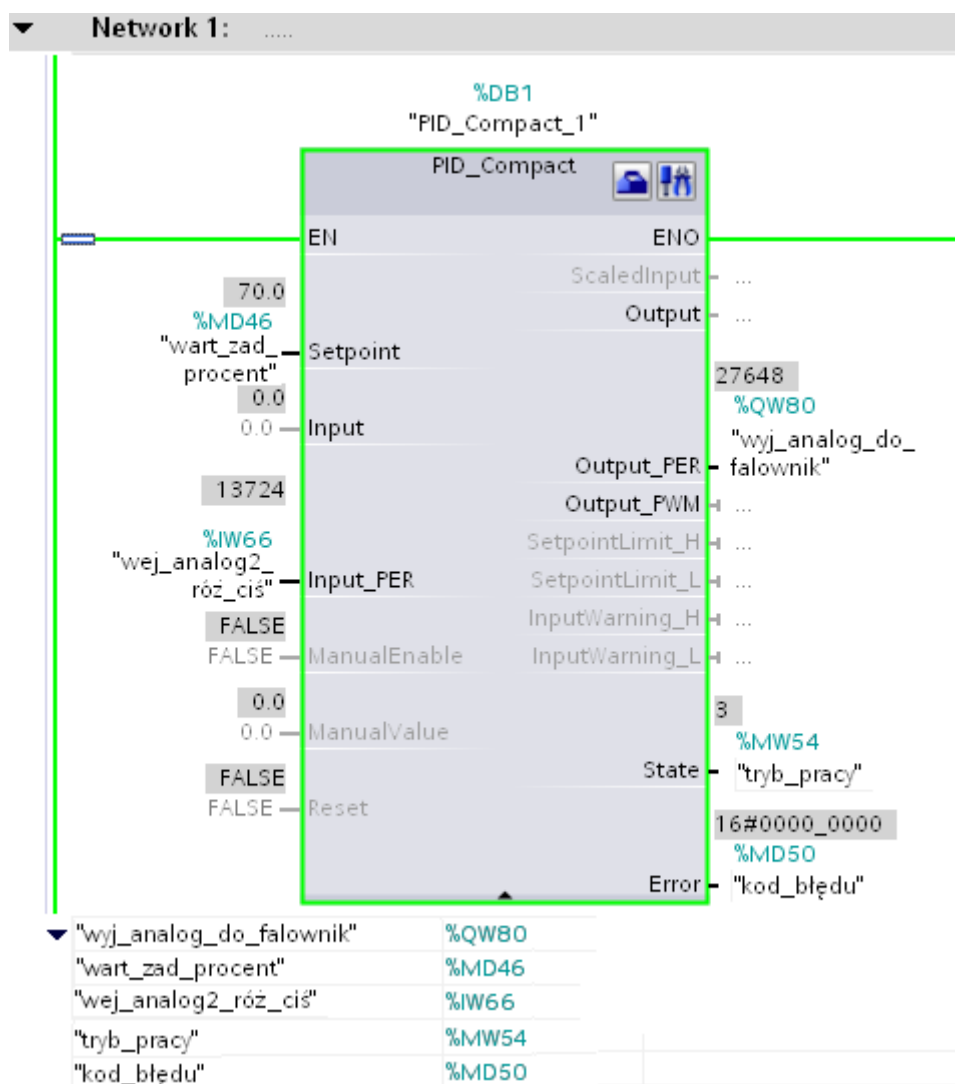
Rys. 42. Monitorowanie działania programu w OB1 (**wariant 1**)

Z PRZESUNIĘCIEM POCZĄTKU – „5530” (4ma)



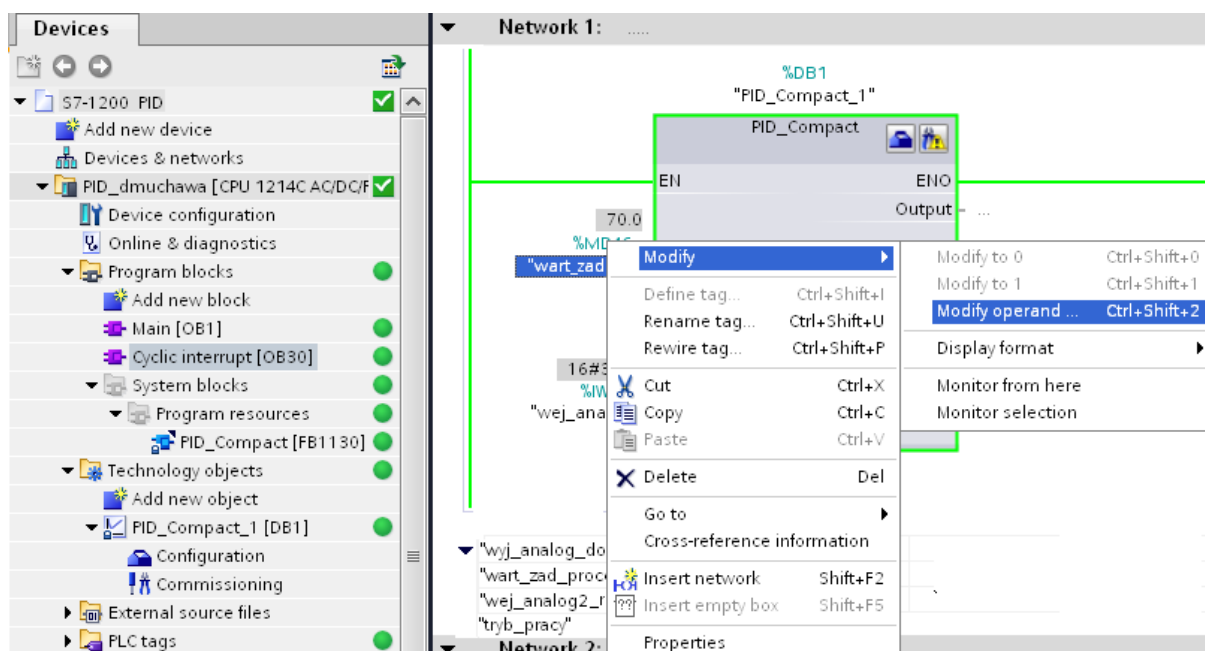
Rys. 42. Monitorowanie działania programu w OB1 (**wariant 2**)

Aby wprowadzić wartość zadaną do bloku PID_Compact, należy w oknie *Project tree* otworzyć *Cyclic interrupt* [OB30]. Na rys. 43 przedstawiono blok PID_Compact w OB30 w wersji rozwiniętej w trybie *Monitoring*.



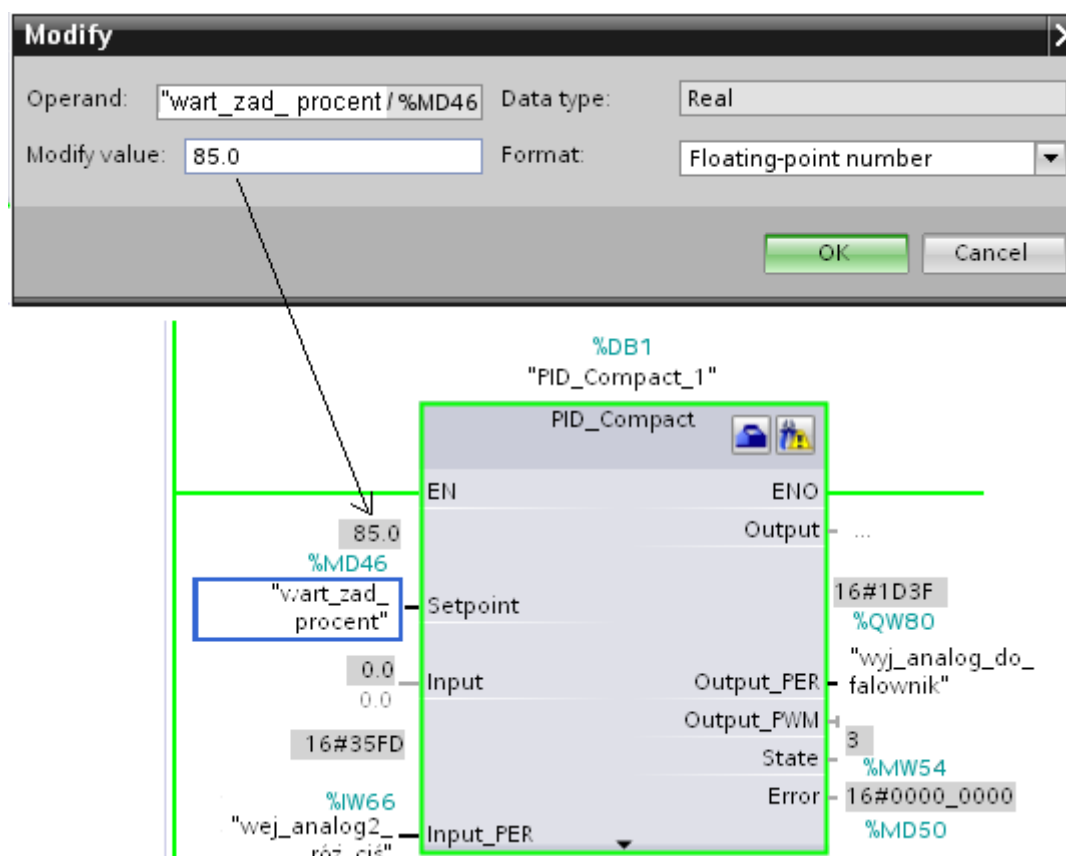
Rys. 43 Blok *PID_Compact* w OB30 w wersji rozwiniętej w trybie *Monitoring*.

