

Programowanie cyfrowe w praktyce

Spis treści

1.	Wstęp	3
2.	Sterownik PLC	4
2.1	Budowa sterownika PLC	4
2.2	Panele Operatorskie	19
3.	Programowanie sterowników PLC	21
3.1	Programowanie w języku drabinkowym LD	23
3.2	Rozszerzenia wierszy w języku drabinkowym LD	26
Wiersze w języku drabinkowym są rozwijane o liczne bloki dodatkowe wymienione w tabelach 3.4, 3.5 i 3.6		26
4.	Programowanie sterownika Siemens LOGO	28
4.1	Ustanawianie połączenia Ethernet	28
4.2	Programowanie LD sterownika w Logo!Soft Comfort v8 oraz tworzenie funkcji użytkownika - UDF	31
4.3	Timery w LOGO!	42
4.4	Przykłady zastosowania podstawowych bloków funkcyjnych sterownika Siemens LOGO!	46
5.	Programowanie sterownika GE-FANUC serii 90-30	60
5.1	Moduły sterownika GE-Fanuc 90-30	60
5.2	Timery GE-Fanuc 90 - 30	70
5.3	Liczniki sterownika GE-Fanuc 90-30	71
5.4	Przesyłanie danych Move na sterowniku GE Fanuc 90-30	73
5.5	Konwertery sterownika GE Fanuc 90-30	73
5.6	Operacje logiczne na słowach sterownika GE Fanuc 90-30	76
5.7	Podstawowe funkcje matematyczne na sterowniku GE Fanuc 90-30	77
5.8	Rozszerzone funkcje matematyczne –GE Fanuc	80
5.9	Regulacja PID na sterowniku GE-Fanuc 90-30	85
5.10	Nastawy regulatora PID:	90
5.11	Strojenie regulatora PID	93
6.	Programowanie sterownika Siemens SIMATIC S7-1200	96
6.1	Tworzenie projektu w programie TIA PORTAL V15	97
6.2	Dodawanie panelu operatorskiego do projektu w TIA PORTAL v15	105
6.3	Przygotowanie panelu operatorskiego do pracy	108
6.4	Programowanie sterownika Siemens S7-1200 w TIA PORTAL v15	112
6.5	Porty HSC na sterownikach	118
6.6	Programowanie portu HSC na sterowniku S7-1200 - tworzenie projektu w programie TIA PORTAL v.15	120
6.7	Regulacja PID na sterowniku Siemens SIMATIC S7-1200	128
6.8	Programowanie i parametryzacja regulatora PID na sterowniku Siemens SIMATIC S7-1200	130

1. Wstęp

Automatyka powstawała w oparciu o układy analogowe. Powstał uniwersalny regulator PID, który znalazł zastosowanie do wielu procesów technicznych. Regulatory PID przetwarza sygnał sterowania w sposób ciągły i z określoną jakością regulacji. Równolegle wymaga się sterowania sygnałami przełączającymi. Załączenie układu wymaga załączenia silników, grzałek, wentylatorów, pomp, zaworów co wymaga sterownia logicznego i sekwencyjnego. Ilość sygnałów odbieranych i sterujących wzrasta, a cały proces sterowania komplikuje się. Stopień złożoności układu można prosto zrealizować w technice cyfrowej. Sygnały przełączające w tej wersji mają naturalną wartość bitową. Przetwarzanie sygnałów analogowych należy dostosować do techniki cyfrowej.

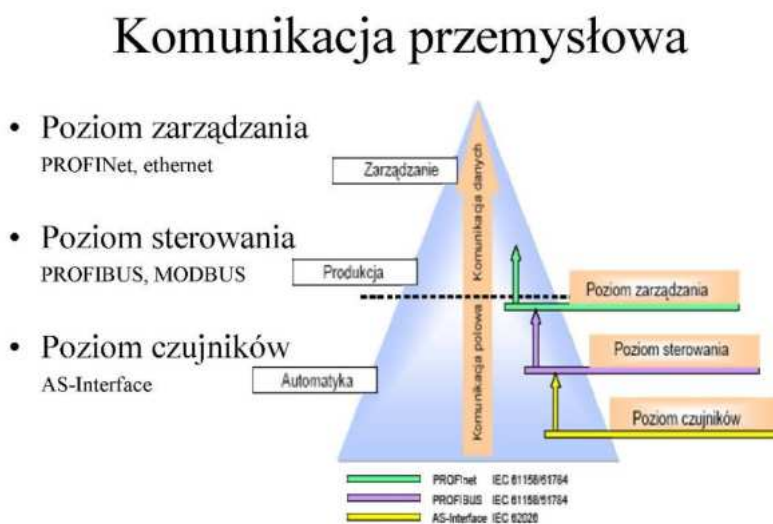
Sterowanie cyfrowe jest szeroko stosowane i stało się podstawą obecnych układów automatyki. Do realizacji układów cyfrowych stosuje się sterowniki PLC (ang. Programmable Logic Controller). Sterownik przetwarza sygnały binarne (dwupołożeniowe, przełączające) i analogowe zmieniając do wartości cyfrowej. Odminną formą sygnału jest sygnał impulsowy szybkodzienny.

Różne aspekty stosowania sterowników i przetwarzania sygnałów oraz programowania są przedmiotem opracowania.

2. Sterownik PLC

W wersji analogowej automatyka rozwijała się w oparciu o układy mechaniczne, pneumatyczne, hydrauliczne i elektryczne. Przetwarzanie kilku sygnałów wymagało dużego rozwinięcia układu technicznego sterującego. Rozwój układów cyfrowych doprowadził do prostego sterowania systemów wielosygnałowych z minimalizacją centralnego układu sterowania. Znane są przykłady sterowania mechaniczne i pneumatyczne do realizacji funkcji logicznej, ale ich minimalizacja fizyczna jest ograniczona i nie rozwinęły się. Tylko sterowanie logiczne w oparciu o technikę cyfrową dało pełną minimalizację fizycznego układu i jest szeroko stosowane w technice.

Komputery osobiste są najlepszym przykładem rozwinięcia techniki cyfrowej, ale nie są przystosowane do sterowania. Układem cyfrowym o przeznaczeniu sterującym jest sterownik, który rejestruje informacje bezpośrednio z obiektu, przetwarza sygnały według zapisanego algorytmu, a następnie ustawia wyjścia sterujące sterownika połączone z obiektem. Sterownik jest ważnym elementem w procesie sterowania i źródłem informacji w procesie zarządzania. Pozycja sterowania na grafie to 'poziom sterownia', Rys. 2.1.



Rys. 2.1 Pozycja sterownika w procesie sterowania

2.1 Budowa sterownika PLC

Z uwagi na konstrukcję sterowników dzielimy je następująco:

- sterowniki kompaktowe
- sterowniki kompaktowe z możliwością rozbudowy
- sterowniki modułowe małe, średnie i duże

Sterowniki kompaktowe to małe układy sterowania w zwartej obudowie, małej liczbie wejść i wyjść. W małej obudowie występują wszystkie elementy składowe sterownika, tzw. nano i mikro sterowniki. Takie układy obsługują pojedyncze obwody sterowania, zastępują wcześniej stosowane regulatory. Najprostszy sterownik kompaktowy to układ LOGO firmy Siemens do układów przełączających, Rys. 2.2.

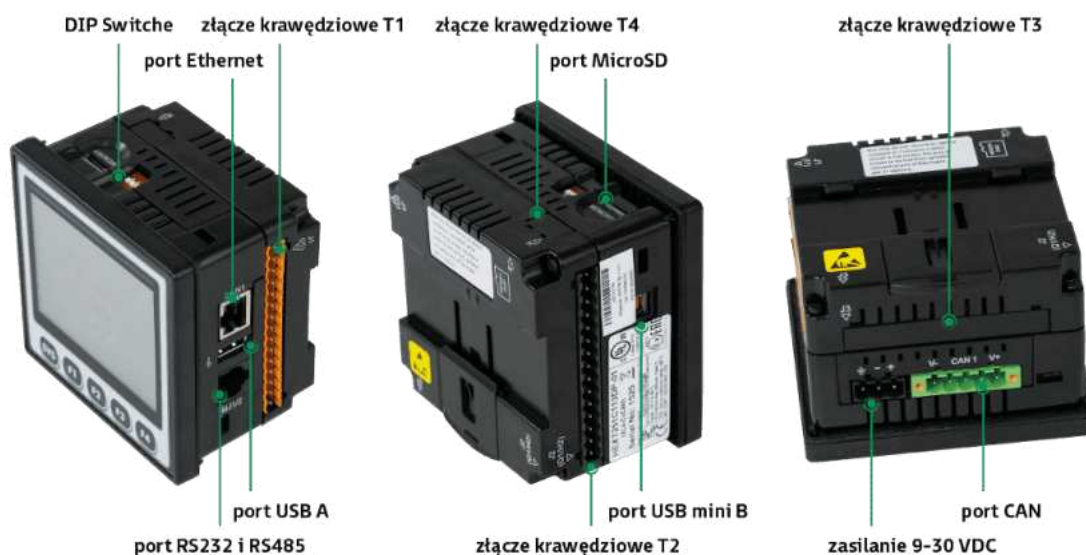


Rys. 2.2 Sterownik kompaktowy LOGO firmy Siemens

Producenci elastycznie dopasowują się do potrzeb rynku i wprowadzają sterowniki z możliwością dołączenia zewnętrznych modułów w ograniczonej ilości, Rys. 2.3 i Rys. 2.4.

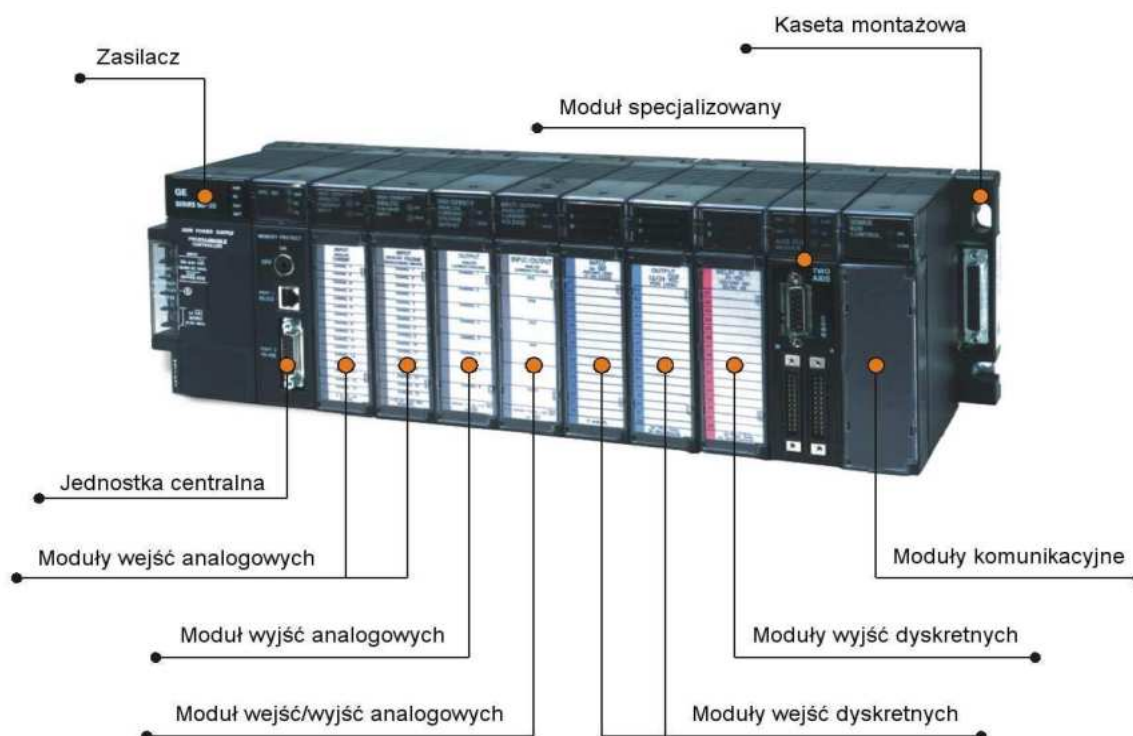


Rys. 2.3 Sterownik VersaMax firmy GE-Fanuc



Rys. 2.4 Sterownik firmy Horner serii X14e

Sterowniki modułowe cechują się strukturą segmentową. Oddzielnie występuje zasilacz, CPU i moduły wejść/wyjść. Ilość dołączonych modułów decyduje o wielkości sterownika. W tych rozwiązaniach dobiera się potrzebne moduły do projektu, ogranicza to koszty układu i stosuje się w większych projektach, rys. Rys. 2.5, Rys. 2.6 i Rys. 2.7.



Rys. 2.5 Sterownik modułowy firmy GE-Fanuc serii 90-30



Rys. 2.6 Sterownik Mitsubishi typu FX3U o zasilaniu AC 100-240 V; zintegrowane wejście 16 DC 24 V i zintegrowane wyjście tranzystorowe 16 DC oraz moduły zewnętrzne



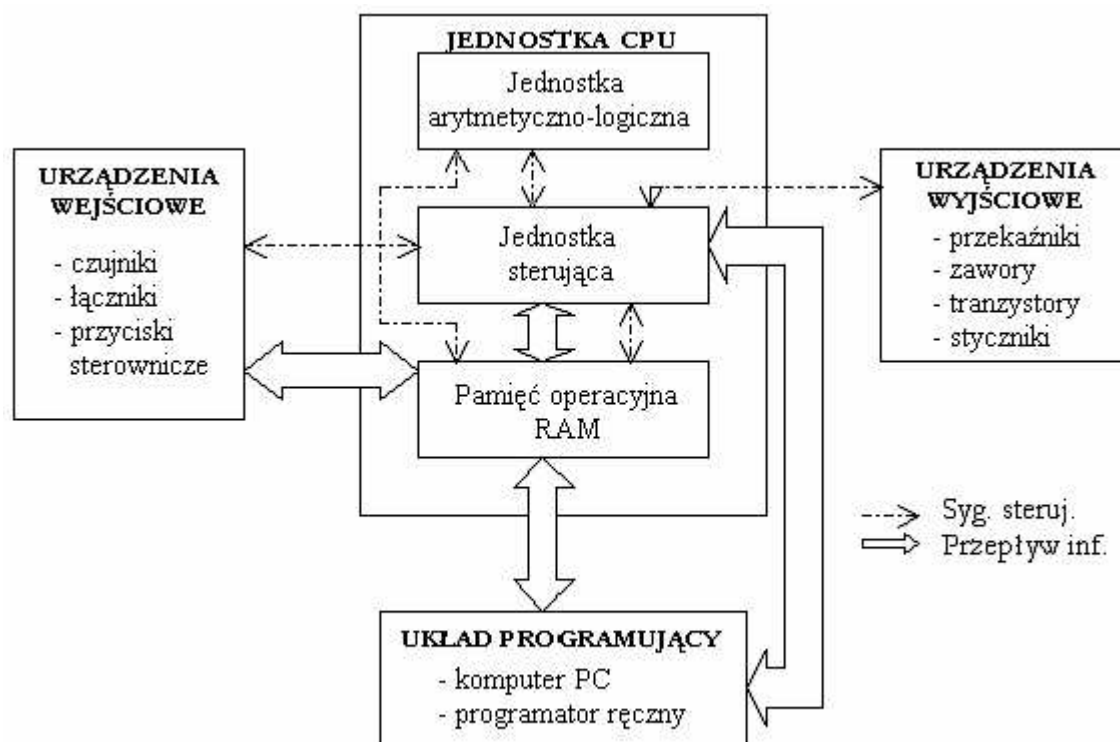
Rys. 2.7 Sterownik S7-1200 firmy Siemens zintegrowany oraz z modułami zewnętrznymi

Centralnym elementem każdego układu cyfrowego jest procesor, który przetwarza dane cyfrowo. Program sterowania i dane procesowe są zapisywane w pamięci operacyjnej. Przykładową strukturę sterownika przedstawia Rys. 2.8

Sterownik zbudowany jest z następujących komponentów:

- modułu zasilania,
- jednostki centralnej CPU (ang. Central Processor Unit) ,
- pamięci operacyjnej i stałej,
- modułu wejść cyfrowych,
- modułu wejść analogowych,
- modułu wyjść cyfrowych,

- modułu wyjść analogowych,
- modułu komunikacyjnego.