Po kliknięciu prawym przyciskiem na nazwę „wart_zad_procent” zmiennej (%MD46) w bloku *PID_Compact* (wejście *Setpoint*) wybrać *Modify/Modify operand* (rys. 44).



Rys. 44. Otwieranie okna do wprowadzenia wartości zadanej w regulatorze.

W oknie do modyfikacji wartości zmiennych (wybrano „wart_zad_procent”- %MD46) w pozycji *Modify value* wpisać wartość np. 85.0 (rys. 45).



Rys. 45. Ustawianie wartości zadanej w oknie modyfikacji zmiennych.

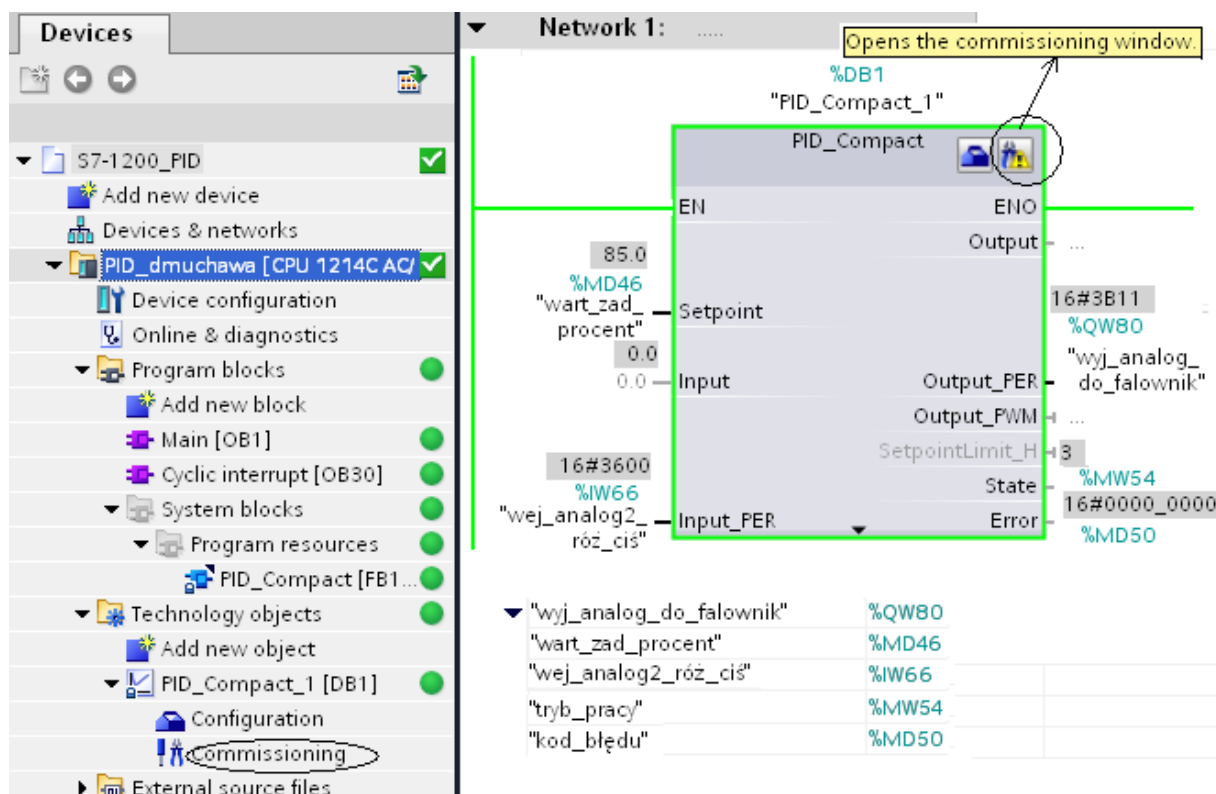
Do podglądu zmiennych w trybie *Online* wykorzystuje się wstępnie utworzoną tablicę *Watch table_1* w której również istnieje możliwość modyfikacji wartości zadanej - %MD46 (rys. 46).

[CPU 1214C AC/DC/Rly] ▶ Watch and force tables ▶ Watch table_1						
						0.5* 27648
	Name	Address	Display format	Monitor value	Modify value	
1	"wej_analog2_róż_cis"	%IW66	DEC_unsigned	13824		
2	"wart_zad_procent"	%MD46	Floating-point nu...	50.0	50.0 %	
3	"wyj_analog_do_falownik"	%QW80	DEC_unsigned	14938		

Rys. 46. Podgląd zmiennych w trybie *Online* – Watch table_1.

3.3.1.1 Automatyczny dobór nastaw (*Autotuning*)

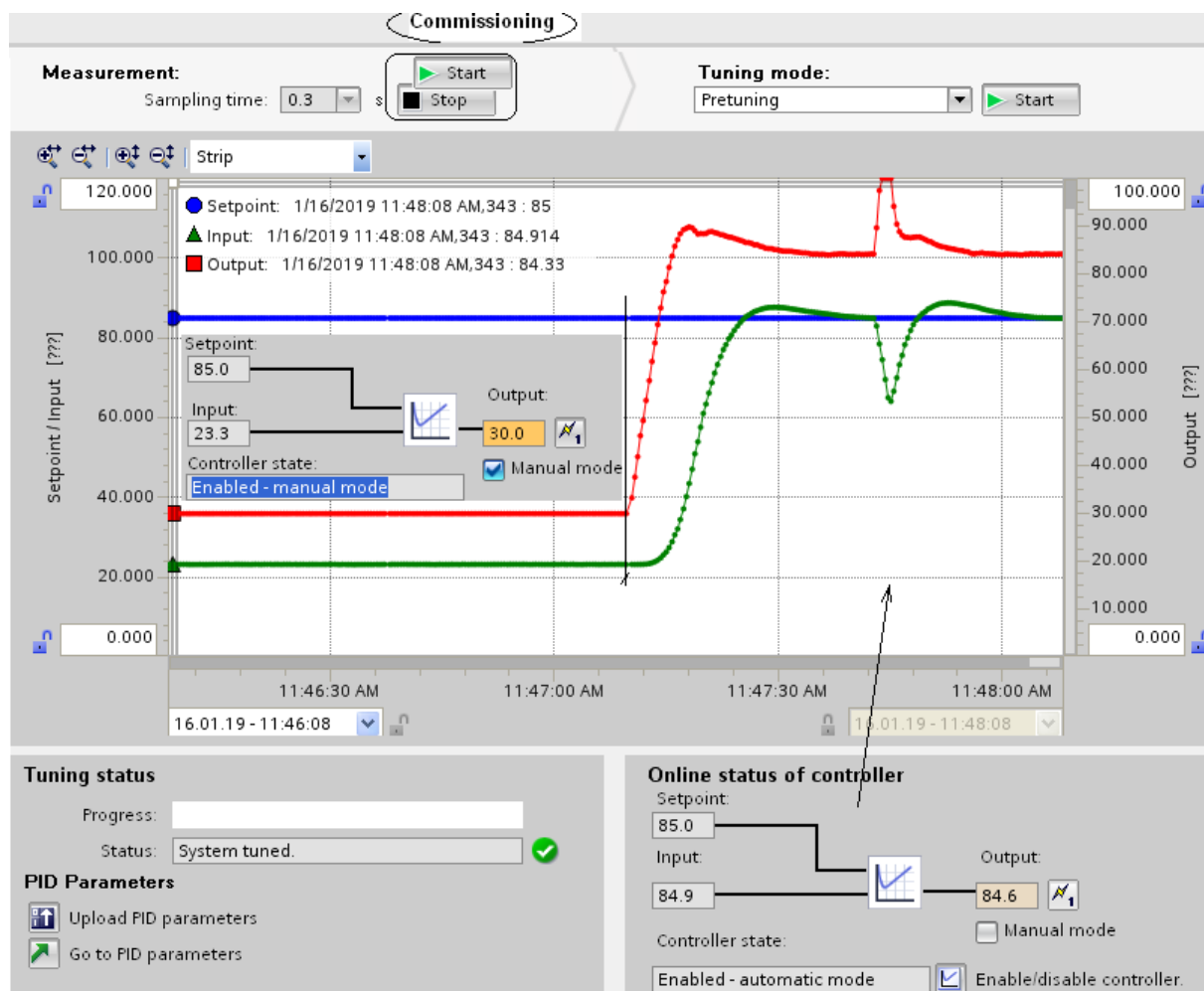
Następnym krokiem będzie przeprowadzenie procesu automatycznego doboru nastaw (*Autotuning*) w dwóch trybach (*Pretuning* oraz *Fine tuning*) dla skonfigurowanego regulatora PID. W prawym górnym rogu bloku *PID_Compact* kliknąć ikonę *Commissioning* (uruchomienie, rozruch) lub wybrać w oknie *Project tree* zakładkę *Technology objects/PID_Compact_1 [DB1]/Commissioning* (rys. 47).



Rys. 47. Wybór opcji *Commissioning* regulatora *PID_Compact* (OB30).

W nowo otwartym oknie wykresu *Tuning* (rys. 48) ustawiane są zakresy zmian pomiarów *Setpoint* (SP)/*Input* (PV) oraz *Output* (CV), a także (w polu *Measurement*) okres ich próbkowania. Okres próbkowania wartości pomiarowych wykorzystywany jest jedynie przez „trend” (nie przez blok funkcyjny regulatora), obrazujący zmienne SP, PV i CV. Do

ustawienia jest również czas obserwacji. Uaktywnienie rejestratora pomiarów nastąpi po kliknięciu klawisza *Start*.