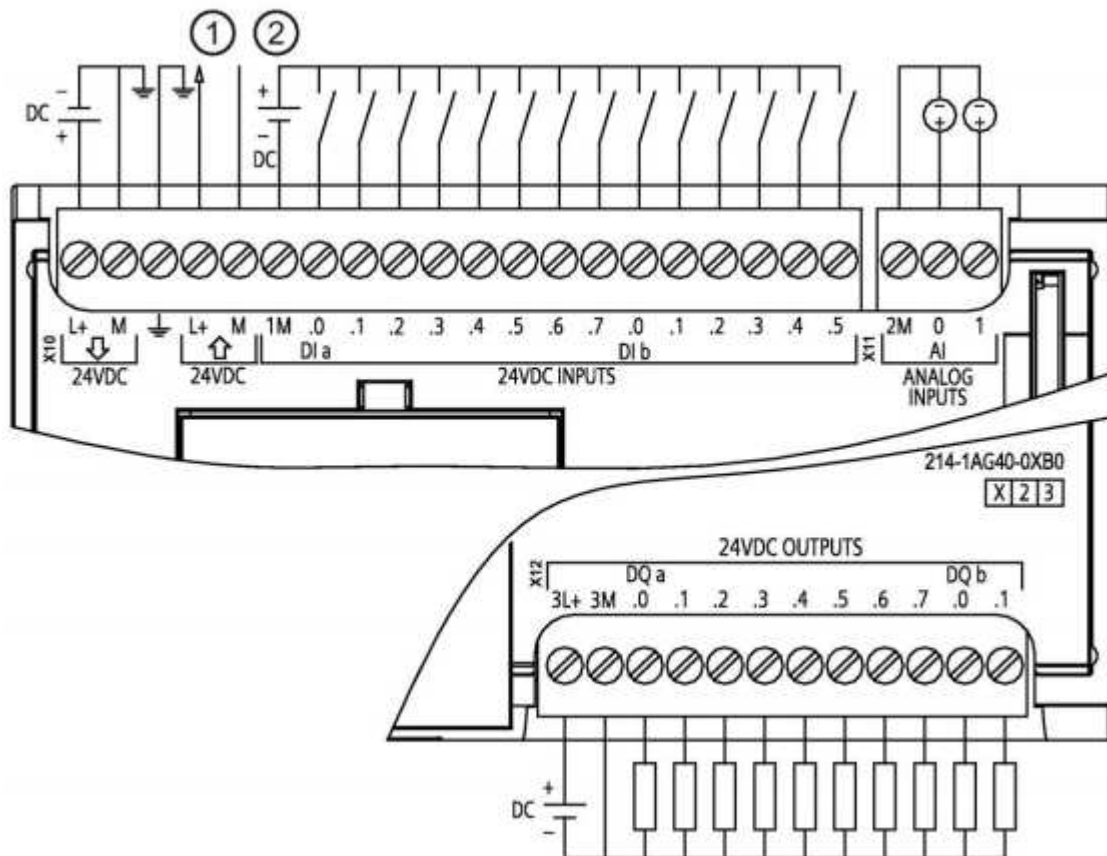
Rys. 2.8 Budowa sterowników PLC

Zasilacz sterownika jest często modułem zewnętrznym sterownika odpowiedzialnym za jego zasilanie oraz wszystkie moduły rozszerzające. Może pełnić funkcje dodatkowe na przykład komunikację przez port szeregowy. W prostych układach dodatkowo zasilacz może być źródłem energii dla obwodów wejściowych i wyjściowych. W tym wypadku jego moc jest znacznie większa od potrzeb samego sterownika.

Jednostka centralna CPU odpowiada za realizację programu sterowania. Wymaga dobranej mocy obliczeniowej. Stosowane procesory są proste, powinny wykonywać operacje matematyczne zmiennoprzecinkowe. Podstawowym działaniem jest arytmetyka stałopozycyjna 16-to bitowa. Jednostki centralne firmy Siemens serii S7-300/400 mają oznaczenie CPU312/412 i wyżej, S7-1200 jako CPU 1211 i wyżej, natomiast firma GE Fanuc stosuje oznaczenia IC693CPU311 i wyżej dla serii 90-30. Integralną częścią układu cyfrowego jest pamięć operacyjna do zapisu danych oraz pamięć stała do przechowywania programu EPROM.

Wszystkie wymienione elementy są podstawą każdego układu cyfrowego. Sterownik nie ma **pamięci masowej, klawiatury, karty graficznej i ekranu** w znaczeniu komputerowym. Charakterystycznymi elementami sterownika są **układy wejścia/wyjścia cyfrowe i analogowe** w odpowiedniej ilości do obsługiwanego procesu. Grafy Rys. 2.9 i Rys. 2.10 przedstawiają zasadę podłączania obwodów zewnętrznych na przykładzie kontrolera S7-1200. Każdy obwód zewnętrzny wymaga zasilania. Zasilanie sterownika jest 24VDC. Wejścia cyfrowe mają również zasilanie stałe

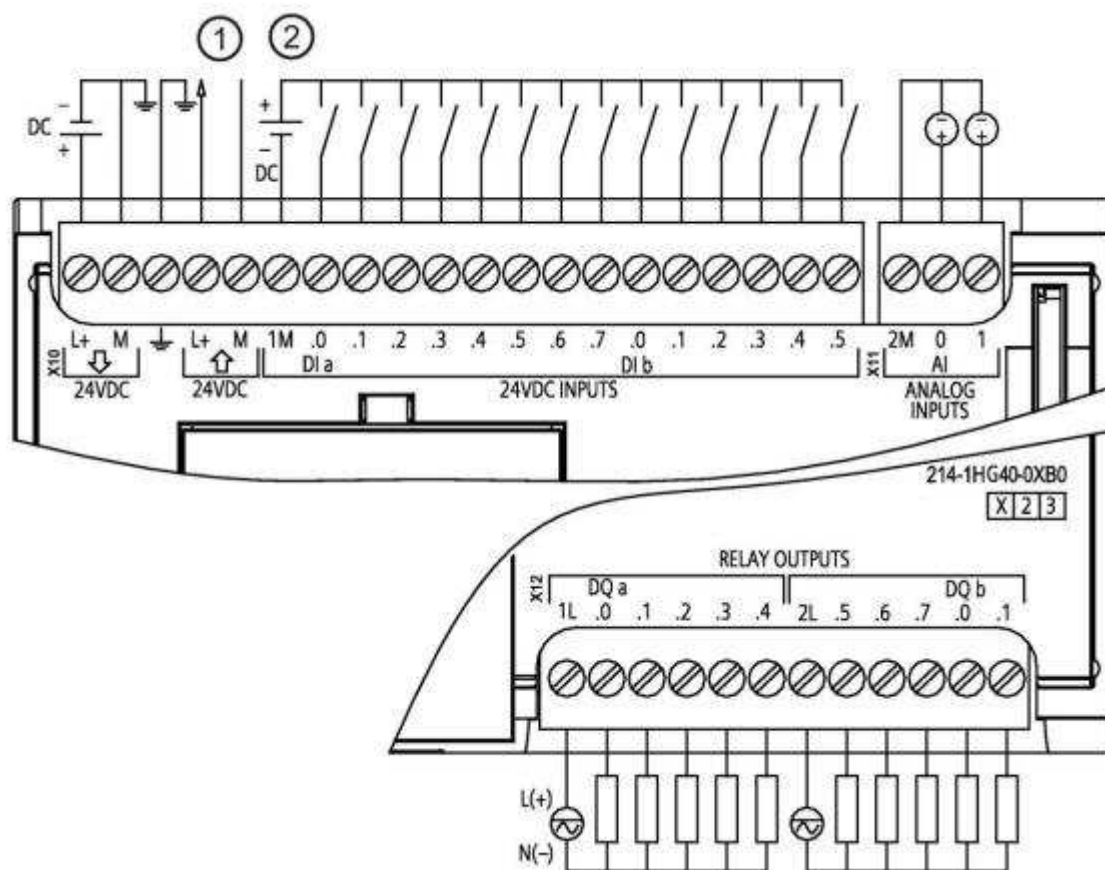
24VDC, ale nie oznacza to tego samego źródła zasilania. Z uwagi na bezpieczeństwo układu wskazane jest źródło zasilania niezależne.



Rys. 2.9 Podłączenie obwodów zewnętrznych sterownika – wyjścia tranzystorowe

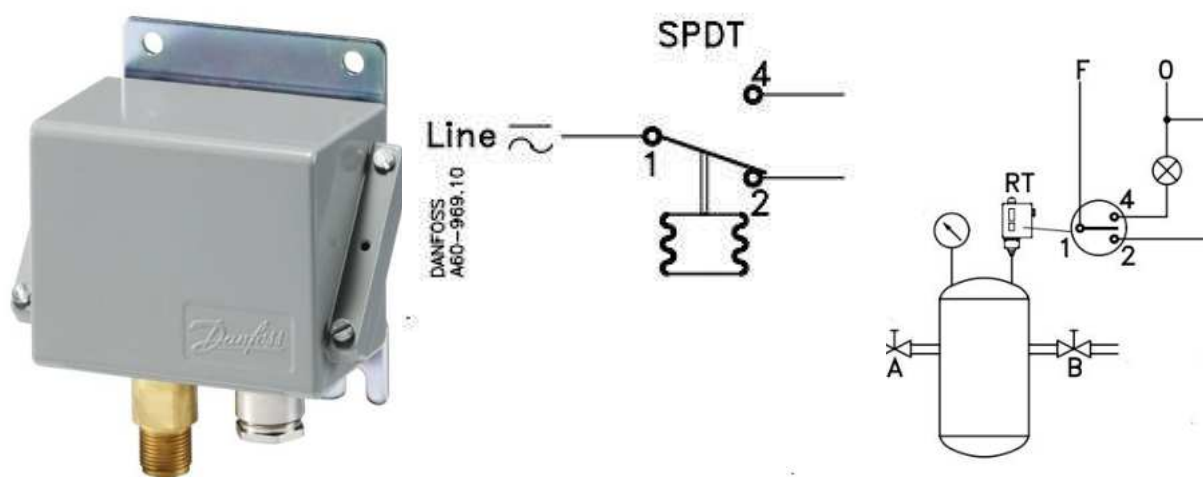
Wejścia cyfrowe dają informację z obiektu sterowania o wartości binarnej 0 lub 1. Wartość 0 jest definiowana dla napięcia wejściowego poniżej 5V i prądzie 1mA. W praktyce oznacza otwarcie obwodu. Wartość 1 jest definiowana dla napięcia wejściowego powyżej 15V i prądzie 2.5mA. W praktyce oznacza zamknięcie obwodu – dane dla S7-1200. Są to wejścia o numeracji od %I0.0 do %I1.5. Impedancja wejściowa obwodu dyskretnego wewnętrzna jest około 10kΩ. Opór wewnątrz wejścia ogranicza prąd w tym obwodzie, a czujnik zewnętrzny jest tylko przełącznikiem mechanicznym otwórz/zamknij obwód i wymaga zasilania. Prądy w tych obwodach są niskie i pozwalają używać wspólnego źródła zasilania dla wszystkich wejść dyskretnych. Przewód masowy jest wspólny i tylko jeden. W tym wypadku wszystkie czujniki binarne pracują na jednej wartości napięcia.

Każde wejście cyfrowe jest obwodem elektrycznym według prawa Ohma, w którym występuje opór elektryczny (wewnątrz wejścia), zasilanie i przełącznik mechaniczny wyłączenia obwodu. Wejście cyfrowe wykrywa napięcie i prąd na oporze wewnętrznym i zapisuje cyfrowo wartość 0 lub 1 do pamięci adresu wejściowego. Dalsze przetwarzanie informacji jest cyfrowe według programu.



Rys. 2.10 Podłączenie obwodów zewnętrznych sterownika – wyjścia przekaźnikowe

Znaczenie informacji binarnej zależy od podłączonego czujnika dwustanowego. Prosty układ stosowanym technicznie w układach jest presostat i termostat.



Rys. 2.11 Presostat, schemat połączenia, styki presostatu

Czujniki binarne zawierają styk przełączający, Rys. 2.11. Użytkownik może połączyć się do styków 1 – 2 (NZ – normalnie zamknięte) lub do styków 1 – 4 (NO – normalnie otwarte). Wybór styku powiązany jest z funkcją w układzie. Przykładowo w układzie analogowym do załączenia alarmu na sprężarce niskie ciśnienie oleju smarowego właściwym wyborem jest styk 1 – 2 razem z opóźnieniem na załączenie sprężarki. Po włączeniu sprężarki i upływie czasu zwłoki ciśnienie oleju przerzuci styki presostatu do pozycji 1 – 4. Obwód 1 – 2 będzie otwarty i bez alarmu. Zanik ciśnienia oleju załączy alarm i zatrzyma sprężarkę.

Jeżeli w układzie sterowania sprężarki zastosowano sterownik lepszym rozwiązaniem jest podłączyć wejście cyfrowe sterownika do styków 1 – 4. Po załączeniu sprężarki i uzyskaniu ciśnienia oleju sterownik odczyta stan wysoki '1'. Nie oznacza to stanu alarmowego, tylko sprawdzenie obwodu alarmowego – obwód i styk 1 -4 jest sprawny, zamknął się i przepływa przez niego prąd. Diagnostyka obwodu alarmowego jest ważna z uwagi na bezpieczeństwo pracy samej sprężarki, została wykonana automatycznie bez udziału człowieka i jest realizowana przy każdym załączeniu sprężarki. W tym rozwiązaniu operator nie spotka się z alarmem fałszywym. Przy załączonym obwodzie i stanie wysokim '1' alarm załączy sterownik programowo poprzez zastosowanie podstawowych funkcji logicznych – negacja sygnału.

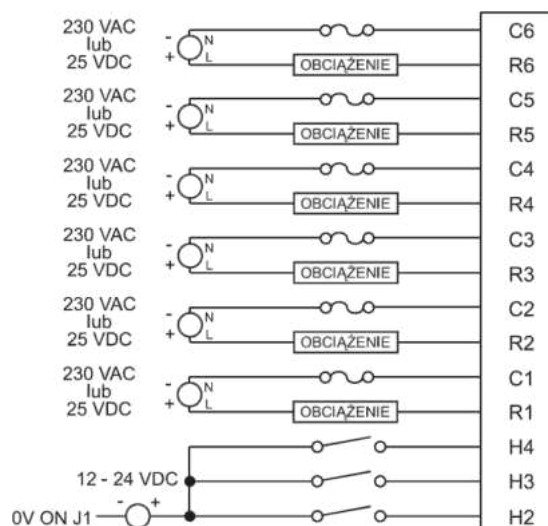
Obwód zamknięty interpretuje się jako sprawny. Obwód otwarty ma podwójną interpretację: jest otwarty lub uszkodzony. Zaleca się stosowanie obwodów zamkniętych, które można diagnozować automatycznie przez system. W ten sposób znane elementy techniczne połączone ze sterowaniem cyfrowym istotnie podnoszą bezpieczeństwo pracy urządzeń technicznych.

Wyjścia cyfrowe są jednym ze sposobów oddziaływania na obiekt poprzez układ sterowania. W tym przypadku załączamy elementy typu zawory, pompy, wentylatory, grzałki itp. Całe urządzenie załączamy sekwencyjnie według przyjętego algorytmu. Wyjścia cyfrowe rys ... i ... są typu tranzystorowe i przekaźnikowe. Wyjście cyfrowe po otrzymaniu wartości logicznej '1' załącza obwód, ale nie ma oporu wewnętrznego. Prąd w tym obwodzie zależy od źródła zasilania i mocy pobieranej przez odbiornik. Konstrukcja wybranych sterowników zapewnia każdy niezależny obwód wyjściowy, bez wspólnego przewodu zerowego. Przykładem jest sterownik firmy Horner typu HEXE220C112 o sześciu wyjściach cyfrowych i sześciu przewodach zerowych.

W każdym obwodzie występuje styk wyjścia cyfrowego, niezależne źródło zasilania i odbiornik. Obciążenie obwodu musi odpowiadać parametrom wyjścia sterownika. Bezpośrednie podłączenie odbiornika do wyjścia cyfrowego jest możliwe, jeżeli nie przekraczamy parametrów znamionowych wyjścia sterownika. Odbiorniki większej mocy są załączane poprzez układy pośrednie mocy. To pozwala sterować poprzez sterownik każdym odbiornikiem, w tym układem zasilania trójfazowego.

Wyjścia tranzystorowe charakteryzują małym prądem obciążenia do 0.5A i tylko dla zasilania prądem stałym VDC. W tych układach stosuje się wspólny przewód masowy i wspólne źródło zasilania. Zaletą tych układów jest duża częstotliwość przełączeń.

Wyjścia przekaźnikowe typu relay (Rys. 2.12) to przełączniki mechaniczne z pełną separacją galwaniczną od sterownika. Zapewnia to większą obciążalność złącza, załączanie obwodów dla prądu stałego VDC i prądu zmiennego VAC oraz stosowanie różnych napięć w każdym obwodzie przy niezależnym przewodzie zerowym. Wyjścia typu przekaźnikowego mają ograniczoną ilość przełączeń, przykładowo 100000 załączeń pod obciążeniem znamionowym. W tym rozwiązaniu możliwe jest bezpośrednie załączanie grzałek małej mocy dla utrzymania temperatury przed rozruchem urządzenia.



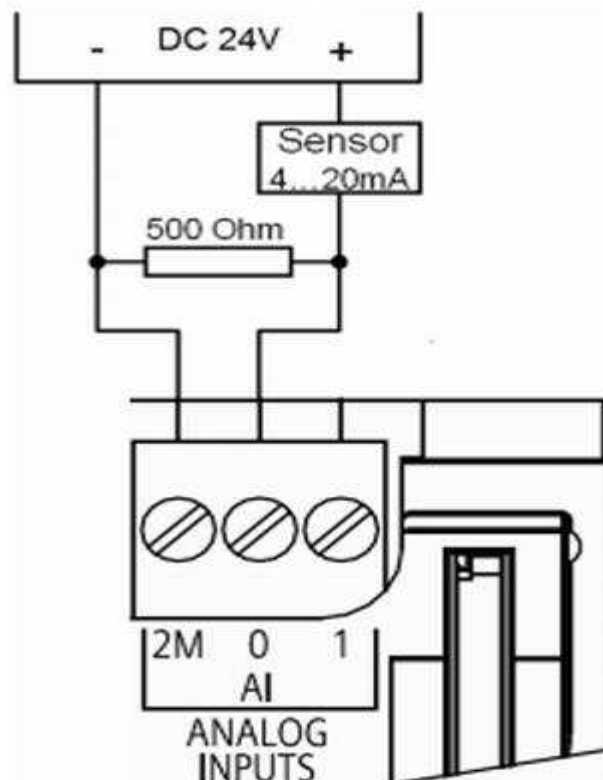
Rys. 2.12 Podłączenie obwodu wyjściowego do sterownika z wyjściem przekaźnikowym

Czujniki analogowe sterownika to pomiary wartości ciągłych. Pomiar wartości temperatury, ciśnienia, poziomu, przepływu i inne wymaga czujnika pomiarowego, przetwornika i modułu analogowo-cyfrowego na sterowniku. Parametry obiektów technicznych należy zmierzyć. W praktyce jest to związane z przejściem sygnału nieelektrycznego na sygnał elektryczny, a następnie przetworzyć na wartość standardową.

Przykładem jest przetwornik ciśnienia CCA-300 firmy SIMEX przeznaczony jest do pomiaru ciśnienia, Rys. 2.12. Elementem pomiarowym jest piezorezystancyjny czujnik krzemowy. Sygnał wyjściowy w standardzie 4-20mA otrzymujemy ze wzmacniacza sygnału. Przesyłanie sygnału do sterownika jest analogowe pojedynczym obwodem elektrycznym, Rys. 2.14 . Czujnik wymaga zewnętrznego źródła zasilania VDC.



Rys. 2.13 Przetwornik ciśnienia CCA-300 firmy SIMEX; zakres pomiarowy od -1...0 bar do 0...16 bar, sygnał wyjściowy: 4...20 mA, czujnik piezorezystancyjny krzemowy, dokładność 0,25%, sygnał wyjściowy 4...20 mA (2-przewodowe) lub 0...10V (3-przewodowe)



Rys. 2.14 Podłączenie sygnału analogowego do sterownika S7-1200
(modułu analogowo-cyfrowego sterownika)

Stosuje się również przetworniki inteligentne 'smart' z protokołem HART. W tym rozwiązaniu dostępny jest komunikator, który umożliwia zdalną nastawę zakresu i diagnostykę urządzenia, Rys. 2.15.

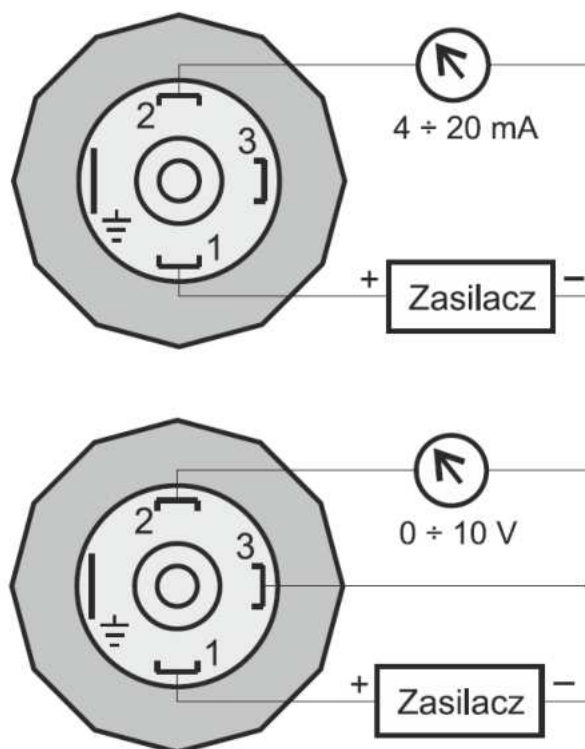
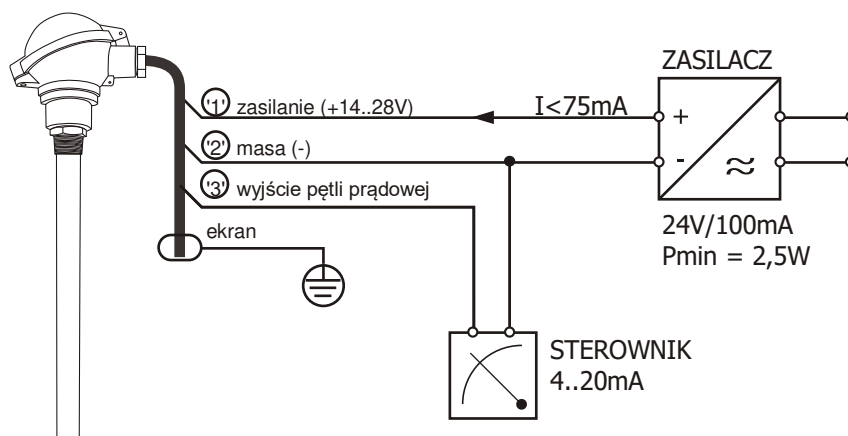


Rys. 2.15 Przetwornik ciśnienia PC-28 Smart z protokołem HART firmy SIMEX;
zakres pomiarowy -100 do +150 kPa, sygnał wyjściowy: 4...20 mA
(dwuprzewodowy) lub 0...10 V (trójprzewodowy)

Automatyka od początku stosuje przetworniki analogowe na sygnał standardowy do przesyłania sygnału na odległość. Stosowanie techniki cyfrowej wymusiło szerokie stosowanie standardu elektrycznego jako:

- sygnał prądowy: 4...20mA lub wcześniej 0...20mA,
- sygnał napięciowy: 0...10 V.

Sygnał napięciowy wprowadza błąd pomiarowy spowodowany spadkiem napięcia na przewodach, nie można stosować przy długich liniach pomiarowych. Tej wady nie ma sygnał prądowy długość linii pomiarowej nie wpływa na wartość prądu w obwodzie. W technice cyfrowej urządzenia mają autodiagnostykę. Prąd 0mA tłumaczy się jako sygnał o wartości 0mA lub uszkodzenie obwodu, stąd standard sygnału 4...20mA jest dominujący, Rys. 2.16.



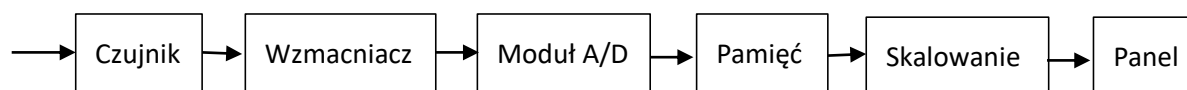
Rys. 2.16 Podłączenie czujnika analogowego z zewnętrznym zasilaniem i sterownikiem

Wejścia analogowe sterownika przetwarzają sygnały analogowe standardowe z czujnika pomiarowego na wartość cyfrową. Odpowiada za to moduł analogowo-cyfrowy A/D. Wartość cyfrowa zapisywana jest w pamięci rejestrowej sterownika, skąd jest przetwarzana cyfrowo przez sterownik. Wartość cyfrowa nie jest standaryzowana i nie odpowiada wartości mierzonej. Obowiązuje zasada przetwarzania proporcjonalnego. Sterownik jako urządzenie cyfrowe (z procesorem) przetwarza dane binarne o znanej długości słowa. Pierwsze komputery PC pracowały z 8-mio bitowym słowem. W miarę postępu długość słowa była wydłużana np. do 128-mio bitowego. W sterownikach dominuje długość słowa 16-to bitowa lub krótsza i procesor 16-to bitowy. Słowo 16-to bitowe dzieli się na dwa 8-mio bitowe baity. Numeracja pamięci 16-to bitowej co dwa baity. Kolejne wartości analogowe odczytuje się pod adresem %IW64, %IW66,

Niezależnie od długości słowa podstawowego przetwarzanie cyfrowe może być 8, 10, 12, 14 i 16-to bitowe – tzw. rozdzielczość modułu A/D. Każda wartość analogowa jest przetwarzana przez moduł A/D na wartość cyfrową całkowitą i długości słowa danego modułu A/D. Im więcej bitów ma dane słowo, tym większa jest liczba binarna i dokładność przetwarzania. Najstarszy bit słowa w niektórych układach rezerwuje się na znak sygnału -/+. Przykładowo w rejestrze 16-to bitowym możemy zapisać liczbę -/+0-32767. Dla pomiaru analogowego 0-10V i przetwarzania 16-to bitowego rozdzielczość pomiaru wynosi 0.3mV ($10V/32768=0.000305$). Wysoka rozdzielczość pomiaru wywołuje efekt 'pływania' wartości mierzonej na młodszych bitach. Obniżanie długości słowa modułu A/D likwiduje efekt 'pływania' – stabilizuje pomiar i jego częste zmiany.

Dokładność pomiaru wartości analogowej zależy od klasy dokładności zastosowanego czujnika. Przetwarzanie cyfrowe wprowadza kwantyzację sygnału. Kolejne wartości nie mogą się różnić mniej niż wynika to z rozdzielczości modułu A/D. Zmiana wartości o 0.3mV zmienia wartość cyfrową +/-1 dla przetwarzania 15-to bitowego, to jest mniej niż klasa dokładności 0.5 czujnika. Dla wejścia prądowego kwantyzacja wynosi 0.5μA

Wyświetlanie wartości mierzonej na panelu operatorskim składa się z kilku elementów: czujnika, wzmacniacza, modułu A/D, zapisu do pamięci, procesu skalowania i prezentacji na panelu, Rys. 2.17.



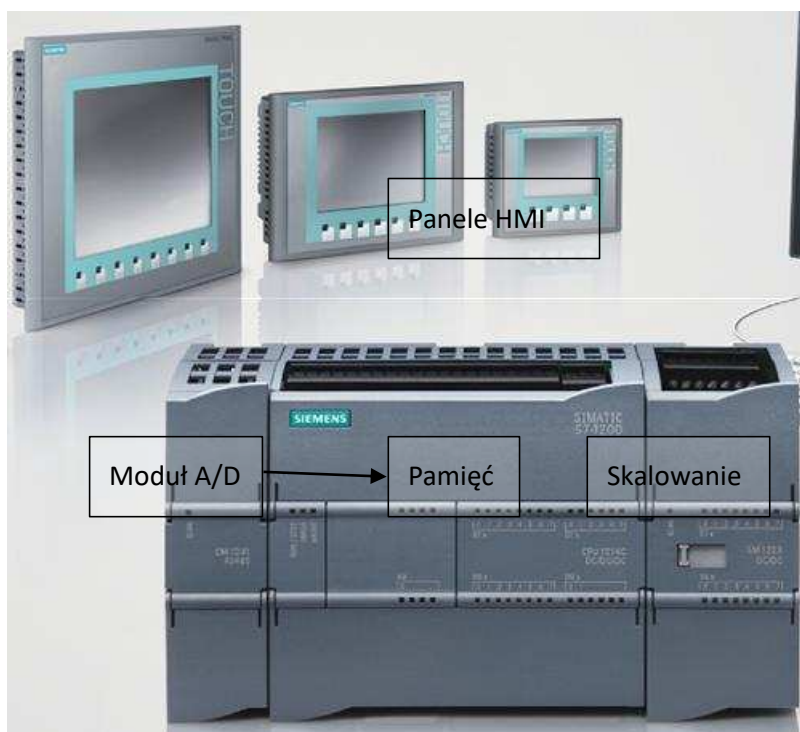
Rys. 2.17 Przykład układu pomiarowego analogowego z przetwarzaniem cyfrowym b

W układzie rzeczywistym występuje czujnik pomiarowy (Rys. 2.18), sterownik i panel operatorski(Rys. 2.19).

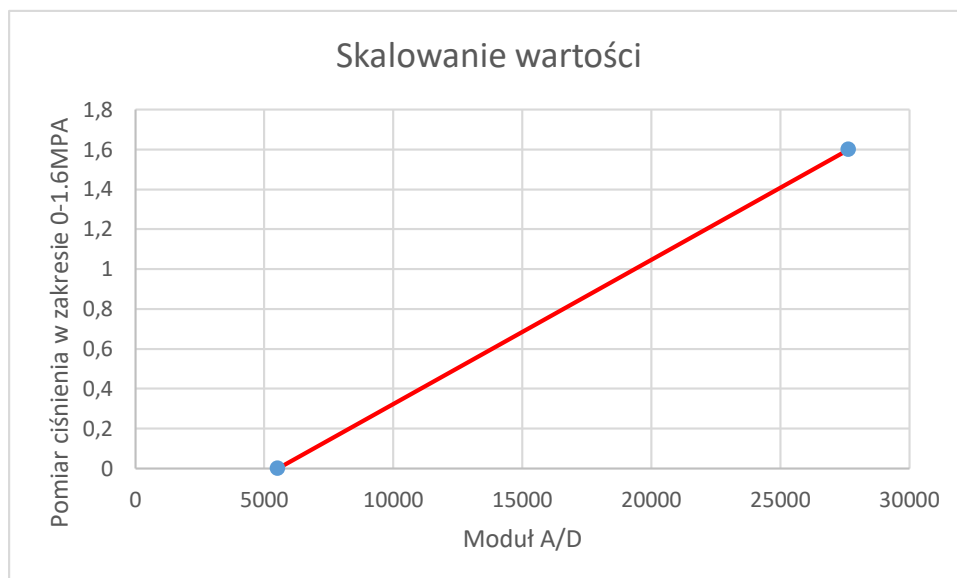
Skalowanie wartości mierzonej i jej prezentacja na panelu ma tylko znaczenie edytorskie. Sterownik przetwarza bezpośrednio wartości z modułu A/D, nie ma wiedzy jaki sygnał fizyczny i w jakich jednostkach mierzy. Skalowanie sygnału jest programowe. Należy w programie zdefiniować rodzaj sygnału, zakres pomiarowy czujnika i jego jednostkę SI. Po przeliczeniu wynik wyświetlamy na panelu operatorskim i jest zrozumiały dla operatora. Często wymaga to przejścia z wartości całkowitej do zmiennej typu rzeczywistego (real) i operacji zmiennoprzecinkowych, Rys. 2.20.



Rys. 2.18 Pomiar ciśnienia – układ rzeczywisty



Rys. 2.19 Sterownik S7-1200 i panel operatorski



Rys. 2.20 Skalowanie wartości w sterowniku

W sterowniku S7-1200 firmy Siemens każde wejście analogowe przyjmuje standard 0-20mA, rzeczywiste parametry to 0-23,7mA i jest przetwarzane na wartość całkowitą 0-32767. Jeżeli teraz podłączymy czujnik pomiarowy w standardzie 4-20mA, to odpowiada to wartościom cyfrowym dla 4mA – 5530, a dla 20mA – 27648. Pozostały zakres ma znaczenie diagnostyczne, prąd poniżej 3.8mA oznacza przerwanie obwodu oraz prąd 20.5mA i więcej oznacza zwarcie w obwodzie.

Wyjścia analogowe to wyjścia sterujące procesem wartościowo, inaczej wyjścia regulacyjne. Sterownik według programu wyznacza nowe wartości sterujące i wprowadza je do modułu cyfrowo-analogowego D/A, który wymusza proporcjonalny sygnał standardowy analogowy w obwodzie wyjściowym. Obowiązuje standard elektryczny napięciowy 0-10V lub prądowy 4-20mA (dopuszczalny 0-20mA). Moduł D/A może przekształcać sygnał o różnej długości słowa, ma własną rozdzielczość, nie więcej niż 16-cie bitów. Często wyjście analogowe ma niższą rozdzielczość od wejścia analogowego.

Sygnał wyjściowy najczęściej kalkuluje się według algorytmu regulatora PID i oznacza pozycję zaworu regulacyjnego, Rys. 2.21. Sygnał sterujący przesyła się analogowo do ustawnika pozycyjnego zaworu regulacyjnego. Zawór regulacyjny wymaga układu wykonawczego – siłownika. W przedstawionym rozwiązaniu siłownik jest pneumatyczny, a ustawnik pozycyjny przekształca sygnał analogowy na sygnał pneumatyczny dostosowany do parametrów siłownika. Dodatkowo koryguje pozycję końcową zaworu do wartości sygnału sterującego poprzez sprzężenie zwrotne z trzpieniem zaworu.

Wyjście analogowe modułu D/A zawsze wymaga układu wzmocnienia mocy dopasowany do elementu wykonawczego układu sterowania. Urządzenie wykonawcze wymaga kalibracji. Wartości skrajne 4 i 20mA należy skalibrować z pozycją zaworu otwarty i zamknięty.

Obwody wejścia i wyjścia analogowe sterownika można łatwo diagnozować ręcznie. Należy zastosować generator sygnału standardowego, Rys. 2.22, i podłączyć do obwodu. Jeżeli sprawdzamy wejście analogowe, to ustawiona wartość sygnału musi być wskazana przez sterownik. Jeżeli sprawdzamy wyjście analogowe, to ustawiona wartość sygnału musi być wykonana przez element wykonawczy.