Rys. 48 Okno wykresu *Tuning* z przebiegiem zmiennych w układzie regulacji z nastawami wyliczonymi przez SIPART DR24

W obszarze *Online status of controller* wyświetlane są bieżące wartości : *Setpoint*, *Input* oraz *Output* (w trybie *Start Measurement*) z możliwością przełączania trybów *automatic/manual* (jeżeli stan regulatora – *Enable controller*). Działanie w *Manual mode* umożliwia ręczne takie ustawienie wyjścia regulatora, aby przepływ powietrza (wartość procesowa – *Input*) różnił się od wartości zadanej, odpowiednio do wybranego trybu automatycznego doboru nastaw.

Możliwe są dwa tryby automatycznego doboru nastaw (*Autotuning*):

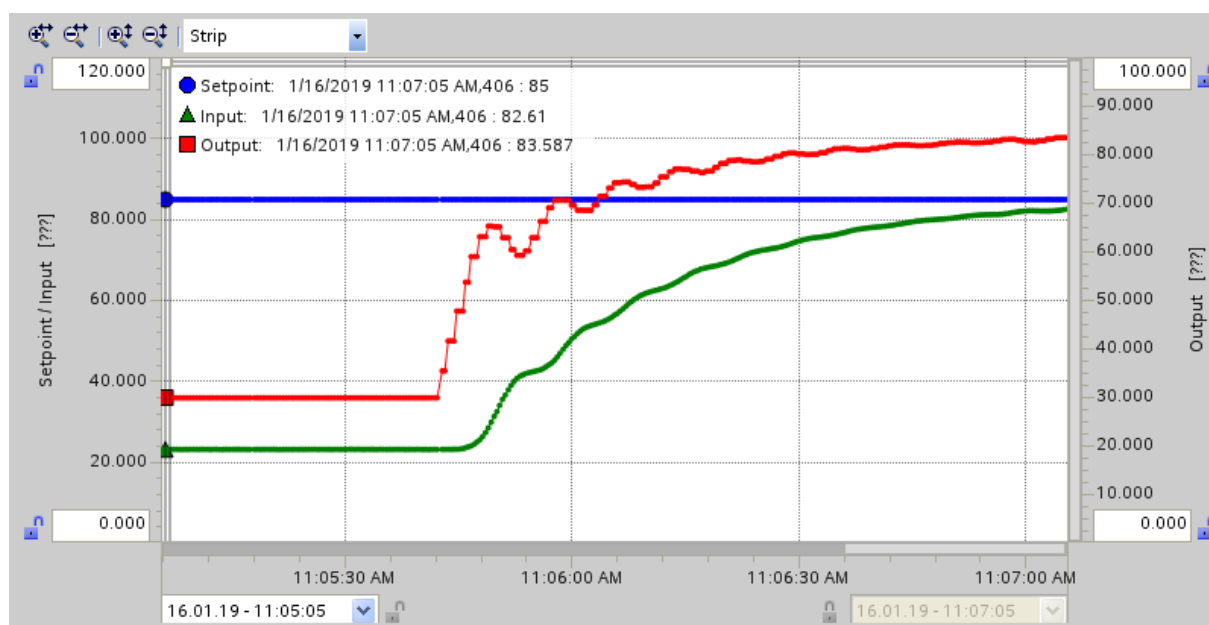
- *Pretuning (strojenie wstępne)*– gdy uchyb (różnica pomiędzy wartością *Setpoint* a *Input*, czyli $SP - PV$) wynosi więcej niż 50% zakresu *Setpoint* oraz jest większy niż 30% różnicy między górnym limitem PV a dolnym limitem PV. Wartość SP powinna być większa od PV.
- *Fine tuning (precyzujące strojenie, dostrojenie)*– gdy uchyb wynosi mniej niż 50% zakresu *Setpoint*.

Lepsze wyniki doboru nastaw będą otrzymane, jeżeli wykona się po pierwszym uruchomieniu regulatora strojenie wstępne – *Pretuning*, a następnie już dostrojenie - *Fine tuning*. Jednak nie zawsze jest możliwość „oddalenia” na obiekcie wartości zadanej od procesowej i tym samym przeprowadzenia strojenia wstępnego.

Na rys. 48 przedstawiono przebieg zmiennych (*Setpoint*, *Input*, *Output*) po pierwszym uruchomieniu układu regulacji (ustawienia wyliczone przez SIPART DR24 – rys. 31b) przed

początkiem działań związanych z doбором nastaw w trybie *Pretuning*. Wstępnie w trybie ręcznym (*Manual mode* w polu *Online status of controller*) ustawiono na wyjściu regulatora sygnał o wartości 30% zakresu i po ustabilizowaniu się przebiegów na wykresie przełączono się w tryb automatyczny (*Enabled-automatic mode*), wystarczy odznaczyć tryb ręczny. Obserwowany proces dojścia do wartości zadanej układu regulacji (również zdolność układu do radzenia z zaistniałym zakłóceniem od strony obciążenia procesu) wskazuje, że ustawienia wyliczone przez SIPART DR24 dają niezłe wyniki w zachowaniu się układu. Na rys. 49 natomiast przedstawiono przebieg zmiennych po pierwszym uruchomieniu układu regulacji z nastawami domyślnymi (rys.31a) ze zmianą parametru *Proportional gain* z 1.0 na 2.0.

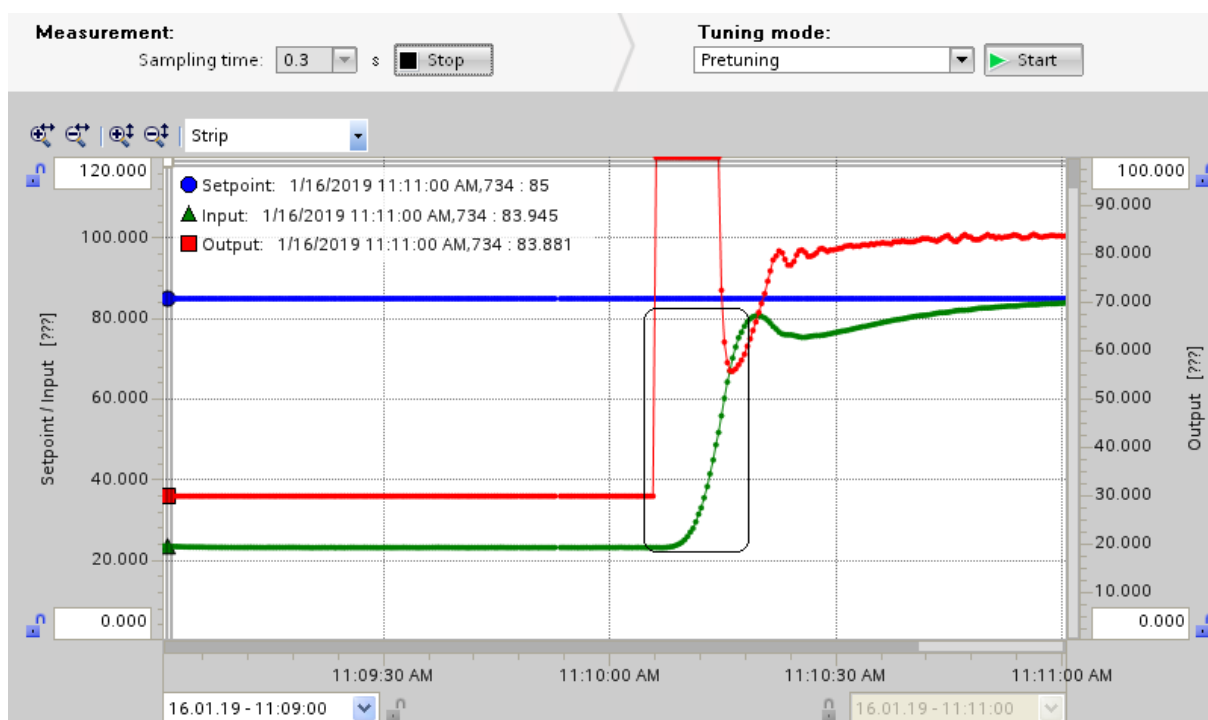
W tym przypadku ewidentnie należy przeprowadzić procedurę doboru nastaw.



Rys. 49. Okno wykresu *Tuning* z przebiegiem zmiennych w układzie regulacji z nastawami domyślnymi (z *Proportional gain* - 2.0).

Po wyborze trybu *Pretuning*, zanim uruchomiony zostanie klawisz *Start*, przestawić regulator w tryb pracy ręcznej (*Manual mode*) i wpisać wartość np. 30.0 w polu *Output* aby doprowadzić do „oddalenia” na obiekcie wartości zadanej od procesowej (pojawi się na wyjściu regulatora po akceptacji wpisu). Może być również wybrany, zamiennie, stan *Disabled-inactive* (przyciskiem *Enable/disable controller*).