Rys. 2.21 Układ wykonawczy dla wyjścia analogowego sterownika – zawór regulacyjny z ustawnikiem pozycyjnym firmy Siemens



Rys. 2.22 Generator sygnału standardowego do diagnostyki obwodów sterownika

Porty komunikacyjne sterownika

Sterowanie cyfrowe sprowadza się do prostych obwodów elektrycznych podłączonych do wejść i wyjść sterownika oraz centralnej pozycji układu cyfrowego, który pracuje według programu i zawiera procesor. Program należy wprowadzić do sterownika, a dane o układzie sterowania można pobierać do analizy. Do tego służą moduły komunikacyjne sterownika zintegrowane i zewnętrzne. W innych sytuacjach konieczna jest wymiana danych pomiędzy sterownikami i serwerem – sieć w standardzie Profibus, Profinet, Ethernet, Modbus i złącza RS-232, RS485, RJ45, USB. Również występują moduły do komunikacji bezprzewodowej.

Zadaniem sterownika jest bezawaryjne i bezpieczne sterowanie obiektem technicznym. Gromadzenie danych oraz ich przesyłanie jest funkcją dodatkową rozwiniętą w technice cyfrowej. W tym zakresie sterownik tylko udostępnia dane bieżące o procesie poprzez wybrany port komunikacyjny. Pozostałe operacje przetwarzania danych przejmują serwery. W ten sposób powstaje struktura rozproszona układu automatyki i sterowanie nadrzędne – systemy SCADA. Sterownik realizuje sterowanie bezpośrednie obiektem.

Przykłady producentów i sterowników zawiera tabela 2.1.

Tabela 2.1 Lista wybranych producentów sterowników

Producent	Inteligentne przekaźniki	Małe	Średnie	Duże
Siemens	Logo	SIMATIC S7-200	SIMATIC S7-300,400	SIMATIC S7-1200
Schneider Electric	Zelio	Nano, Micro, Twido	Premium, Compact, Momentum	Quantum
GE Fanuc	VersaMax-Nano	VersaMax-Micro	90-30, VersaMax, PACSystems RX3i	90-70, PACSystems RX7i
Mitsubishi Electric	ALPHA	MELSEC FX1, FX2	MELSEC QnAS	MELSEC QnA, MELSEC System Q
Omron		CPM1, CPM2, CQM1H	C200H-alpha, CJ1, CS1	CVM1
Rockwell Automation (Allen-Bradley)	Pico	MicroLogix	SLC500, FlexLogix, ControlLogix	PLC-5

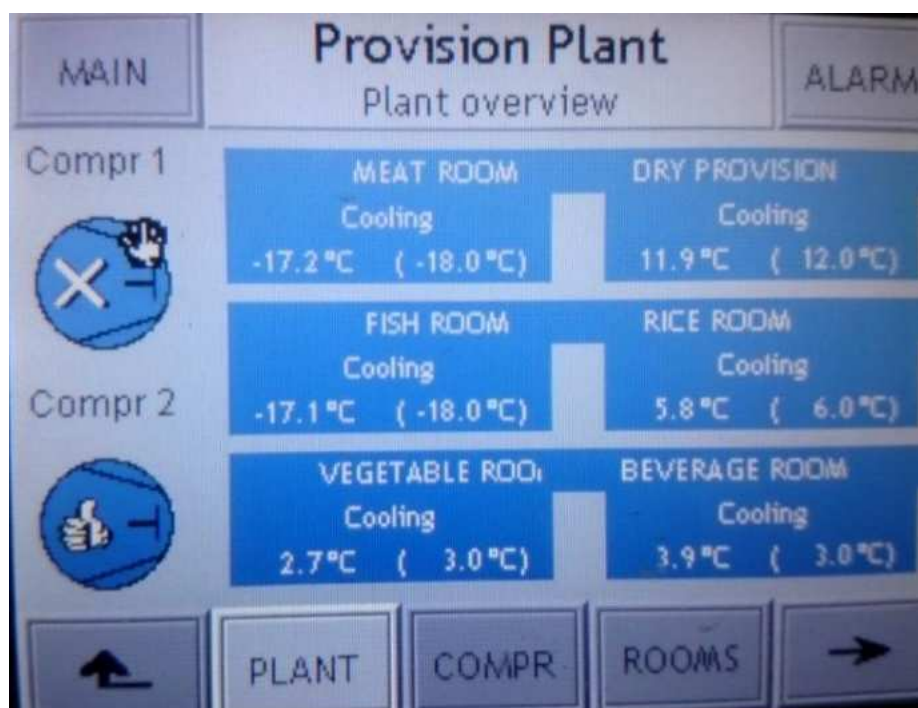
2.2 Panele Operatorskie

Panele operatorskie są częścią układu sterowania, pełnią funkcję graficzną, prezentują graficznie proces, udostępniają parametry cyfrowo lub wykreślne – histogramy, umożliwiają

wprowadzenie nowych wartości jako sterowanie lokalne, Rys. 2.23 i Rys. 2.24. Połączenie panelu ze sterownikiem realizuje się przez port komunikacyjny.



Rys. 2.23 Panel operatorski z przykładowym procesem firmy Siemens



Rys. 2.24 Panel operatorski firmy Mitsubishi z programem chłodni prowiantowej

Panele operatorskie identyfikuje się skrótem HMI – Human Machine interface. Są przykłady, gdzie producent integruje panel HMI ze sterownikiem PLC – firma Horner, Rys. 2.25.



Rys. 2.25 Sterownik firmy Horner z panelem HMI, model X5 i X2

Takie podejście stosuje się dla małych układów sterowania. Upraszcza to tworzenie aplikacji do wizualizacji procesu.

3. Programowanie sterowników PLC

Sterownik PLC został zbudowany do obsługi wszystkich wejść/wyjść. Poprawnie połączony układ jest wstępem do właściwej pracy systemu. Układ wymaga programu działania. Sterownik jest układem uniwersalnym do obsługi dowolnego systemu zbudowanego według podanych zasad obwodów elektrycznych i wymaga indywidualnego programu do danego systemu. System sterowania składa się z **obwodów analogowych elektrycznych**, a do właściwego funkcjonowania zawiera część cyfrową tj. **sterownik i program sterowania**. Zasady pisania programu sterowania zostaną przedstawione poniżej.

Producent sterownika stosuje ogólne zasady programowania układów zapewniając własną aplikację programową dostosowaną do własnych wyrobów z licznymi modyfikacjami. W efekcie każdy sterownik ma liczne wyjątki, co sprawia, że każdy układ wymaga szerokiej wiedzy z zasad programowania pod kątem producenta i modelu sterownika.

Do poprawnej pracy sterownika konieczny jest program sterowania. Wymaga się jego zapisu na nośnikach trwałych i zewnętrznych oraz realnego odczytu przez sterownik. Utrata zawartości pamięci operacyjnej po wyłączeniu zasilania nie oznacza utraty programu. Wymiana uszkodzonego sterownika na nowy związana jest z przełożeniem karty pamięci z programem. Każde załączenie sterownika związane jest z odczytem programu z nośnika trwałego. Pamięć trwała zewnętrzna skutecznie chroni nas przed utratą programu. Sama konstrukcja sterownika ma zapis programu w pamięci trwałej typu EPROM. Pamięć zewnętrzna np. karty SD są dodatkowym zabezpieczeniem nie w każdym sterowniku. Program tworzymy na komputerach osobistych z wymaganą aplikacją. Zapis programu jest dostępny w komputerze i przesyłany do sterownika dowolnym złączem np. port szeregowy RS232/485, port USB lub ethernetowy RJ45 oraz zdalnie na podstawie adresu IP w sieci LAN. Do celów komercyjnych wymagany jest zapis na pamięci zewnętrznej oraz jej odczyt po załączeniu

zasilania sterownika. Normalne załączenie sterownika oznacza odczyt programu z pamięci EPROM wewnętrznej lub zewnętrznej.

W każdym przypadku od momentu załączenia sterownik wykonuje cały program w pętli zamkniętej, Rys. 3.1.



Rys. 3.1 Algorytm pracy sterownika PLC

Start i realizacja programu to:

- inicjalizacja sterownika, wczytanie programu, sprawdzenie układów sterownika
- odczyt sygnałów wejściowych sterownika
- przeliczenie programu sterowania
- zapis sygnałów wyjściowych sterownika
- obsługa portów komunikacyjnych, połączenie z serwerem, możliwa zmiana programu, połączenie z innymi sterownikami, przesyłanie danych
- autodiagnostyka obwodów zewnętrznych i układów wewnętrznych sterownika, ewentualne przerwanie działania
- rozpoczęcie nowej pętli programu

Przedstawione kroki obejmują wykonanie pełnego programu z odczytem wejść i ustawieniem wyjść. Po wykonaniu programu sterownik inicjuje kolejny pełny krok realizacji programu i to powtarza w nieskończoność do momentu zatrzymania sterownika lub błędu i zatrzymania. W pamięci sterownika jest tylko jeden aktualny program.

Występuje pojęcie czasu realizacji programu i czas próbkowania układu. Czas realizacji programu musi być krótszy od czasu próbkowania. Dobór czasu próbkowania zależy od parametrów dynamicznych i stałych czasowych obiektu sterowania. W domyśle nowa pętla programu rozpoczyna się po upływie czasu próbkowania zdefiniowana przez programistę. W praktyce układy przełączające nie muszą być realizowane co stały okres czasu i sterownik może rozpocząć nowy cykl programu zaraz po zakończeniu poprzedniego. Istnieją fragmenty sterowania, które zawierają stałe czasowe i te

elementy programu są programowane z wymaganym okresem próbkowania. Stosowane rozwiązania zależą od producenta sterownika i wymagają dobrej znajomości procesu i języka programowania. Różnice w programowaniu są widoczne w programowaniu wybranego modelu sterownika.

Przykładowe aplikacje do programowania sterowników:

- LOGO!Soft Comfort V8.0 dla sterowników LOGO bezpłatny
- Cscape 9.90 SP8 dla sterowników Horner bezpłatny
- Proficy Machine Edition dla sterowników GE-Fanuc licencjonowany
- TIA Portal v18 dla sterowników Siemens licencjonowany, (Totally Integrated Automation TIA)

Środowisko programistyczne dobierane jest przez producenta danego sterownika i może być w postaci :

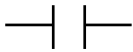
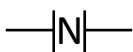
1. LD - ladder diagram, programowanie drabinkowe
2. FBD - function block diagram, programowanie według schematów blokowych
3. ST - structured text, algorytm zapisany w języku wyższego poziomu np. C++
4. IL - instruction list, tekstowa lista instrukcji
5. SFC - sequential function chart, użycie grafów sekwencji

3.1 Programowanie w języku drabinkowym LD

Najbardziej rozpowszechniony i dominujący w programowaniu sterowników jest język drabinkowy LD łączony z pozostałymi formami programowania, co ma wygenerować efektywny kod programu. Programowanie danego sterownika jest specjalizacją samą w sobie. Wymaga od programisty znajomości sterownika, programowania, automatyki i procesu technologicznego. Zmiana dowolnego punktu wymusza wprowadzenie nowych rozwiązań w programowaniu. Wymaga się ciągłego doskonalenia danego programu i nowych jego wersji. Jest to proces dobrze znany z informatyki.

Elementy stosowane do pisania programu w języku drabinkowym przedstawiono w tabeli 3.1.

Tabela 3.1 Podstawowe elementy binarne obsługi wejść/wyjść

Element	Opis
	Styk wejściowy normalnie otwarty
	Styk wejściowy normalnie zamknięty
	Wykrywanie zbocza sygnału narastającego na wejściu
	Wykrywanie zbocza sygnału opadającego na wejściu

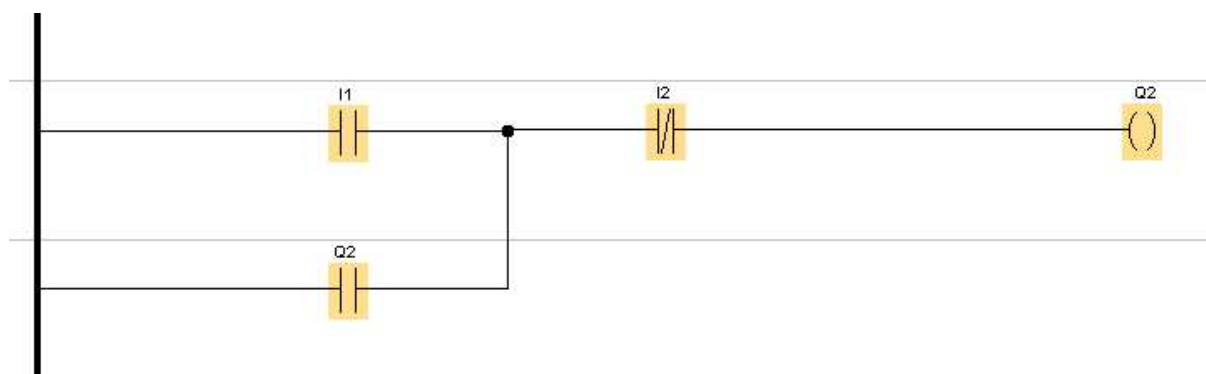
TIME	Zmienna o długości 4 bajtów jako liczba milisekund, zawiera dzień, godzinę, minuty, sekundy i milisekundy w zakresie -24d_20h_31m_23s_648ms...24d_20h_31m_23s_647ms
LTIME	Zmienna o długości 8 bajtów jako liczba nanosekund
DATE	Zmienna o długości 2 bajtów, liczba dni od 1 stycznia 1990r
Time_of_Day	Zmienna o długości 4 bajtów jako liczba milisekund od północy danego dnia, 00:00:00.000 to 23:59:59.999
CHAR	Zmienna znakowa o długości jednego bajtu
STRING	Ciąg znaków

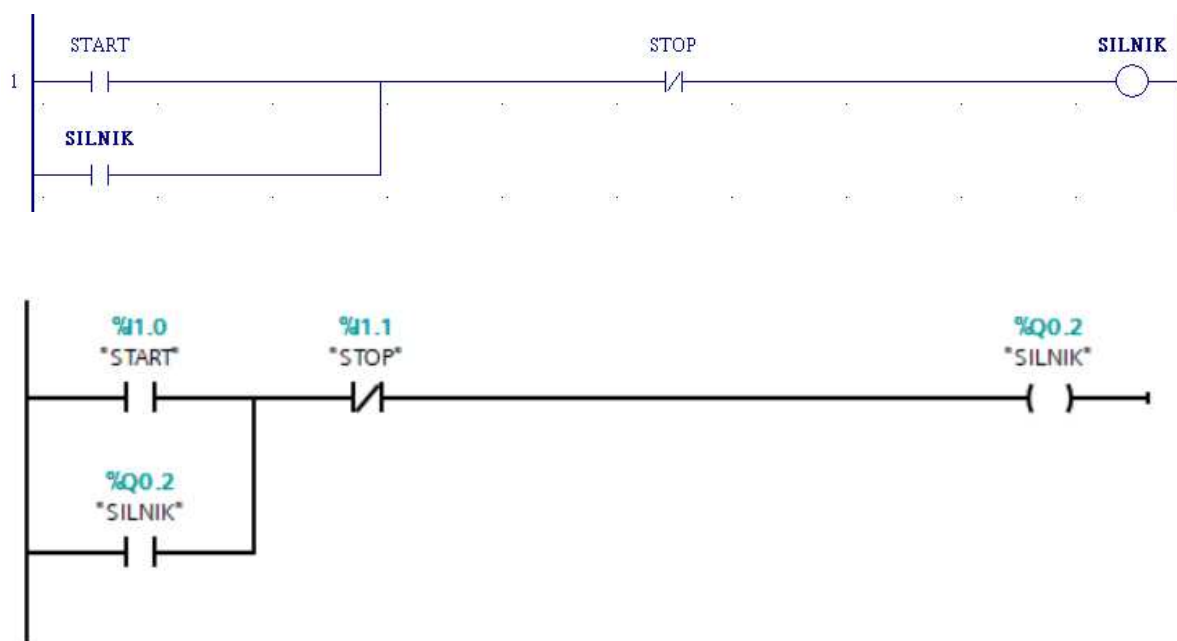
Tabela 3.3 Adresowanie we/wy na sterownikach

GE-Fanuc	SIMATIC S7-1200	Opis
%I3	%I0.3	Wejście binarne
%Q4	%Q0.4	Wyjście binarne
%M	%M, %MD	Adresowanie dostępnej pamięci
%S, %G, %R, %S, %T	Deklaracja zmiennej	Zmienne systemowe, globalne, rejestrowe, systemowe, tymczasowe
%AI1	%IW64	Wejścia analogowe
%AQ1	%QW80	Wyjścia analogowe

Zmienna, adres i numer adresu są ze sobą powiązane, są różnej długości. Przykłady adresowania we/wy podano w tabeli 3.3. Konieczne jest planowanie numeru adresu, żeby nie pokryły się dla kolejnej zmiennej. Zmienna binarna może zajmować jeden bajt (GE-Fanuc %I5) lub jeden bit (SIMATIC %I0.5). Wówczas na jednym baidzie deklarujemy osiem zmiennych binarnych. Zmienne zapisywane na kilku bajtach wymagają planowania przesunięć adresowych. Po deklaracji typu integer np. adres %M4, kolejny wolny adres jest %M6 oraz po deklaracji typu real np. adres %M8, kolejny wolny adres jest %M12. Numer adresu oznacza kolejny bajt. Dla zmiennych tablicowych (array) odstępów w numeracji adresów są wielokrotne.

Po wstępnych opisach można podać przykład zapisu wiersza w języku LD, Rys. 3.2. Na sterowniku LOGO podaje się numery wejść (I1, I2, ...) oraz wyjść (Q1, ...). Na sterowniku GE-Fanuc wprowadzamy nazwy zmiennych, do których przypisane są adresy we/wy. Na sterowniku S7-1200 widoczne są nazwy zmiennych i adresowanie.





Rys. 3.2 Sterowanie silnikiem włącz/wyłącz na sterowniku LOGO, GE-Fanuc, S7-1200

3.2 Rozszerzenia wierszy w języku drabinkowym LD

Wiersze w języku drabinkowym są rozwijane o liczne bloki dodatkowe wymienione w tabelach 3.4, 3.5 i 3.6.

Tabela 3.4 Pozostałe elementy logiczne w programowaniu LD

Element	Opis
EQ(=), NE(<>)	Relacja równości, nierówności
GT(>), LT(<)	Relacja większy, mniejszy
LE(<=), QE(>=)	Relacja większy równy i mniejszy równy
Range(a,b)	Relacja w zakresie
AND, OR, NOT, XOR	Funkcje logiczne na słowie
SHL, SHR	Przesunięcie bitów na słowie w lewo i w prawo z dopisaniem zera
ROL, ROR	Przesunięcie cykliczne bitów na słowie w lewo i w prawo
Przerzutnik SR	Ustawia wyjście na 1 po wprowadzeniu 'Set' i zeruje wyjście po wprowadzeniu 'Reset'

Tabela 3.5 Timery w programowaniu LD

Element	Opis
ON Delay Timer - ONDTR	Ustawia wyjście po załączeniu bloku i upływie zadanego czasu,

	zeruje wyjście po wyłączeniu bloku
Off Delay Timer - OFDT	Ustawia wyjście po załączeniu bloku, zeruje wyjście po zadany czasie
On/Off Delay Timer - TMR	Ustawia wyjście po załączeniu bloku i upływie zadane czasu, zeruje wyjście po wyłączeniu bloku i upływie zadane czasu

Timerów jest dużo więcej i występują duże różnice pomiędzy producentami sterowników i danej wersji producenta sterownika

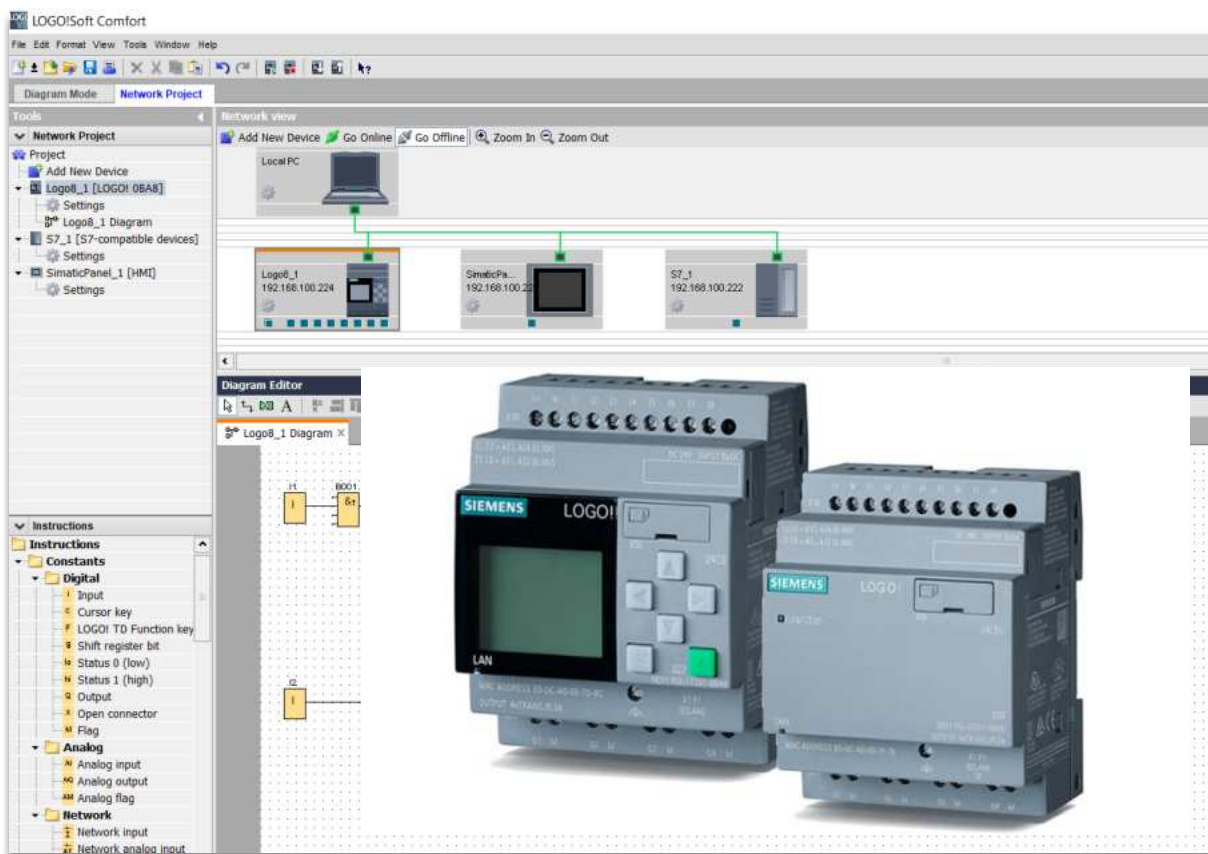
Tabela 3.6 Liczniki w programowaniu LD

Element	Opis
Licznik w górę - UPCTR	Zlicza impulsy wejściowe w górę, ustawia wyjście na 1 po przekroczeniu zadanej wartości
Licznik w dół - DNCTR	Zlicza impulsy wejściowe w dół, ustawia wyjście na 1 po przekroczeniu zadanej wartości

Wymienione elementy są formą bloku wpisaną w drabinkę programu, mają po kilka wejść i wyjść, charakteryzują się wejściem zezwalającym (enable) i wyjściem potwierdzającym poprawne wykonanie bloku.

4. Programowanie sterownika Siemens LOGO

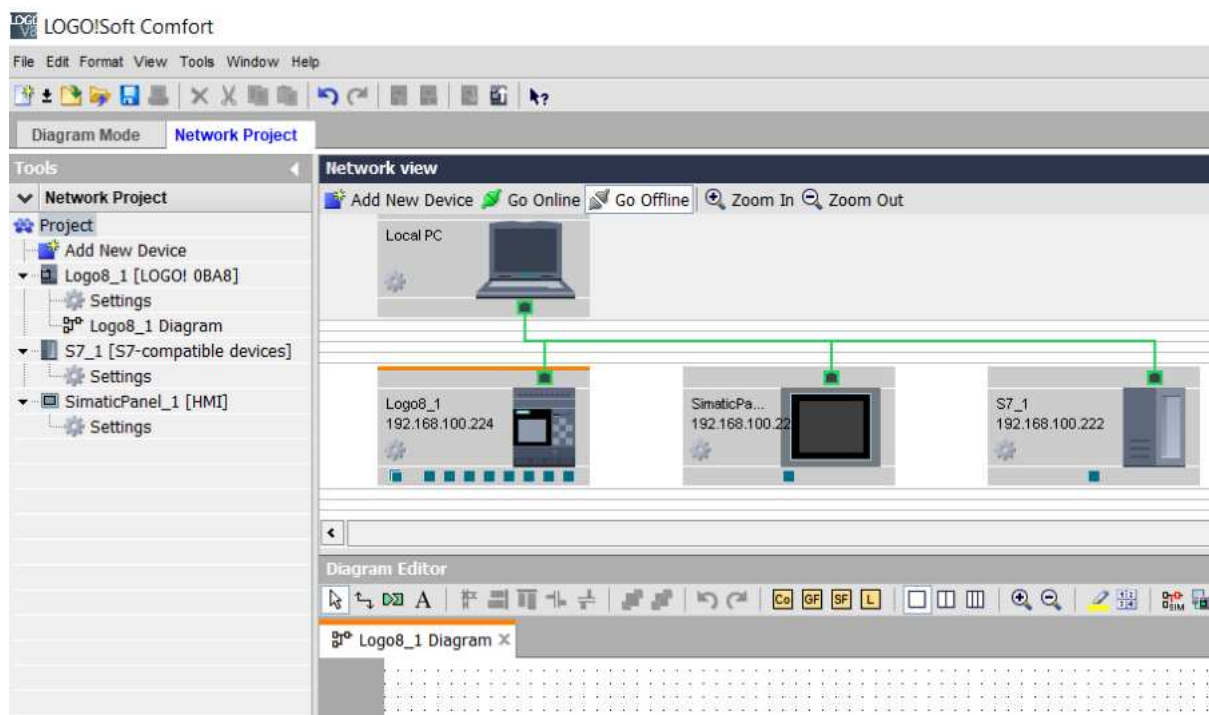
Prosty sterownik Siemens LOGO oraz program przedstawiono na Rys. 4.1.



Rys. 4.1 Sterownik Siemens LOGO i program LOGO Soft Comfort

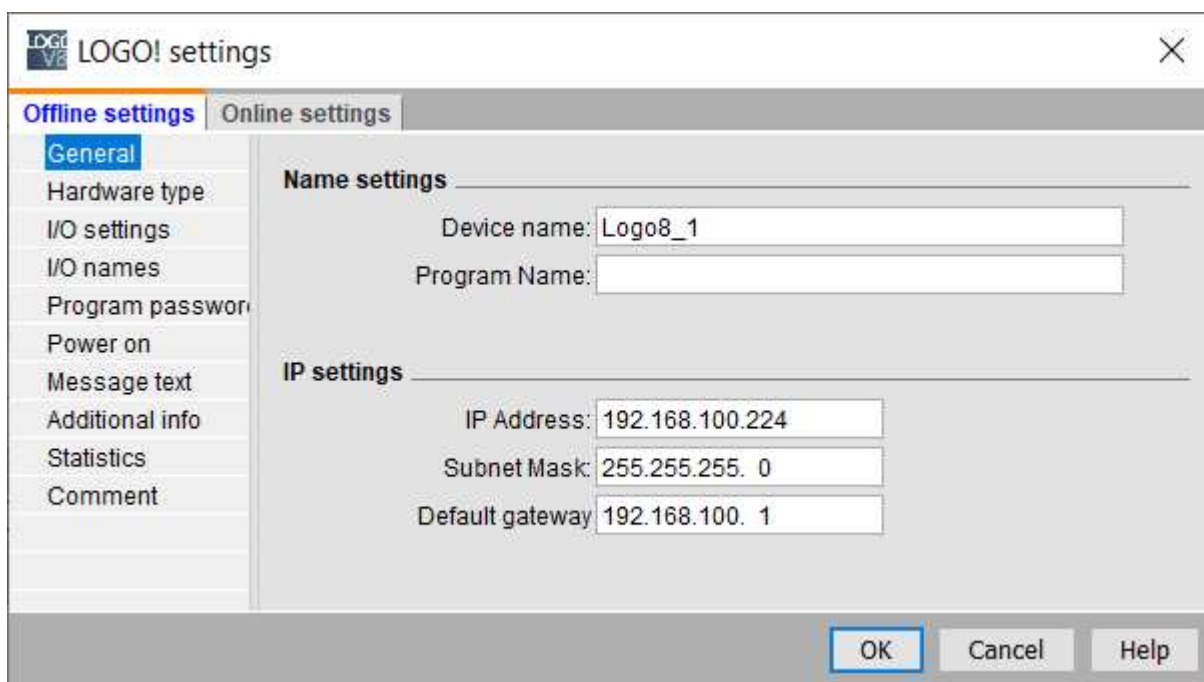
4.1 Ustawianie połączenia Ethernet

Połączenie z nowym Logo! odbywa się za pomocą połączenia Ethernet (Rys. 4.2) – służy do tego protokół TCP/IP v.4. Komputer PC, na którym jest tworzony program i przesyłany następnie do pamięci sterownika musi być w tej samej sieci, co Logo. Oznacza to, że maska podsieci sterownika i komputera PC muszą być takie same, natomiast adresy IP muszą leżeć w puli dostępnej dla danej maski. Odpowiedni adres należy wpisać ręcznie korzystając z menu sterownika – w naszym wypadku jest to Logo Soft Comfort (*Tools à Transfer à Configure Network Address*). Jednak przy pierwszym połączeniu sterownik i komputer PC muszą odnaleźć się w sieci, natomiast parametry protokołu TCP/IP prawie na pewno będą różne. Dlatego za pierwszym razem należy sprawdzić, jaki adres ma nasz komputer PC, a następnie ustawić ręcznie adres za pomocą menu sterownika Logo. Komputer i Logo muszą pracować w tej samej sieci oraz adres sterownika musi być znany, wpisany i ustalony „ręcznie”, Rys. 4.3 i Rys. 4.4.



Rys. 4.2 Przykładowy projekt sieci i adresów IP ze sterownikiem LOGO

Rys. 4.3 Przykładowy adres IP komputera



Rys. 4.4 Przykładowy adres IP sterownika Logo

Po wykonaniu adresowania sterowników w programie przystępujemy do programowania LOGO!.

Programowanie FBD - Function Block Diagram

Sterownik LOGO! posiada moduł graficzny i klawiaturę podstawową do programowania bezpośredniego. Bloki programowe to funkcje używane w celu określenia sposobu konwersji sygnału wejściowego na wyjściowy. Blokami są bramki logiczne podstawowe (AND, NAND, OR, NOR, XOR itd.), przerzutniki, liczniki, rejestry oraz inne, bardziej rozbudowane funkcje. Pojedynczy diagram może zawierać do 400 bloków, ale pod warunkiem, że rozmiar otrzymanego w wyniku ich umieszczenia na schemacie programu wynikowego nie przekroczy wielkości pamięci wyrażonej w bajtach. Oprócz samych funkcji, w pamięci przechowywane są również robocze wartości programu (np. stan licznika godzin) oraz dane bloków pod warunkiem, że została uaktywniona opcja podtrzymania pamięci. W Logo dostępne są następujące zasoby:

- pamięć programu: 3800 bajtów,
- maksymalna liczba bloków: 400,
- maksymalna liczba podtrzymywanych zmiennych: 250.

Aby wyświetlić informację o dostępnej pamięci, należy wykonać następujące czynności:

- przełączyć Logo w tryb programowania,
- umieścić kursor w pozycji „Edit” i nacisnąć OK.
- umieścić kursor w pozycji „Memory” i nacisnąć OK.

Na ekranie LOGO! pojawi się komunikat „Free Memory” zawierający informację o zajętości pamięci programu, liczbie używanych bloków oraz podtrzymywanych zmiennych.

Umieszczenie bloku na schemacie powoduje automatyczne nadanie mu numeru porządkowego. Numeracja ułatwia orientację oraz wyszukiwanie błędów.

Każdy blok można połączyć do wejścia posługując się jego numerem. Dzięki temu można w programie wykorzystywać wyniki działania bloków pośrednich. Maksymalnie do 100 bloków można

nadać nazwy składające się z co najwyżej 8 liter. Dzięki takiemu zabiegowi ułatwia się użytkownikowi orientację oraz modyfikację odpowiednich parametrów za pomocą klawiatury sterownika.

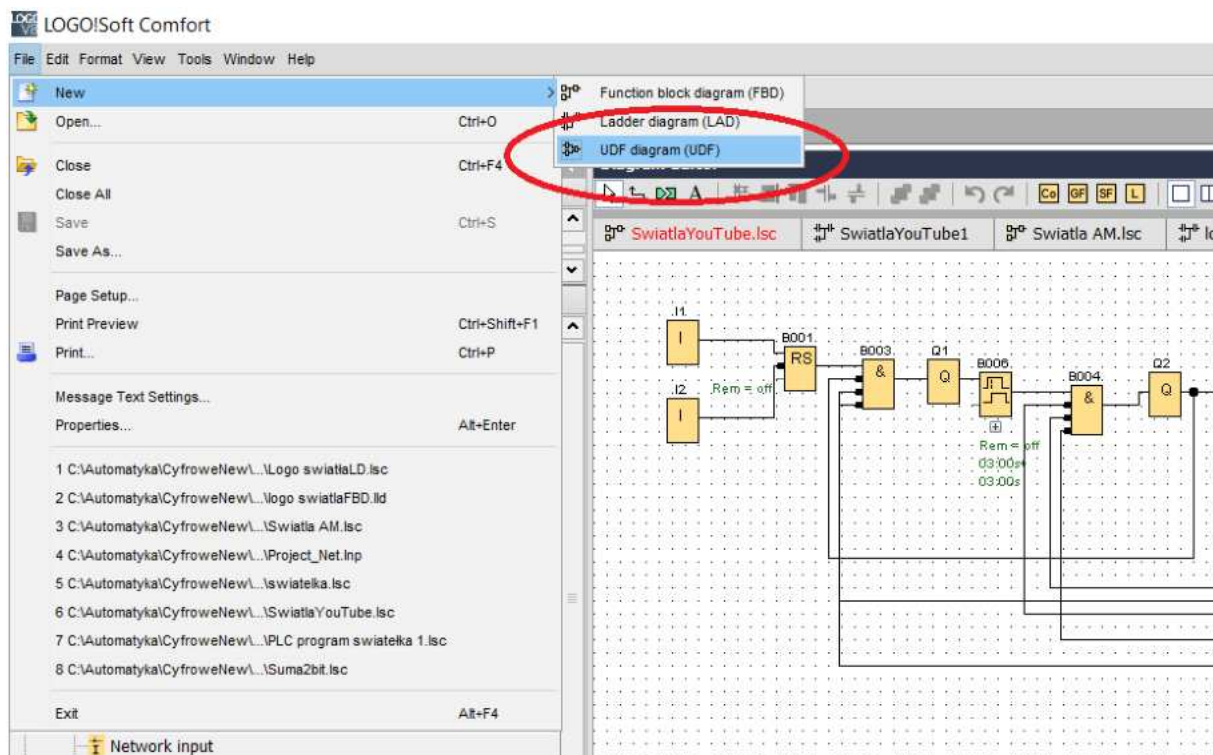
Lepsze rezultaty programowe otrzymamy wprowadzając programowanie przez komputer PC.