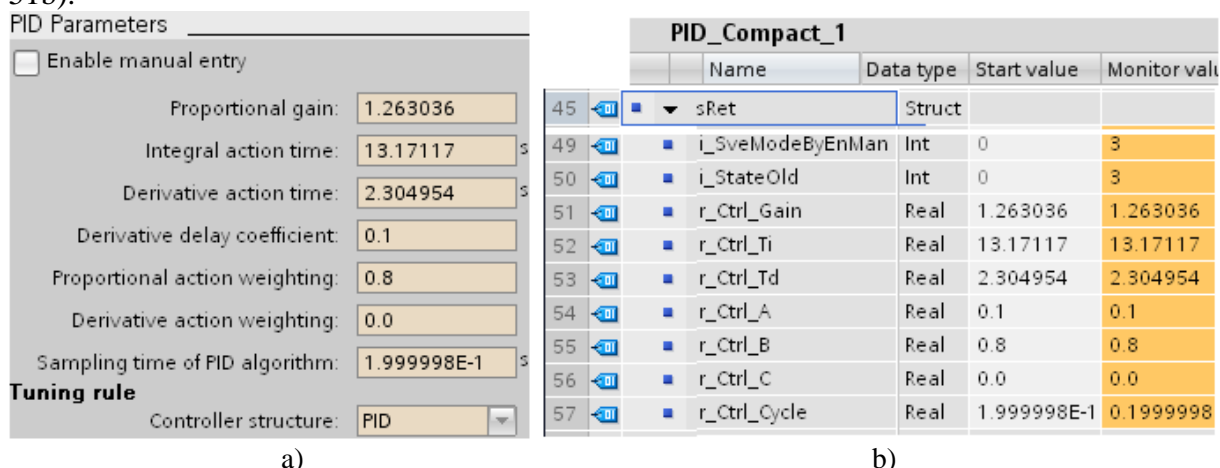
Kliknięciem *Start* w polu *Tuning mode* rozpoczyna się proces doboru nastaw *Pretuning* (rys. 50). W polu *Tuning status* można obserwować stan bieżący procesu doboru nastaw („pasek postępu”) oraz pojawianie się odpowiednich komunikatów: *Pretuning in progress* i po zakończeniu – *System tuned*. W polu *Controller state* podczas procesu doboru nastaw sygnalizowany jest stan *Enabled – pretuning*. Podczas rozpoczętego procesu wstępnego strojenia regulatora badana jest szybkość przebiegu zmiennej *Input* (po określeniu punktu przegięcia na charakterystyce) oraz czas opóźnienia reakcji na skok sygnału *Output* i na podstawie tych danych obliczane są wartości nastaw (*TIA Portal Help/Using technology functions/PID control/PID parameter settings*). Wynik obliczeń jest otrzymywany dosyć szybko i regulator przechodzi z stanu *Enabled – pretuning* w stan *Enabled-automatic mode*.



Rys. 50. Okno wykresu *Tuning* z przebiegiem zmiennych po rozpoczęciu procesu doboru nastaw *Pretuning*

Metoda strojenia bazuje na analizie odpowiedzi obiektu na wymuszenie skokowe, gdy odpowiedź aperiodyczną aproksymuje się za pomocą odcinka osi czasowej (opóźnienie) i półprostej o nachyleniu α (Ziegler J.G, Nichols N.B., Optimum Settings for Automatic Controllers, Transactions of the A.S.M.E, 1942.)

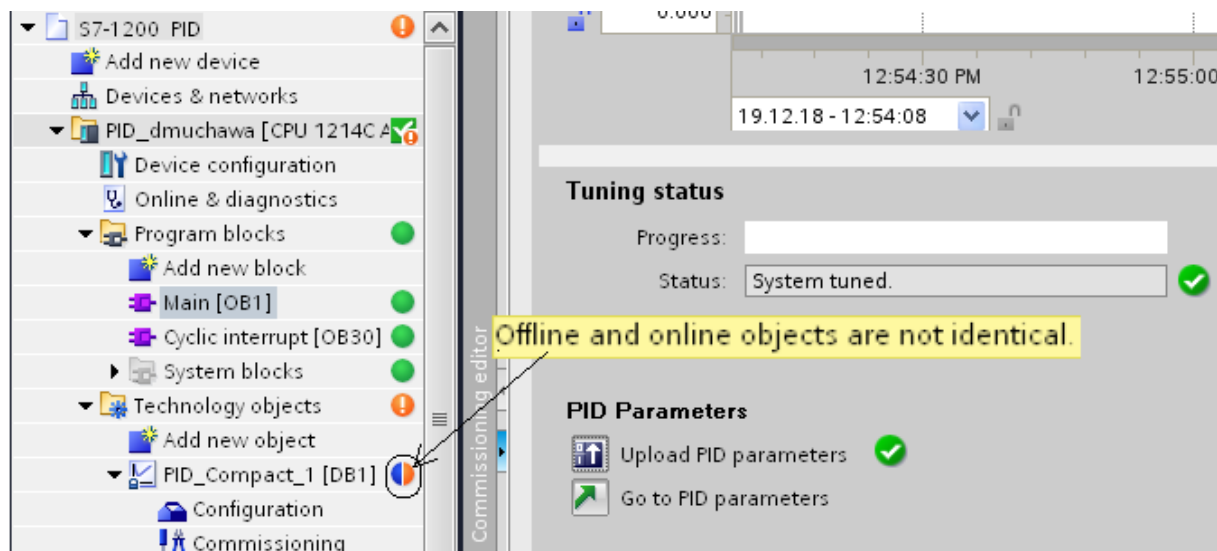
Na rys. 51a pokazano obliczone wartości nastaw w trybie *Pretuning* (po naciśnięciu przycisku *Go to PID parameters* w polu *PID Parameters* okna wykresu *Tuning* nastąpi przejście do odpowiedniej zakładki w oknie konfiguracji obiektu technologicznego *PID_Compact_1 [DB1]*- tryb online) oraz widok *online* okna edytora bloku danych DB1 z nastawami (rys. 51b).



Rys. 51 Nastawy PID : a) parametry obliczone w wyniku doboru nastaw *Pretuning* b) widok *online* części okna edytora DB1 – parametry nastaw *startowe* i *monitorowane*

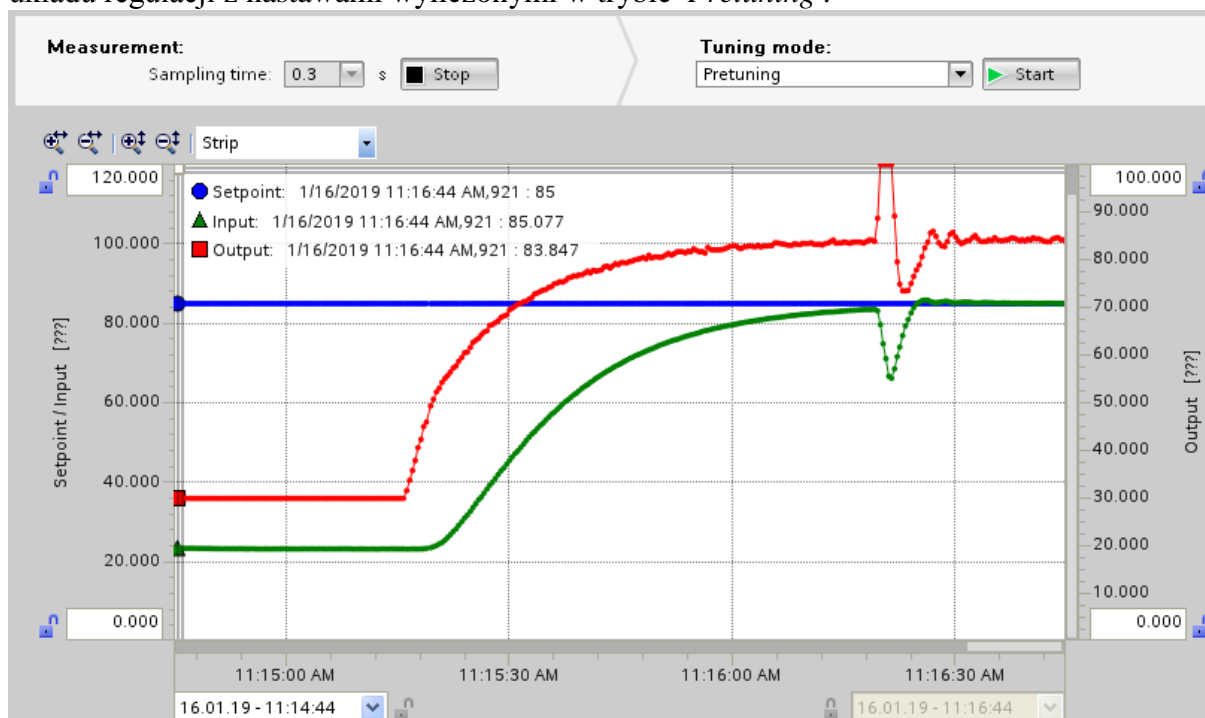
Nowe obliczone nastawy zapisywane są (wyeksportowane z S7-1200) do projektu po naciśnięciu przycisku *Upload PID parameters* w polu *PID Parameters* okna wykresu *Tuning*, przy którym wyświetlany jest wykrzyknik ostrzegawczy, zamieniający się w znak potwierdzenia zapisu (wartości startowe i monitorowane w sterowniku są wtedy jednakowe –

rys. 51b). W trybie *Online* zaobserwować jednak można niezgodność projektu i aktualnej aplikacji w sterowniku (wyświetlany jest komunikat „*offline and online objects are not identical*” oraz przy obiekcie technologicznym DB1 świeci informacja o niezgodności w postaci odpowiedniej ikonki) (rys. 52). Wymagane jest więc w tym momencie przesłanie aktualnego, zmodyfikowanego *Software* do sterownika .



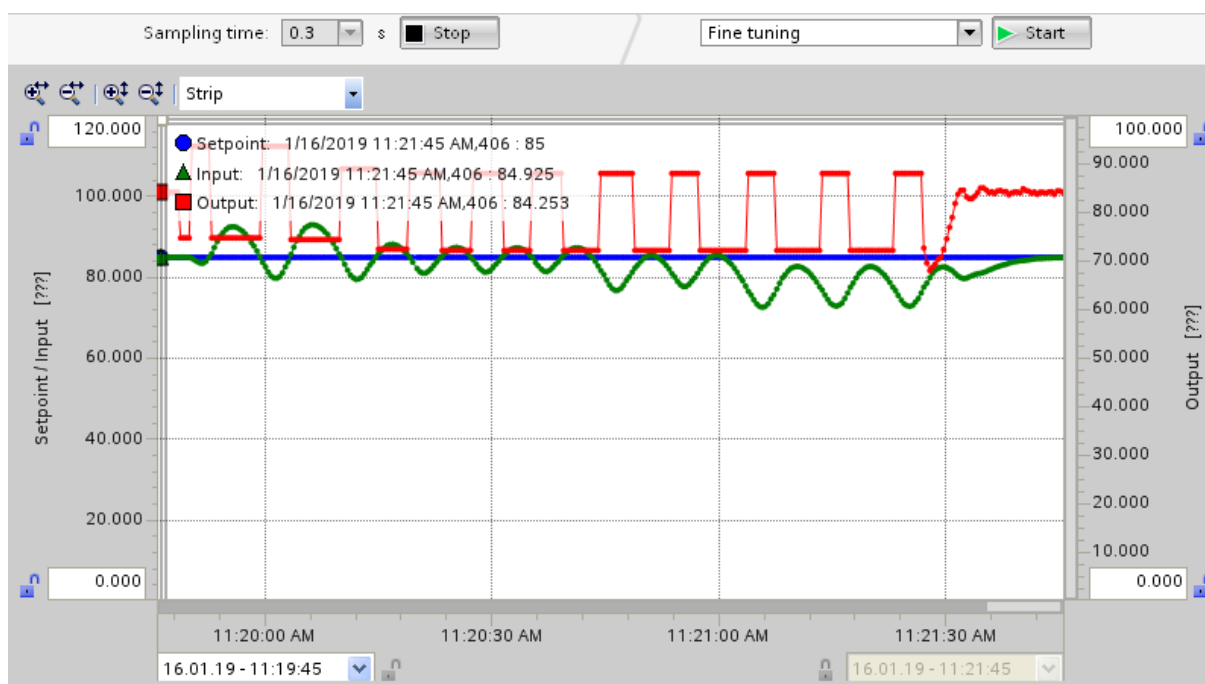
Rys. 52. Nieodpowiedniość projektu i aktualnej aplikacji w sterowniku obserwowana w trybie *online* po zakończeniu procesu *Pretuning*

Na rys. 53 przedstawiono przebieg zmiennych (*Setpoint*, *Input*, *Output*) po uruchomieniu układu regulacji z nastawami wyliczonymi w trybie *Pretuning* .



Rys. 53 Okno wykresu *Tuning* z przebiegiem zmiennych w układzie regulacji z nastawami wyliczonymi w trybie *Pretuning*

Następnie przeprowadzony jest drugi z trybów automatycznego doboru nastaw (*Autotuning*) - *Fine tuning* (*precyzujące strojenie, dostrojenie*). W tym trybie badany jest układ zamknięty regulacji w pobliżu wartości zadanej (punktu pracy) na zakłócenie w postaci ciągu dwupołożeniowych sygnałów. Dobór nastaw regulatora bazuje na zautomatyzowanej metodzie cyklu granicznego Zieglera –Nicholsa (Ziegler J.G, Nichols N.B., Optimum Settings for Automatic Controllers, Transactions of the A.S.M.E, 1942.) zaproponowanej przez Astroma K. i Hagglunda T. (K.J. Åström, T. Häggglund, Automatic tuning of simple regulators with specifications on phase and amplitude margins, Automatica, 20 (1984)). W oknie *Tuning* należy wybrać w polu *Tuning mode* tryb – *Fine tuning*. Układ regulacji jest uruchomiony i regulator (z nastawami obliczonymi w procesie *Pretuning*) stabilizuje zmienną procesową na poziomie wartości zadanej w trybie *Enabled – automatic mode*. Rozpoczęcie procesu doboru nastaw (*Fine tuning*) rozpoczyna się kliknięciem *Start* w polu *Tuning mode* (rys. 54). W polu *Tuning status* można obserwować stan bieżący procesu doboru nastaw („pasek postępu”) oraz pojawienie się odpowiedniego komunikatu: *Fine tuning in progress*. W polu *Controller state* podczas procesu doboru nastaw sygnalizowany jest tryb *Enabled – fine tuning*. Po zakończeniu procesu doboru nastaw ukazuje się komunikat– *System tuned* a regulator przechodzi w stan *Enabled – automatic mode*.



Rys. 54. Okno wykresu *Tuning* z przebiegiem zmiennych po rozpoczęciu procesu doboru nastaw *Fine tuning*

Nowe obliczone nastawy zapisywane są do projektu po przyciśnięciu przycisku *Upload PID params* w polu *PID Parameters* okna wykresu *Tuning*. W trybie *Online* zaobserwować można nieodpowiedniość projektu i aktualnej aplikacji w sterowniku (podobnie jak po zakończeniu procesu *Pretuning* - rys. 52), wymagane jest więc przesłanie do sterownika aktualnego *Software*.

Na rys. 55a pokazano obliczone wartości nastaw w trybie *Fine tuning* (po naciśnięciu przycisku *Go to PID parameters* w polu *PID Parameters* okna wykresu *Tuning* – tryb *online*). Widok okna edytora bloku danych DB1 (przed operacją *Upload PID params* w polu *PID Parameters* okna wykresu *Tuning*) z obliczonymi wartościami nastaw w procesie *Fine tuning* jako *monitorowane* oraz wartościami nastaw po procesie *Pretuning* jako *startowymi* przedstawia rys. 55b.

☐ Enable manual entry

Proportional gain: 1.807881
 Integral action time: 4.28886 s
 Derivative action time: 1.080485 s
 Derivative delay coefficient: 0.1
 Proportional action weighting: 2.662181E-1
 Derivative action weighting: 0.0
 Sampling time of PID algorithm: 9.999924E-2 s

Tuning rule
 Controller structure: PID

PID_Compact_1					
	Name	Data type	Start value	Monitor value	
51	r_Ctrl_Gain	Real	1.263036	1.807881	
52	r_Ctrl_Ti	Real	13.17117	4.28886	
53	r_Ctrl_Td	Real	2.304954	1.080485	
54	r_Ctrl_A	Real	0.1	0.1	
55	r_Ctrl_B	Real	0.8	0.2662181	
56	r_Ctrl_C	Real	0.0	0.0	
57	r_Ctrl_Cycle	Real	1.999998E-1	0.09999924	

Rys. 55 Nastawy PID : a) parametry obliczone w wyniku doboru nastaw *Fine tuning* b) widok *online* części okna edytora DB1 – parametry nastaw *startowe* i *monitorowane*

W trybie *Online* po operacji *Upload PID params* w polu *PID Parameters* okna wykresu *Tuning* zaobserwować można (rys. 56), podobnie jak w trybie *Pretuning*, niezgodność projektu i aktualnej aplikacji w sterowniku (wyświetlany jest komunikat „*offline and online objects are not identical*” oraz przy obiekcie technologicznym DB1 świeci informacja o niezgodności w postaci odpowiedniej ikonki). Wymagane jest więc przesłanie aktualnego, zmodyfikowanego *Software* do sterownika .

<div> <div>Technology objects</div> <div>Add new object</div> <div>PID_Compact_1 [DB1]</div> <div>Configuration</div> <div>Commissioning</div> <div>External source files</div> <div>PLC tags</div> <div>PLC data types</div> <div>Watch and force tables</div> </div>	49	i_SveModeByEnMan	Int	0	3
	50	i_StateOld	Int	0	3
	51	r_Ctrl_Gain	Real	1.807881	1.807881
	52	r_Ctrl_Ti	Real	4.28886	4.28886
	53	r_Ctrl_Td	Real	1.080485	1.080485
	54	r_Ctrl_A	Real	0.1	0.1
	55	r_Ctrl_B	Real	2.662181E-1	0.2662181
	56	r_Ctrl_C	Real	0.0	0.0
	57	r_Ctrl_Cycle	Real	9.999924E-2	0.09999924

Rys. 56. Nieodpowiedniość projektu i aktualnej aplikacji w sterowniku obserwowana w trybie *online* po zakończeniu procesu *Fine tuning*

Na rys. 57 pokazano obliczone wartości nastaw w trybie *Fine tuning* (jako startowe) w zakładce *Advanced settings/PID Parameters* zakładki *Configuration* obiektu technologicznego *PID_Compact_1 [DB1]*- tryb *offline*.