4.2 Programowanie LD sterownika w Logo!Soft Comfort v8 oraz tworzenie funkcji użytkownika - UDF

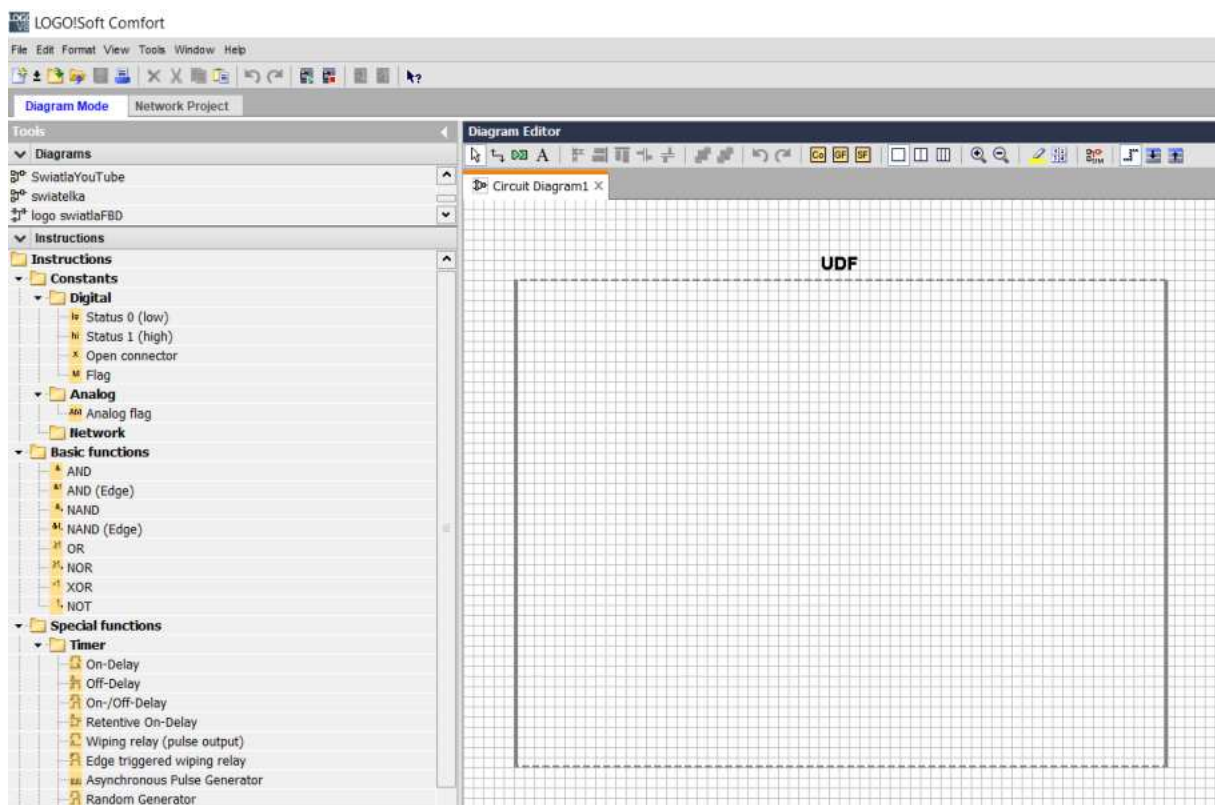
Przydatną cechą środowiska Logo!Soft Comfort jest możliwość tworzenia funkcji użytkownika, które dla odróżnienia od bloków FBD, noszą nazwę UDF od słów User Defined Function. Po umieszczeniu na schemacie, blok UDF przypomina zwykły blok FBD. Istotnym ograniczeniem dla bloków UDF, które trzeba brać pod uwagę, są:

- maksymalna liczba wejść wynosząca 8.
- maksymalna liczba wyjść wynosząca 4.

Tworzenie bloku UDF rozpoczyna się od wybrania z menu programu **File -> New -> UDF Diagram (UDF)**, jak pokazano na Rys. 4.5. Na ekranie zostanie otwarte okno (Rys. 4.6), w którym można umieścić niezbędne komponenty bloku. Wielkość okna można korygować z menu podręcznego **Properties**.



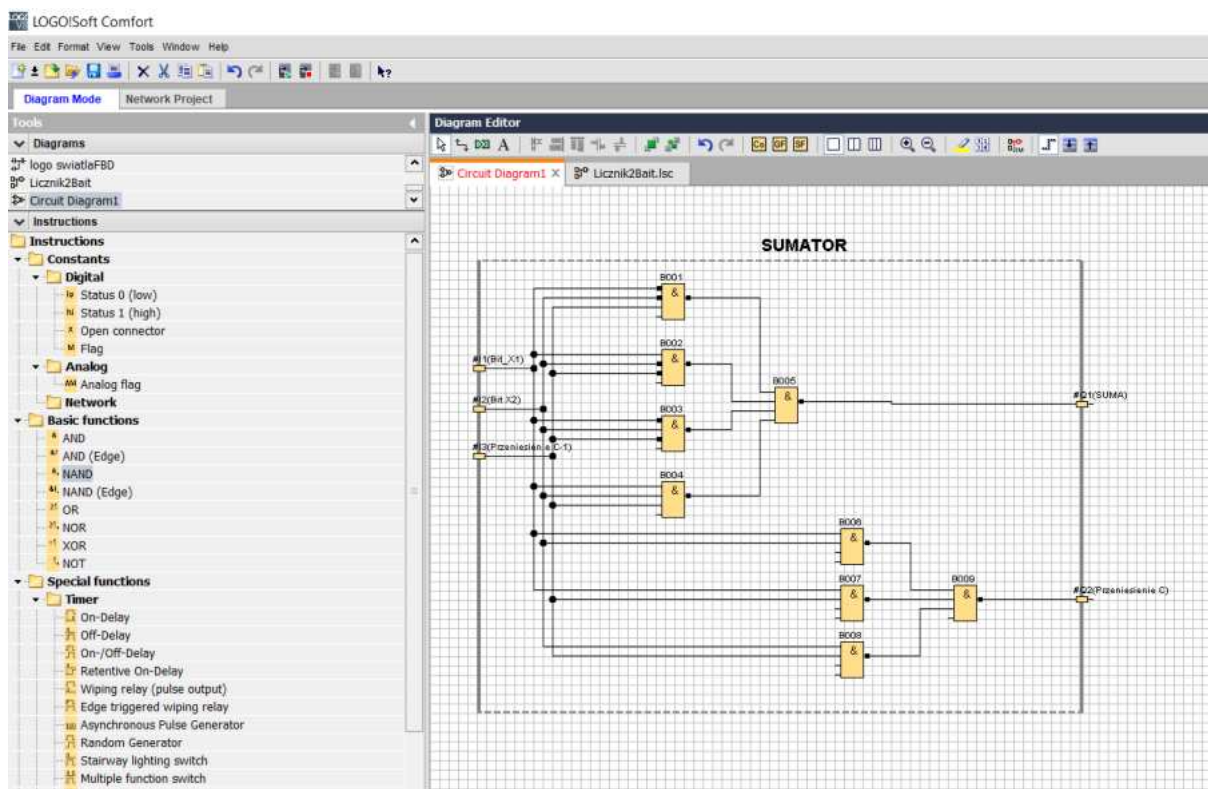
Rys. 4.5 Tworzenie nowego bloku UDF



Rys. 4.6 Okno edycji bloku UDF

Dla przykładu zaprojektowano blok sumatora dwóch bitów z przeniesieniem do kolejnego sumatora. Składa się on z funkcji logicznej sumy dwóch bitów oraz drugiego wyjścia funkcji logicznej bitu przeniesienia do kolejnego sumatora. Układ składa się wyłącznie z bramek NAND.

Schemat sumatora dwóch bitów złożonego z bramek podstawowych NAND i zminimalizowanego pokazano na Rys. 4.7. Sumator ma wejście bitu X1 oraz wejście bitu X2 i można go łączyć kaskadowo, konieczne jest wejście pożyczki (*BO – Borrow Output*) i wyjście przepełnienia (*CO – Carry Output*).



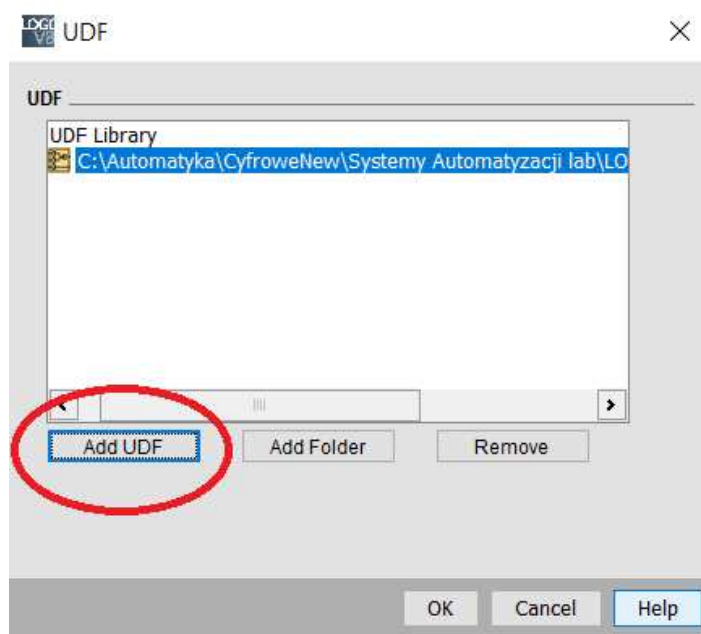
Rys. 4.7 Schemat sumatora dwu bitowego z przepełnieniem złożonego z bramek NAND

Zatrzymanie kursora nad symbolem wejścia lub wyjścia bloku FBD powoduje wyświetlenie skojarzonej z nim nazwy. Wejścia i wyjścia są ponumerowane i nie mają żadnych nazw opisowych. Aby je nadać, trzeba zatrzymać kursor na obszarze schematu UDF, nacisnąć prawy przycisk myszy i z menu podręcznego wybrać **Edit UDF Properties**. Zostanie wyświetlone okno pokazane na Rys. 4.8, w którym nie tylko można nazwać wejścia i wyjścia, ale również nazwę bloku UDF i zabezpieczyć go za pomocą hasła przed nieautoryzowaną edycją.

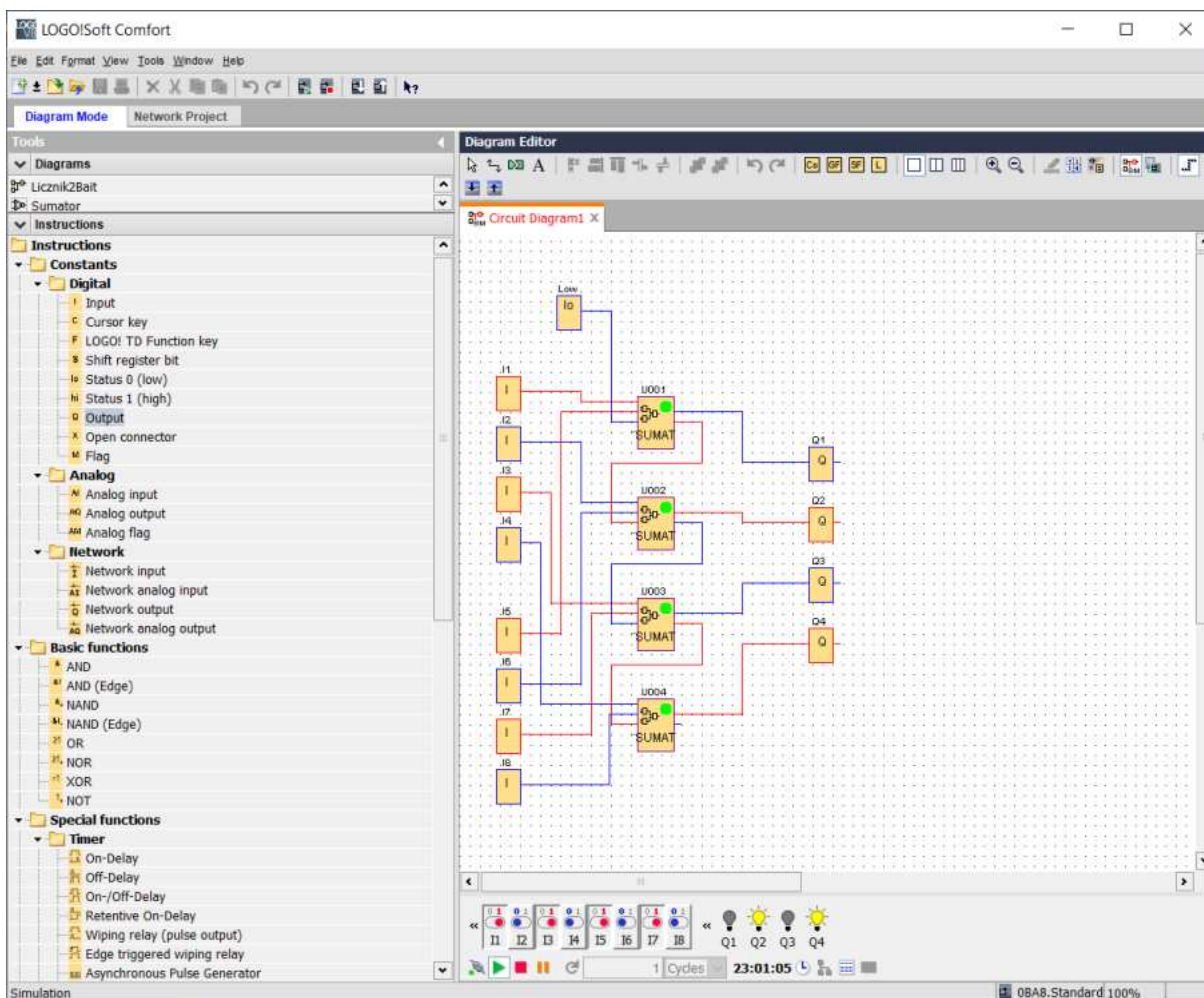
przykładzie sumatora pokazano na Rys. 4.11. Nazwa nadana blokowi UDF w oknie jego właściwości jest wyświetlana zarówno na drzewie, jak i na symbolu bloku na schemacie.



Rys. 4.9 Dodawanie bloku UDF do drzewa projektu z menu kontekstowego



Rys. 4.10 Dodanie bloku UDF do drzewa projektu



Rys. 4.11 Przykładowy schemat utworzony z użyciem bloku UDF – Sumator 4bit

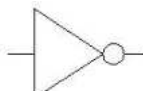
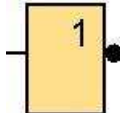
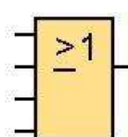
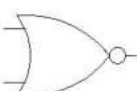
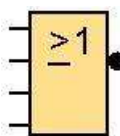
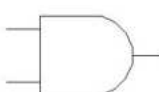
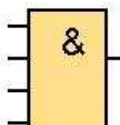
Bloki UDF mają zapisane numer wersji. Każde otwarcie bloku UDF do edycji, a następnie jego zapisanie na dysku powoduje otwarcie okna, w którym program proponuje nadanie numeru wersji. W ten sposób łatwo sprawdzić czy i kiedy blok był poprawiany oraz ile razy. Użytkownik ma możliwość dowiedzieć się, z którą aktualizacją ma do czynienia i czy jest ona zgodna z oczekiwaną. Jeśli blok UDF znajdujący się na schemacie FBD różni się od zapamiętanego w bibliotece, to na jego symbolu zostanie wyświetlony czerwony wykrzyknik. W przeciwnym wypadku jest wyświetlana zielona kropka jak pokazano na Rys. 4.11. W przypadku niezgodności można najechać myszką na symbol bloku, nacisnąć prawy przycisk i z menu podręcznego wybrać **Update UDF**. Jeśli chcemy poznać strukturę bloku, jak jest zaprojektowany, to w ten sam sposób z menu podręcznego należy wybrać **Expand UDF**.

Mimo iż blok UDF wygląda jak predefiniowany przez producenta pakietu blok FBD, to składa się on z pewnej liczby bloków FBD, która zmniejsza liczbę dostępnych bloków w programie. To znaczy, jeżeli do wykonania sumatora z tego przykładu użyto 9 bloków FBD, to za każdym razem, maksymalna liczba dostępnych bloków FBD wynosząca 400 zostanie pomniejszona nie o jeden, jak dla bloku UDF, ale o dziewięć.