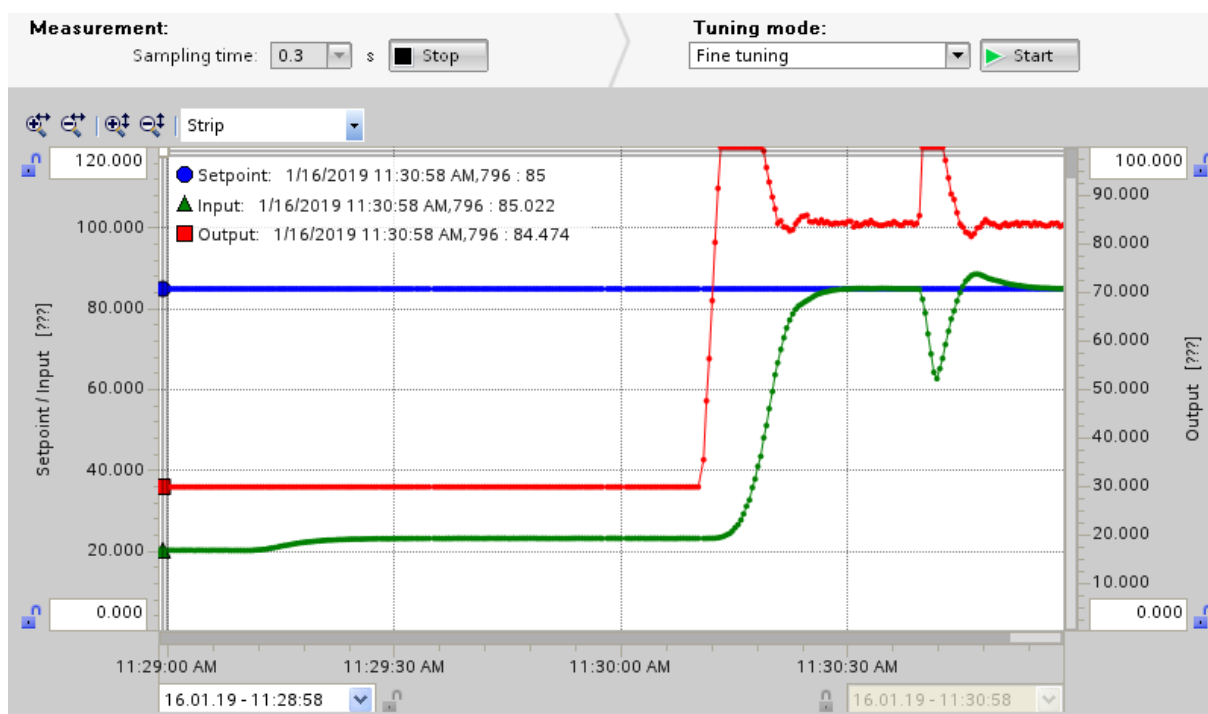
☒ Enable manual entry

Proportional gain: 1.807881
 Integral action time: 4.28886 s
 Derivative action time: 1.080485 s
 Derivative delay coefficient: 0.1
 Proportional action weighting: 2.662181E-1
 Derivative action weighting: 0.0
 Sampling time of PID algorithm: 9.999924E-2 s

Tuning rule
 Controller structure: PID

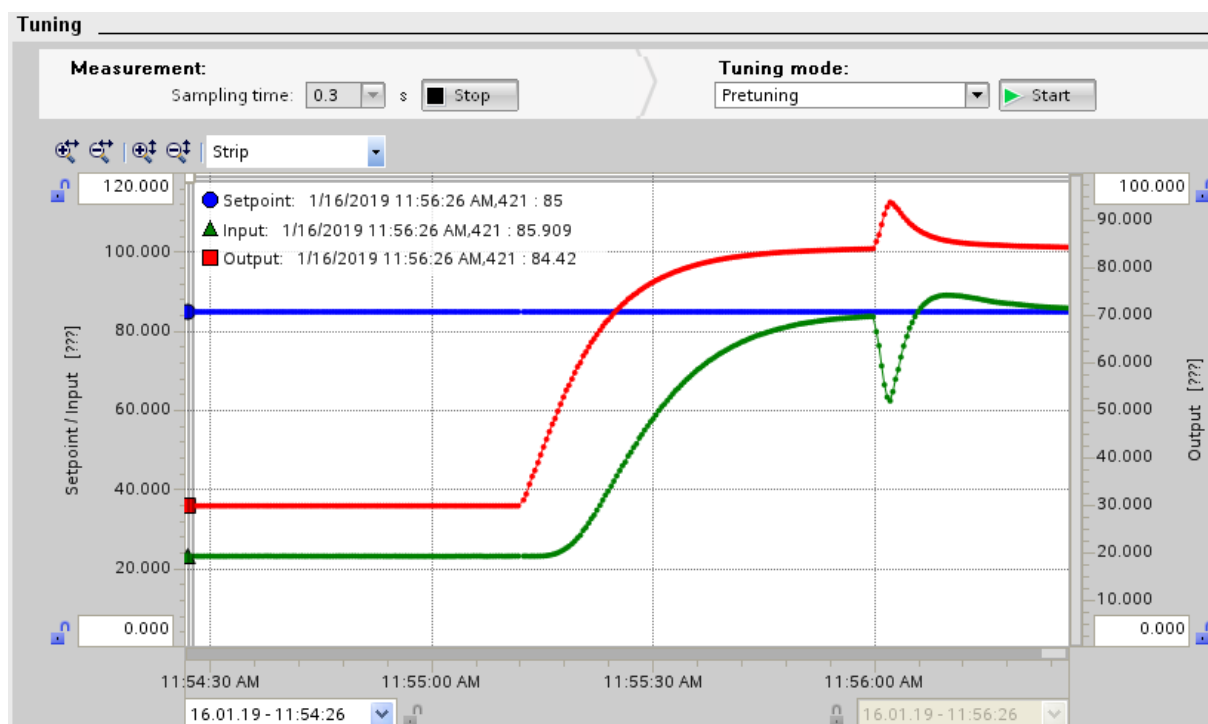
Rys. 57 Nastawy PID obliczone w wyniku doboru nastaw *Fine tuning* jako startowe (tryb *offline*)

Na rys.58 przedstawiono przebieg zmiennych (*Setpoint*, *Input*, *Output*) po uruchomieniu układu regulacji z nastawami wyliczonymi w trybie *Fine tuning*



Rys. 58 Okno wykresu *Tuning* z przebiegiem zmiennych w układzie regulacji z nastawami wyliczonymi w trybie *Fine tuning*

Na rys. 59 przedstawiono przebiegi zmiennych w układzie regulacji przepływu powietrza z nastawami regulatora PI wyliczonymi metodą λ (Vance VanDoren, PhD, PE, *Understanding a particularly conservative PID controller design technique*, Control Engineering 04/16/2013) : $K_p = 0.362$, $T_i = 5.4$ s



Rys. 59 Okno wykresu *Tuning* z przebiegiem zmiennych w układzie regulacji z nastawami wyliczonymi metodą λ

Porównując przykładowe przebiegi zmiennych na rys.48 (nastawy wyliczone przez jednostkę SIPART DR24) , rys.58 (nastawy wyliczone przez sterownik S7-1200 z wykorzystaniem

trybów *Pretuning* a następnie *Fine tuning*) oraz rys.59 (nastawy wyliczone metodą λ) można zauważyć (bez uogólnienia), że najbardziej korzystnym wygląda przebieg w układzie z nastawami wyliczonymi w trybie *Fine tuning* (S7-1200) - brak przeregulowań a dojście do wartości ustalonej (czas regulacji) jest najszybsze.

Podstawy doboru nastaw metodą λ

Ideą stosowania metody jest dojście procesu do nowej wartości zadanej w określonym czasie i bez jej przekroczenia. Takie podejście jest szczególnie cenne, gdy istnieją krytyczne granice, których proces nie powinien przekroczyć. Nie jest ona odpowiednia dla każdej aplikacji, szczególnie tych, które wymagają szybkiej reakcji.

Danymi do obliczeń są: wzmocnienie procesu (obiektu) K_o , stała czasowa procesu T_p i czasu opóźnienia D (otrzymane w układzie otwartym na podstawie przebiegu wyjścia obiektu na skok sygnału wartości zadanej, odpowiedź skokowa).

Wyjście sygnału $CV(t)$ z regulatora PI:

$$CV(t) = K_c \left[e(t) + \frac{1}{T_i} \int e(t) dt \right] \quad (2)$$

gdzie: $e(t) = SP(t) - PV(t)$ oznacza błąd między zmienną procesową $PV(t)$ a wartością zadaną $SP(t)$.

Zasady dostrajania λ wymagają wyliczenia

$$K_c = \frac{T_p}{K_o(\lambda + D)} \quad \text{oraz} \quad T_i = T_p \quad (3)$$

gdzie: $K_o = \Delta X / \Delta Y$, T_p i D określić można z wykresu

Tak dostrojony regulator PI teoretycznie doprowadzi stopniowo zmienną procesową do zmienionej skokowo wartości zadanej w ciągu około 4λ sekund podczas pracy w trybie zamkniętej pętli i zrobi to bez jej przekroczenia. Funkcja nadmiernego tłumienia może być szczególnie użyteczna w aplikacjach, w których zmienna procesowa musi być utrzymywana w pobliżu pewnej wartości granicznej, której zmienna procesowa nie może przekroczyć. Regulator nigdy przypadkowo nie naruszy takiego ograniczenia, ponieważ nigdy nie doprowadzi zmiennej procesowej do wartości zadanej. Regulator dostrojony metodą λ nigdy nie spowoduje niestabilnych oscylacji zmiennej procesowej, ponieważ nigdy nie będzie musiał odwracać kursu po zmianie wartości zadanej. Zmienna procesowa zawsze będzie postępować stopniowo w górę lub stopniowo w dół, aż do osiągnięcia nowej wartości zadanej.

Z drugiej strony, strojenie λ ma swoje granice, szczególnie gdy liczy się prędkość (czas regulacji). Powoduje to, że powolny proces staje się jeszcze wolniejszy, zmienna procesowa pozostaje poza specyfikacją (standardowym odchyleniem PV od SP) przez długi czas. Przy obliczeniu parametru K_c dla λ jest zazwyczaj przypisywana wartość pomiędzy T_p a $3T_p$, co powoduje, że odpowiedź w pętli zamkniętej na zmianę wartości zadanej jest trzy razy dłuższa niż odpowiadająca jej odpowiedź skokowa w otwartej pętli. Jeszcze większa wartość λ jest wymagana, jeśli czas opóźnienia D jest znaczący. W takich przypadkach $\lambda > D$ jest praktyczną dolną granicą, ponieważ nie można oczekiwać, że regulator zareaguje szybciej niż pozwala na to czas opóźnienia.

Jednak prawdopodobnie największą wadą regulatora z nastawami obliczonymi metodą λ jest jego ograniczona zdolność do radzenia sobie z zewnętrznym obciążeniem procesu. Nadal może przywrócić zmienną procesową z powrotem do wartości zadanej, jeśli przypadkowe obciążenie zakłóci zmienną procesową, ale nie podejmie żadnych wysiłków, aby zrobić to szczególnie szybko lub skutecznie. Nawet pomiar zakłóceń nie pomoże, ponieważ reguły strojenia λ nie przewidują odpowiedniego zachowania się w stosunku

do zmian obciążenia, jedynie tylko do samego procesu.

Najlepsze, co użytkownik może zrobić, to ustawić możliwie najmniejszą wartość λ , aby zwiększyć ogólną prędkość działania regulatora, ale spowoduje to że będzie on mniej niezawodny. Strojenie metodą *lambda* i tak nie byłoby szczególnie dobrym wyborem, gdy wymagana jest szybka reakcja, ponieważ istnieją inne reguły strojenia, które są znacznie bardziej skuteczne w aplikacjach wrażliwych na czas.

Istnieją dodatkowe ograniczenia strojenia *lambda*:

- nie można go zastosować do procesu, który sam jest oscylacyjny oraz gdy czas opóźnienia D jest szczególnie duży. Obliczenia wymagane do obliczenia K_c i T_i podlegają przybliżeniu, które staje się coraz mniej dokładne wraz ze wzrostem D . Zaproponowano kilka alternatywnych podejść w celu poprawy dokładności tego przybliżenia, ale wysiłki te spowodowały również znaczne zamieszanie - wiele zestawów zasad strojenia zwanych „strojeniem *lambda*”. Wszystkie osiągają w przybliżeniu tę samą wydajność w pętli zamkniętej, ale nie wyglądają podobnie. Niektóre dotyczą wyłącznie regulatora PI, podczas gdy inne wymagają pełnego regulatora PID. Niemniej jednak strojenie metodą *lambda* jest stosunkowo proste i intuicyjne, pozostanie popularne w aplikacjach, w których wymagany jest konserwatywny regulator.

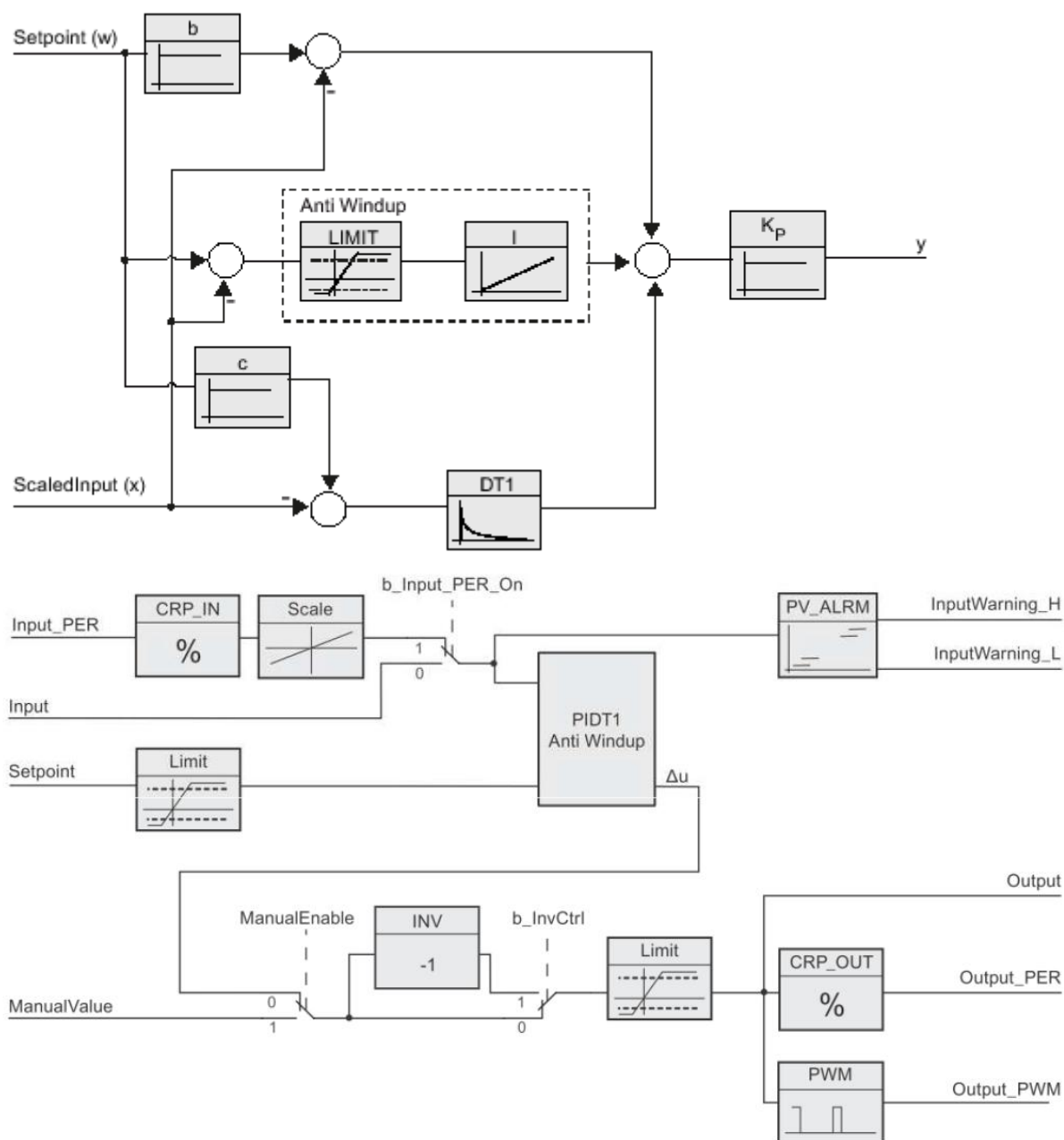
3.6 Literatura

DODATEK [„Help –s7-1200”]

Transmitancja regulatora PID o wyjściu ciągłym w S7-1200

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

Symbol	Description
y	Output value
K _p	Proportional gain
s	Laplace operator
b	Proportional action weighting
w	Setpoint
x	Process value
T _I	Integral action time
a	Derivative delay coefficient (T1 = a × T _D)
T _D	Derivative action time
c	Derivative action weighting



Dobór nastaw regulatora [xx]

Zwyczajowo przypisuje się pierwszeństwo w tej dziedzinie pracy: John G. Ziegler, Nathaniel B. Nichols – *Optimum Settings for Automatic Controller*, Transactions of the A.S.M.E, 1942. Podano tam dwie metody: cyklu granicznego (badanie w układzie zamkniętym) oraz bazującą na odpowiedzi obiektu na wymuszenie skokowe.

Metoda cyklu granicznego („granicy stabilności”) polega na wyłączeniu w układzie zamkniętym działania całkującego i różniczkującego oraz takim ustawieniu współczynnika wzmocnienia, przy którym w układzie powstają oscylacje o stałej amplitudzie (na tej metodzie, ale zautomatyzowanej, bazuje tryb *Tuning* doboru nastaw regulatora PID w S7-1200). Wzmocnienie to nazywa się wzmocnieniem krytycznym S_u (*ultimate sensitivity*).

Drugim parametrem jest okres oscylacji P_u . W tab.1 podano nastawy spełniające warunek:

współczynnik tłumienia oscylacji (stosunek amplitud drugiej oscylacji i pierwszej oscylacji) w odpowiedzi układu zamkniętego na zakłócenie skokowe (*decay ratio*) jest równy 0,25.

Tab.1 Nastawy określone metodą cyklu granicznego

	P	PI	PID
k_p (<i>sensitivity</i>)	$0,5 S_u$	$0,45 S_u$	$0,6 S_u$
k_i (<i>reset rate</i>)		$1,2/P_u$ [min^{-1}]	$2/P_u$ [min^{-1}]
k_d (<i>pre-act time</i>)			$P_u/8$ [min]

Metoda ta doczekała się też innych modyfikacji, np. Pessen założył jako kryterium aperiodyczność przebiegu odpowiedzi na wymuszenie skokowe (a nie minimum czasu regulacji) i zaproponował wyznaczanie nastaw z zależności podanych w tab.2 ($k_{kr} = S_u$, $T_{kr} = P_u$).