Programowanie podstawowe sterownika LOGO! jest z użyciem bloków FBD, Tabela 4.1. Bloki FBD mają różną funkcjonalność, od wprowadzenia wejść, wyjść poprzez funkcje logiczne i matematyczne, timery po układy złożone jak liczniki, moduły RS, generatory sygnałów impulsowych i moduł regulatora. Każdy użytkownik ma wybór i może programować według bloków FBD lub zmieniać

na programowanie w języku drabinkowym, zasady programowania nieco się zmieniają, Tabela 4.2. Programator LOGO!Soft Comfort ma funkcję automatycznego tłumaczenia programu z jednej wersji na drugą, nie tłumaczy bloków UDF. Dużym udogodnieniem dla programisty jest symulacja pracy programu bez przesyłania go do sterownika.

Tabela 4.1 Podstawowe bramki logiczne, bloki FBD dostępne w sterowniku LOGO!

FUNKCJA LOGICZNA	ZAPIS MATEMATYCZNY FUNKCJI	SYMBOLE BRAMKI LOGICZNEJ		TABLICA PRAWDY															
X1, X2 – wejścia bramki i bloku FBD		Y – wyjście bramki i bloku FBD																	
NOT (negacja)	$Y = \overline{X1}$			<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0									
A	Y																		
0	1																		
1	0																		
OR (suma logiczna)	$Y = X1 + X2$			<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NOR (negacja sumy)	$Y = \overline{X1 + X2}$			<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
AND (iloczyn logiczny)	$Y = X1 \cdot X2$			<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Y																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	

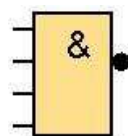
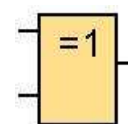
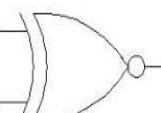
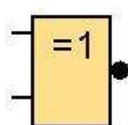
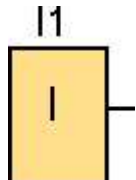
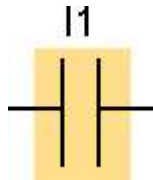
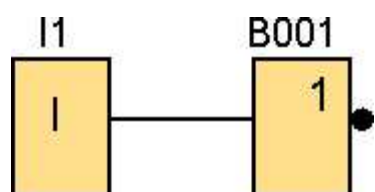
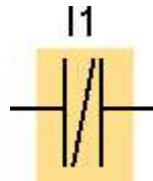
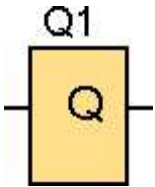
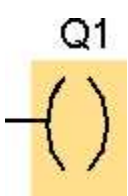
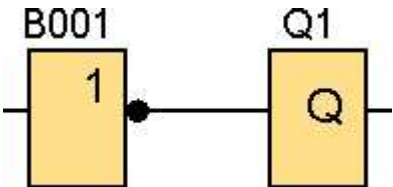
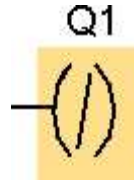
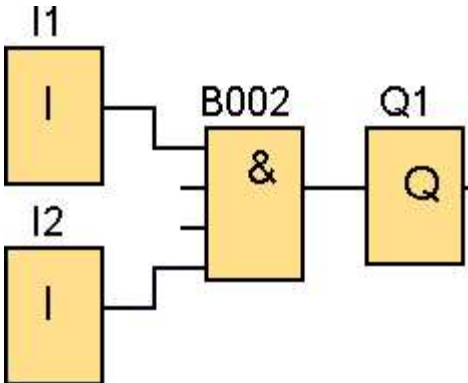
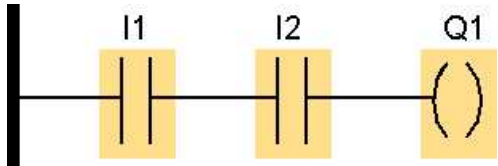
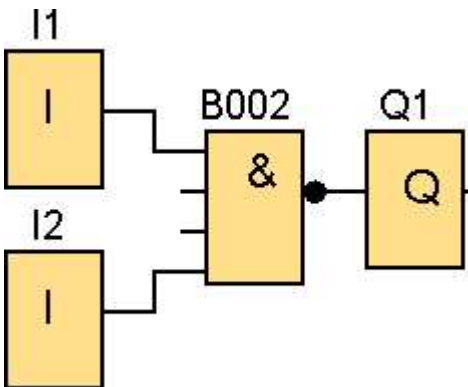
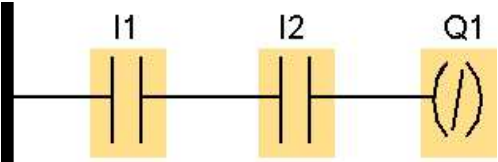
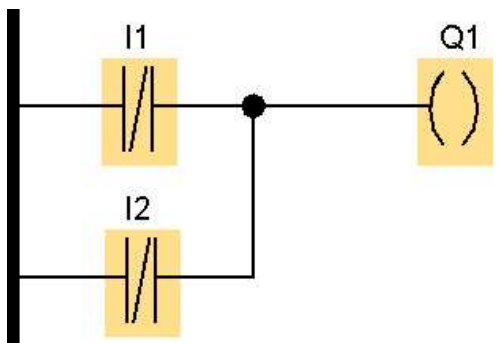
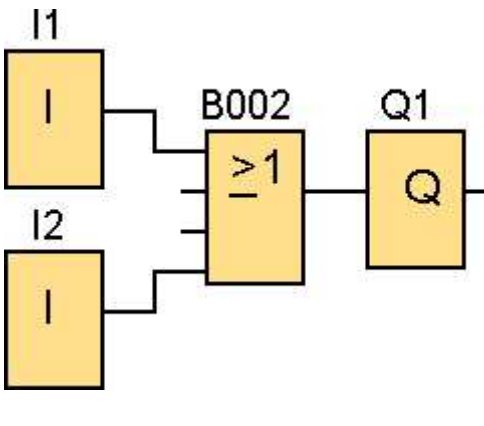
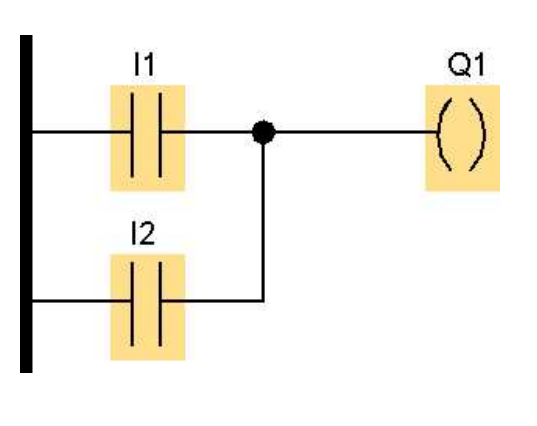
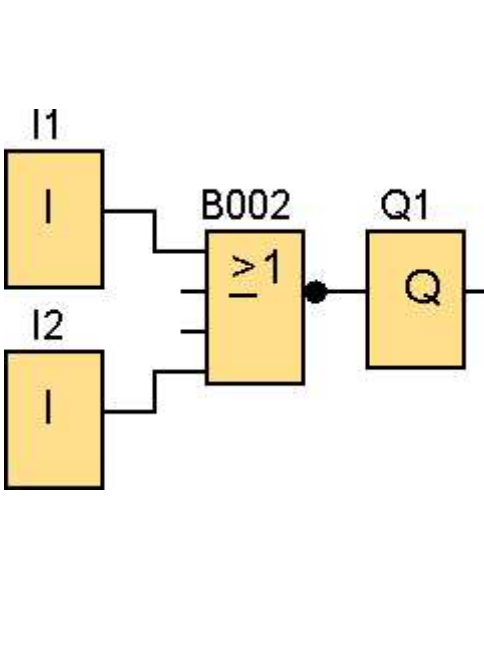
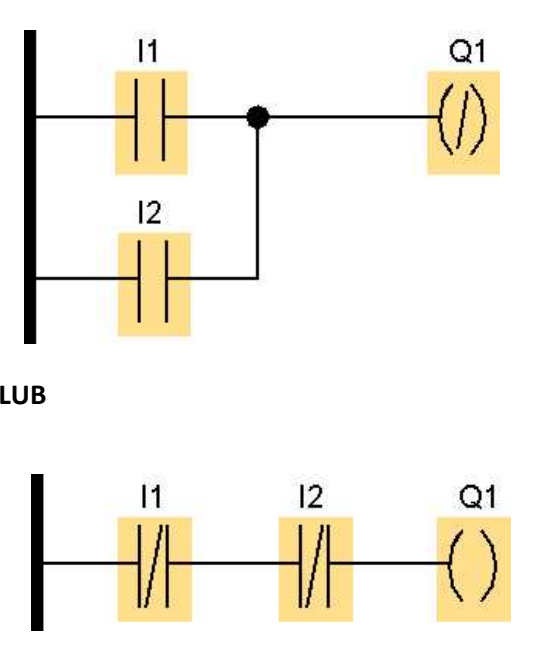
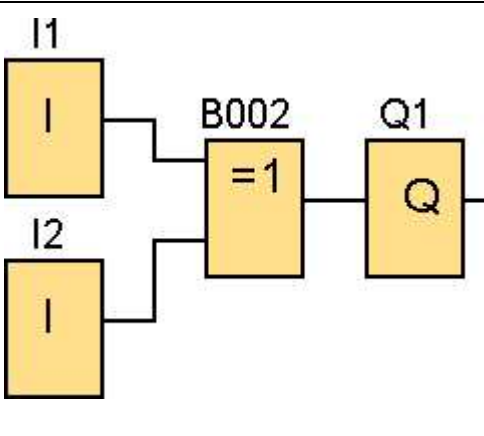
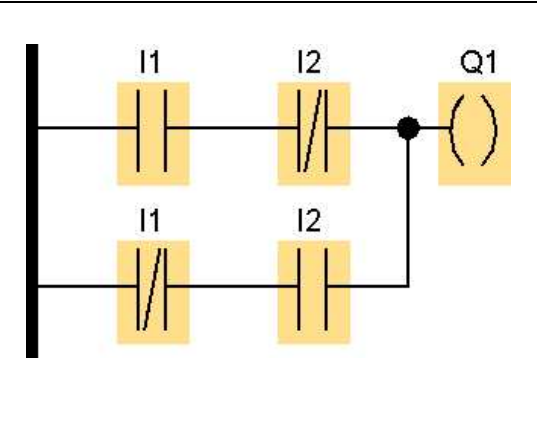
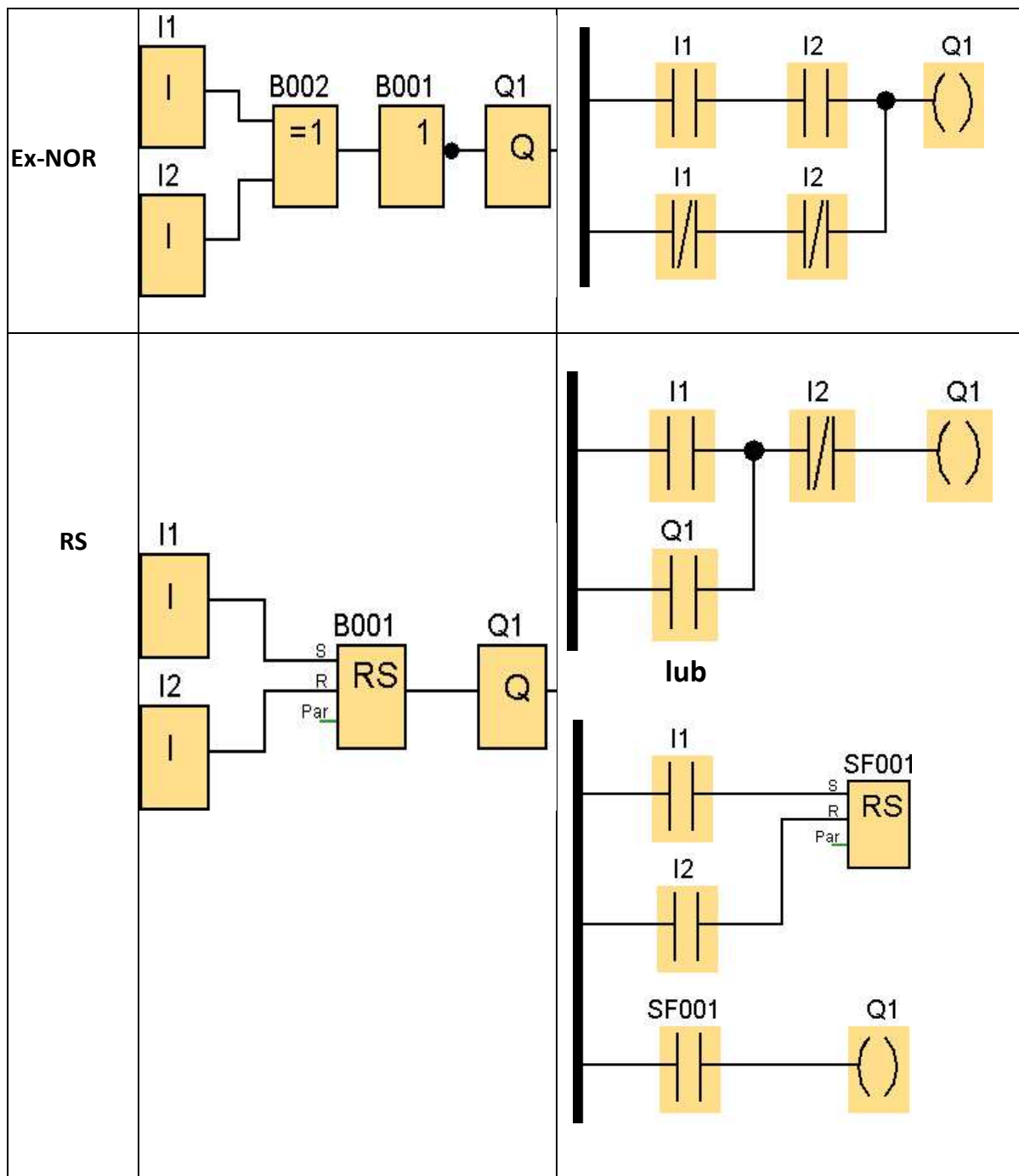
NAND (negacja iloczynu)	$Y = \overline{X1 \cdot X2}$			<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Y																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
XOR (alternatywa wykluczająca)	$Y = X1 \oplus X2$			<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
XNOR (negacja alternatywy)	$Y = \overline{X1 \oplus X2}$			<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	

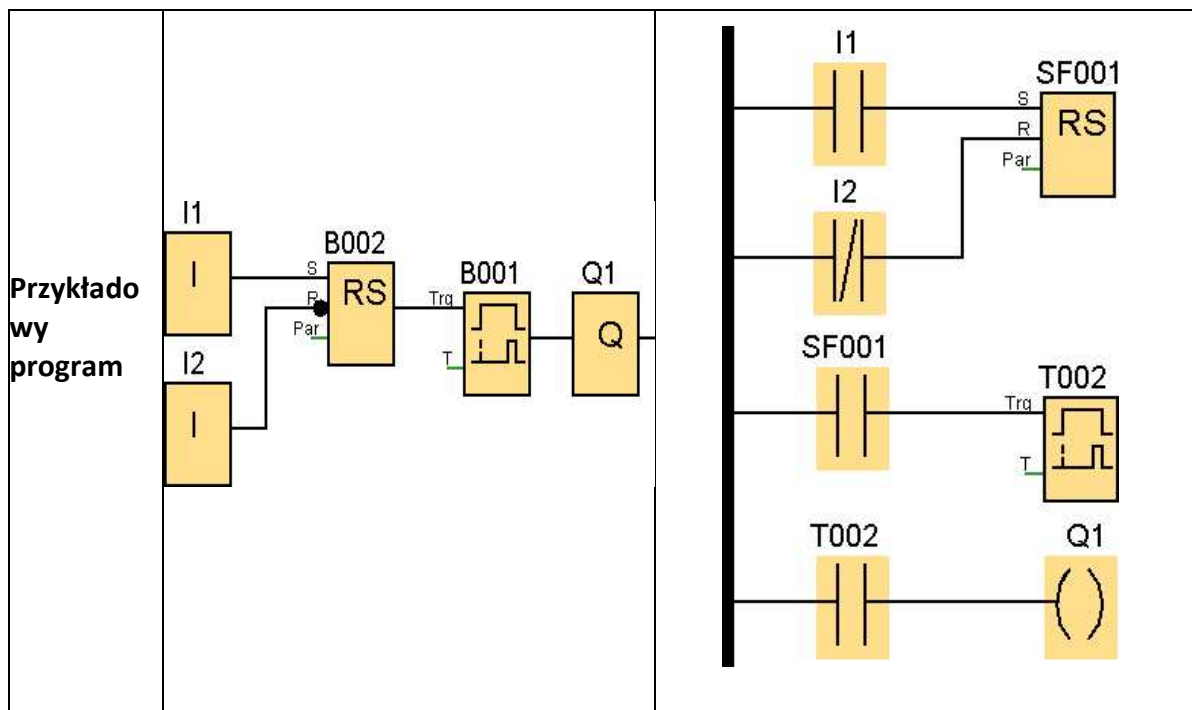
Tabela 4.2 Porównanie elementów FBD (function blok diagram – schemat bloków funkcjonalnych) z LD (ladder diagram – schemat w języku drabinkowym)

	FBD	LD
Wejście		
Wejście zanegowane		

Wyjście		
Wyjście zaneowane		
AND		
NAND		 LUB 

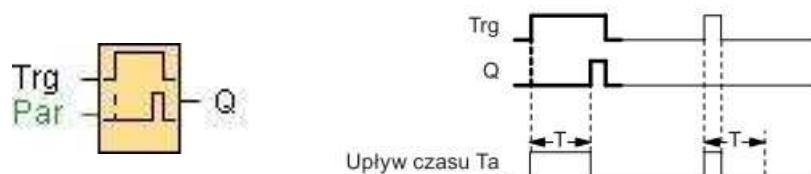
OR		
NOR		 LUB
Ex-OR (XOR)		





4.3 Timery w LOGO!

1. On-Delay – załączenie z opóźnieniem



Rys. 4.12 Timer On-Delay, symbol i wykres czasowy

Wyjście bloku Q jest załączane po upływie zadanego czasu, wprowadzonego jako parametr wejściowy.