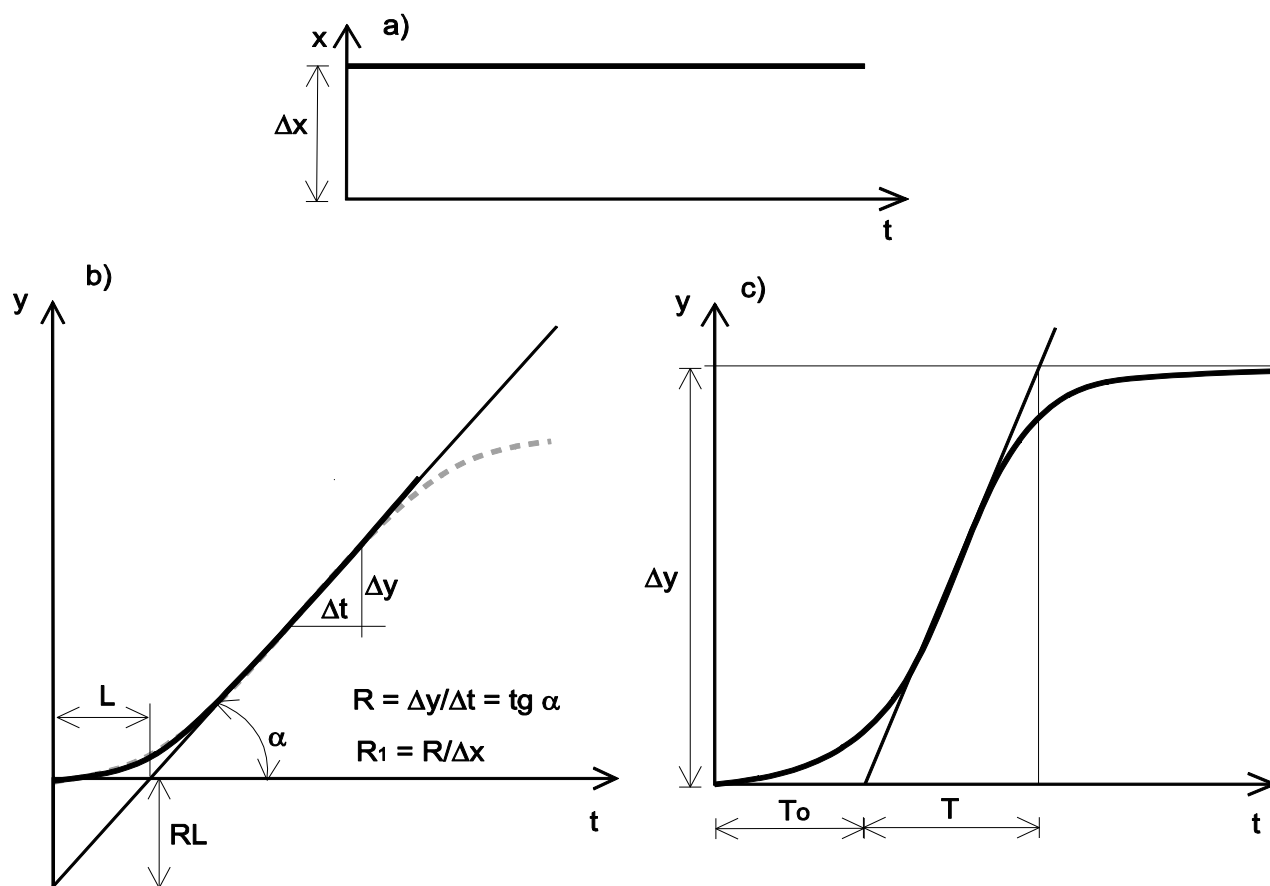
Tab.2. Nastawy regulatora zgodnie z modyfikacją Pessena

	Kp	Ti	Td
P	$0,2k_{kr}$		
PI	$0,15k_{kr}$	$0,4T_{kr}$	
PID	$0,2k_{kr}$	$0,33T_{kr}$	$0,5T_{kr}$

Druga metoda Zieglera-Nicholsa polega na analizie odpowiedzi obiektu na wymuszenie skokowe. Odpowiedź aperiodyczną obiektu aproksymowano za pomocą odcinka osi czasowej (opóźnienie L) i półprostej o nachyleniu α (rys. 1b). Odpowiada to przyjęciu zastępczej transmitancji obiektu astatycznego w postaci:

$$G(s) = \frac{k}{s} e^{-sL}, \quad (1)$$

Określenie wartości parametrów modelu o wybranej strukturze jest zadaniem identyfikacji parametrycznej (identyfikacja „szarej skrzynki”).



Rys. 1 Wyznaczanie parametrów modeli zastępczych wykorzystywanych do doboru nastaw regulatora metodą analizy odpowiedzi skokowej obiektu: a) skokowy sygnał wymuszający, b) obiekt astatyczny, c) obiekt z samowyrównaniem.

Tab.3. Nastawy określone metodą analizy odpowiedzi obiektu na wymuszenie skokowe („druga metoda Zieglera-Nicholsa”)

	P	PI	PID
k_p (sensitivity)	$1/R_1 L$	$0,9 / R_1 L$	$(1,2 - 2) / R_1 L$
k_i (reset rate)		$0,3 / L \text{ [min}^{-1}\text{]}$	$0,5 / L \text{ [min}^{-1}\text{]}$
k_d (pre-act time)			$0,5 L \text{ [min]}$

Metodzie tej zarzuca się nieuwzględnianie w sposób jawny wzmocnienia statycznego obiektu oraz trudności w ustaleniu przebiegu stycznej w punkcie przegięcia.

Parametry wykorzystywane do wyznaczania nastaw pierwszą i drugą metodą Zieglera – Nicholasa są powiązane heurystycznymi zależnościami: $L = P_u / 4$ oraz $R_1 = 8 / (P_u S_u)$.

Obecnie spotyka się wiele modyfikacji, które polegają na wprowadzeniu modelu zastępczego (PT_1 z opóźnieniem) w postaci:

$$G(s) = \frac{ke^{-T_0s}}{Ts + 1}, \quad (2)$$

gdzie k – wzmacnienie obiektu, T_o - czas opóźnienia (zastępczy), T - stała czasowa członu inercyjnego (zastępcza) oraz przyjmowaniu różnych kryteriów optymalności nastaw. Autorzy cytujący zmodyfikowane metody często nie informują, że nie jest to oryginalna druga metoda Zieglera - Nicholasa. Wzorując się na tej metodzie opracowano np. metodę QDR (Quarter – Decay Ratio). Parametry modelu można wyznaczać między innymi w sposób zaproponowany przez K. Küpfmüllera w 1928 roku. Czas opóźnienia T_o i stałą czasową członu inercyjnego T wyznacza się graficznie prowadząc styczną do zdjętej charakterystyki obiektu w punkcie przegięcia. Punkty przecięcia stycznej z osią $0t$ oraz z rzędną poprowadzoną na poziomie sygnału wyjściowego obiektu w stanie ustalonym wyznaczają czasy T_o i T , jak zaznaczono na rys.1c. Zależności umożliwiające wyznaczanie nastaw metodą QDR pokazano tab.4.

Tab.4. Nastawy określone metodą QDR

Typ regulatora	K_p	T_i	T_d
P	$\frac{T}{kT_o}$		
PI	$\frac{0,9T}{kT_o}$	$3,33T_o$	
PID	$\frac{1,2T}{kT_o}$	$2T_o$	$0,5T_o$

We wzorach pojawia się wzmacnienie obiektu, które nie występuje w sposób jawny w metodzie oryginalnej Zieglera-Nicholasa. Nie znaczy to, że nie było tam ono uwzględniane. Rozważmy nastawę wzmacnienia regulatora P:

$$\frac{1}{R_1L} = \frac{\Delta x}{RL} \cong \frac{\Delta x}{\frac{\Delta y}{T}T_o} = \frac{T}{\frac{\Delta y}{\Delta x}T_o} = \frac{T}{kT_o}. \quad (3)$$

Jak widać wynik uzyskany w oryginalnej metodzie Z-N jest bliski wynikowi uzyskanemu metodą QDR.

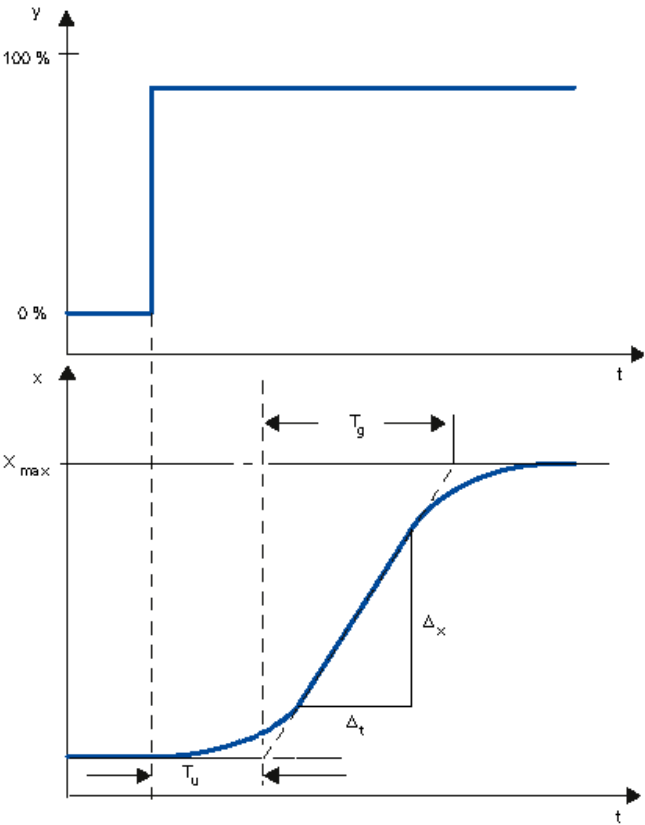
Dobór nastaw w trybie *Pretuning* regulatora PID w S7-1200 (podane w *HELP*):

PID parameter settings

Rule of Thumb for the Parameter Setting

Controller structure	Setting
P	$GAIN \approx v_{max} \times T_u [^{\circ}C]$
PI	$GAIN \approx 1.2 \times v_{max} \times T_u [^{\circ}C]$
PD	$GAIN \approx 0.83 \times v_{max} \times T_u [^{\circ}C]$ $TD \approx 0.25 \times v_{max} \times T_u [min]$ $TM_LAG \approx 0.5 \times TD [min]$
PID	$GAIN \approx 0.83 \times v_{max} \times T_u [^{\circ}C]$ $TI \approx 2 \times T_u [min]$ $TD \approx 0.4 \times T_u [min]$ $TM_LAG \approx 0.5 \times TD [min]$
PD/PID	$GAIN \approx 0.4 \times v_{max} \times T_u [^{\circ}C]$ $TI \approx 2 \times T_u [min]$ $TD \approx 0.4 \times T_u [min]$ $TM_LAG \approx 0.5 \times TD [min]$

Instead of $v_{max} = \Delta_x / \Delta_t$, you can use X_{max} / T_g .



Rys.2