Trg – sygnał 1 na wejściu Trg (Trigger) rozpoczyna odliczanie czasu.

Par – czas T po którym zostanie załączone wyjście.

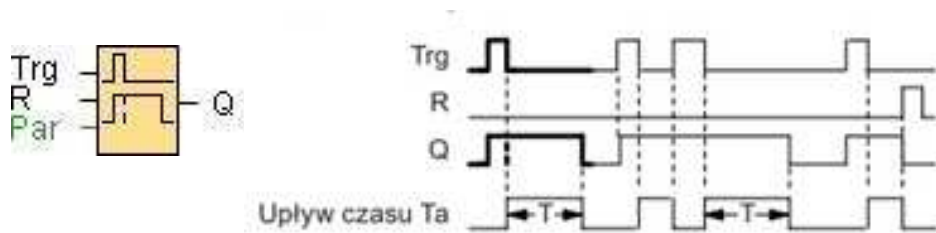
Q – wyjście zmienia stan z 0 na 1 po upływie czasu T, jeżeli wejście Trg jest nadal w stanie 1.

Opis działania

Zmiana stanu na wejściu Trg z 0 na 1 powoduje rozpoczęcie odliczania czasu T. Jeśli na wejściu Trg stan 1 trwa co najmniej tak długo, ile wynosi czas zadany T, po upływie czasu T na wyjściu pojawi się stan 1. Odliczanie zadanego czasu T ulega przerwaniu, jeżeli przed jego upływem wejście Trg powróci do stanu 0.

Wyjście jest zerowane, gdy wejście Trg powróci do stanu 0.

2. Off-Delay – opóźnienie wyłączenia



Rys. 4.13 Timer Off-Delay, symbol i wykres czasowy

Wyjście bloku Q przyjmuje stan 0 po upływie zadanego czasu.

Trg – start odliczania czasu następuje po ujemnym zboczach sygnału (przejście z 1 na 0)

R – sygnał 1 na wejściu R zeruje czas opóźnienia oraz wyjście.

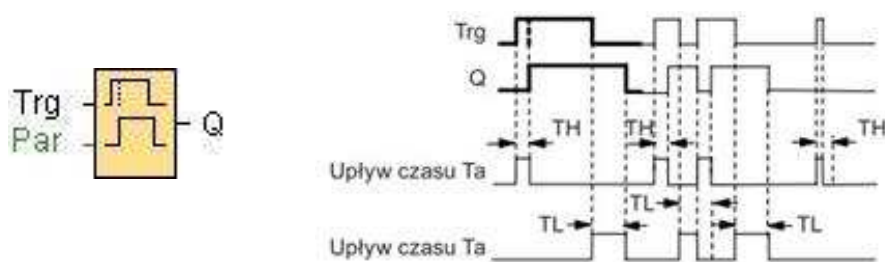
Par – wyjście zostaje wyłączone (przejście z 1 na 0) po upływie czasu opóźnienia T.

Q – wyjście jest załączane sygnałem na wejściu Trg, stan ten jest utrzymywany przez czas opóźnienia.

Opis działania

Stan wyjścia zmienia się na 1 równocześnie ze zmianą wejścia Trg do stanu 1. Odliczenie czasu wyłączenia T jest wznowiane przy zmianie stanu wejścia Trg ze stanu 1 na 0, przy czym wyjście pozostaje włączone, po upływie czasu opóźnienia wyjście Q powraca do stanu 0. Ustawienie wejścia R (Reset) powoduje wyzerowanie czasu opóźnienia oraz stanu wyjścia przed upływem czasu opóźnienia.

3. On/Off-Delay -opóźnienie włączenia/wyłączenia



Rys. 4.14 Timer On/Off-Delay, symbol i wykres czasowy

Wyjście bloku Q przyjmuje stan załączenia 1 po czasie opóźnienia włączenia, a następnie przechodzi do stanu wyłączenia 0 po upływie czasu wyłączenia.

Trg – przejście z 0 na 1 na wejściu Trg rozpoczyna odliczanie czasu opóźnienia włączenia T_H , następnie przejście z 1 na 0 rozpoczyna odliczanie czasu opóźnienia wyłączenia T_L .

Par – czas T_H po którym wyjście zostaje załączone, T_L po którym wyjście zostaje wyłączone.

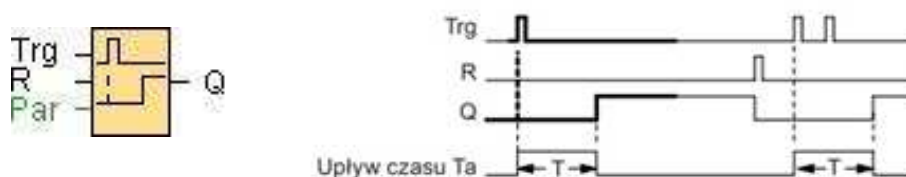
Q – wyjście jest włączone po upływie czasu T_H , jeżeli na wejściu Trg jest utrzymywany stan 1; wyjście jest wyłączone po upływie czasu T_L , jeżeli na wejściu Trg nie wystąpił ponownie stan 1.

Opis działania

Odliczanie czasu T_H jest inicjowane przez zmianę z 0 na 1 sygnału na wejściu Trg. Jeżeli wejście Trg pozostaje w stanie 1 co najmniej przez czas T_H , to wyjście przyjmuje stan 1 po upływie czasu T_H , czas opóźnienia jest zerowany gdy sygnał na wejściu Trg powraca do stanu 0 przed upływem czasu T_H .

Przejście z 1 na 0 na wejściu Trg inicjuje odliczanie czasu T_L . Jeżeli wejście Trg pozostaje w stanie 0 co najmniej przez czas T_L , to wyjście przyjmuje stan 0 po upływie czasu T_L , czas opóźnienia jest zerowany gdy sygnał na wejściu Trg zmienia się na 1 przed upływem czasu T_H .

4. Retentive On-Delay - opóźnione załączenie z podtrzymaniem



Rys. 4.15 Timer Retentive On-Delay, symbol i wykres czasowy

Sygnał na wejściu rozpoczyna odliczanie zadanego czasu, włączenie wyjścia Q następuje po upływie tego czasu.

Trg – zmiana sygnału wejścia Trg z 0 na 1 inicjuje odliczanie opóźnienia załączenia.

R – sygnał reset na wejściu R, który zeruje czas opóźnienia oraz wyjście.

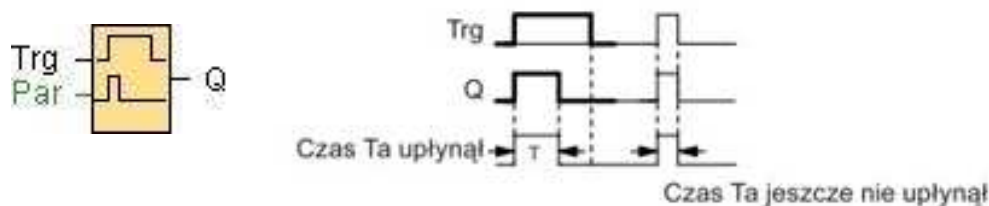
Par – czas T opóźnienia, po którym będzie załączone wyjście.

Q – wyjście zostanie załączone po upływie czasu opóźnienia niezależnie od stanu wejścia Trg.

Opis działania

Zmiana stanu na wejściu Trg z 0 na 1 powoduje rozpoczęcie odliczania czasu opóźnienia T_a . Stan wyjścia Q zmieni się w chwili upływu czasu opóźnienia, kolejne zmiany sygnału na wejściu Trg nie zatrzymują odliczania. Stan wyjścia oraz czas opóźnienia są zerowane dopiero po zmianie stanu na 1 na wejściu reset R.

5. Wiping relay (pulse output) - przekaźnik czasowy z wyjściem impulsowym



Rys. 4.16 Wiping relay (pulse output), symbol i wykres czasowy

Sygnał wejściowy powoduje zainicjowaniu na wyjściu sygnału o określonym czasie trwania.

Trg – sygnał 1 na wejściu Trg (Trigger) rozpoczyna załączenia wyjścia odliczanie czasu do wyłączenia.

Par – wyjście zostaje wyłączone po upływie czasu T, sygnał wyjściowy zmienia stan z 1 na 0.

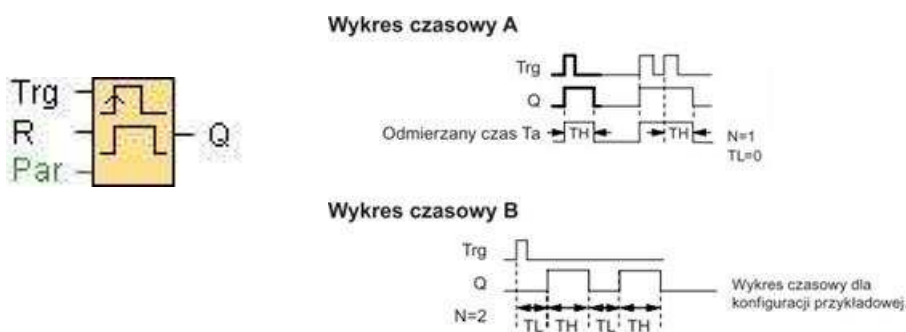
Q – wyjście przyjmuje stan 1, gdy na wejściu Trg pojawi się sygnał 1 i pozostaje wzbudzone przez zadany czas T, chyba że wcześniej zaniknie sygnał na wejściu.

Opis działania

Zmiana stanu na wejściu Trg z 0 na 1 powoduje ustawienie wyjścia w stan 1 i rozpoczęcie odliczania czasu opóźnienia podczas którego wyjście pozostaje włączone. Stan wyjścia zmienia się na 0 w chwili, gdy czas opóźnienia osiągnie zadana wartość parametru T. Jeżeli przed upływem zadanego

czasu opóźnienia nastąpi zmiana sygnału na wejściu Trg z 1 na 0, to stan wyjścia zmienia się z 1 na 0 i zeruje się czas opóźnienia.

6. Edge triggered wiping relay - przekaźnik czasowy wyzwalany zboczem



Rys. 4.17 Edge triggered wiping relay, symbol i wykres czasowy

Sygnał na wejściu powoduje wygenerowanie na wyjściu, po określonym czasie opóźnienia, zadanej liczby impulsów o określonym stosunku czasu trwania impulsu do przerwy między impulsami z możliwością wznowienia.

Trg – sygnał 1 na wejściu Trg (Trigger) rozpoczyna odliczanie czasu w przekaźniku czasowym wyzwalanym zboczem.

R – sygnał reset na wejściu R zeruje licznik czasu opóźnienia oraz wyjście.

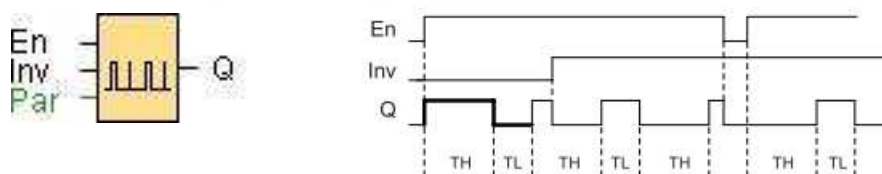
Par – parametr konfiguruje odstęp między impulsami T_L oraz szerokość impulsu T_H . N określa ilość cykli impuls/pauza T_L/T_H . Zakres wartości parametru N jest od 1 do 9.

Q – wyjście Q jest włączane po upływie czasu opóźnienia T_L , a następnie zerowane po upływie czasu T_H , (wykres czasowy B). Dla parametru $T_L = 0$ wyjście jest załączane natychmiast (wykres czasowy A).

Opis działania

Zmiana stanu na wejściu Trg z 0 na 1 powoduje rozpoczęcie odliczania czasu T_L (Time Low). Po upływie czasu T_L stan wyjścia Q zmienia się na 1 na czas T_H (Time High). Jeśli przed upływem czasu $T_L + T_H$ na wejściu Trg pojawi się ponowny impuls wyzwalający, zmiana stanu z 0 na 1, licznik czasu opóźnienia zostaje wyzerowany, a cykl impuls/przerwa rozpoczyna się od początku. Cykl powtarza się N razy.

7. Asynchronous Pulse Generator - asynchroniczny generator impulsów



Rys. 4.18 Asynchronous Pulse Generator, symbol i wykres czasowy

Postać sygnału wyjściowego jest konfigurowana za pośrednictwem czasu trwania impulsu i przerwy między impulsami.

En – wejście zezwalające służy do ustawiania i zerowania wyjścia generatora impulsów asynchronicznych.

Inv – wejście to służy do wprowadzania inwersji sygnału wyjściowego generatora impulsów asynchronicznych.

Par – konfigurujemy wartość szerokości impulsów T_H oraz odstępu między impulsami T_L .

Q – wyjście jest włączane i wyłączane cyklicznie zgodnie z zadanymi wartościami czasu T_H i T_L .

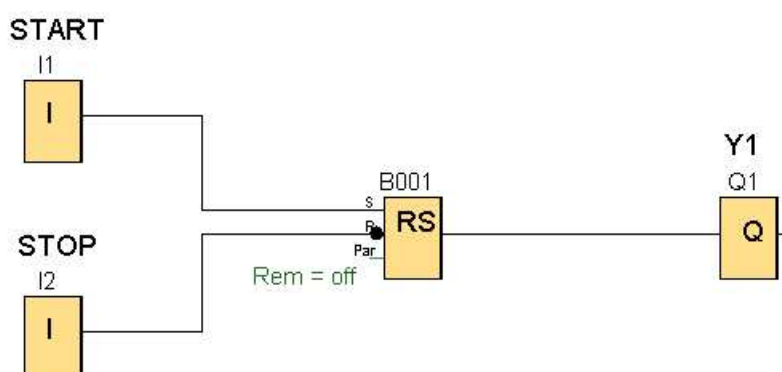
Opis działania

Wyjście jest konfigurowane przez wartość czasu trwania impulsów T_H (Time high) oraz przerwy między impulsami T_L (Time Low). Wejście Inv służy do invertowania wartości sygnału wyjściowego, pod warunkiem, że blok jest aktywowany przez sygnał 1 podany na wejście En.

4.4 Przykłady zastosowania podstawowych bloków funkcyjnych sterownika Siemens LOGO!

1. Podtrzymanie (zapamiętanie naciśnięcia przycisku monostabilnego)

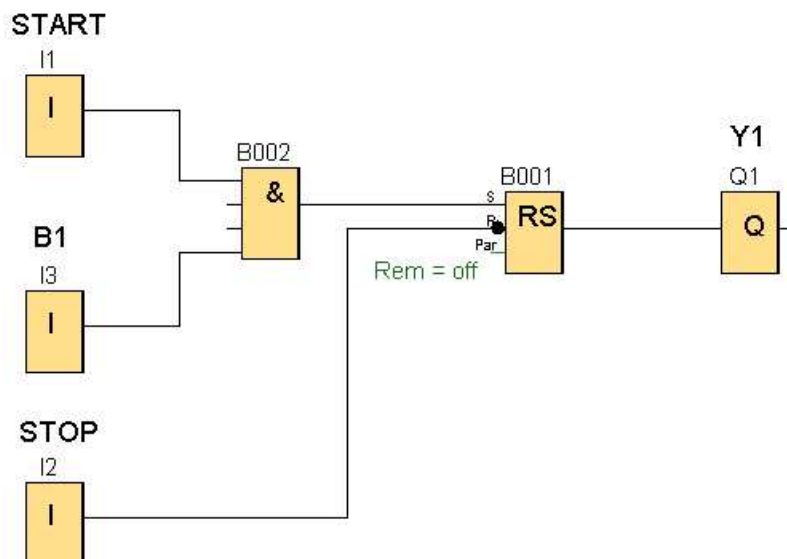
Siłownik jednostronnego działania wysuwa się (zasilana cewka elektrozaworu Y1) po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) START. Wsuniecie siłownika następuje po krótkotrwałym naciśnięciu przycisku monostabilnego (NC) STOP, Rys. 4.19.



Rys. 4.19 Realizacja programu – podtrzymanie wyjścia

2. Warunkowe uruchomienie

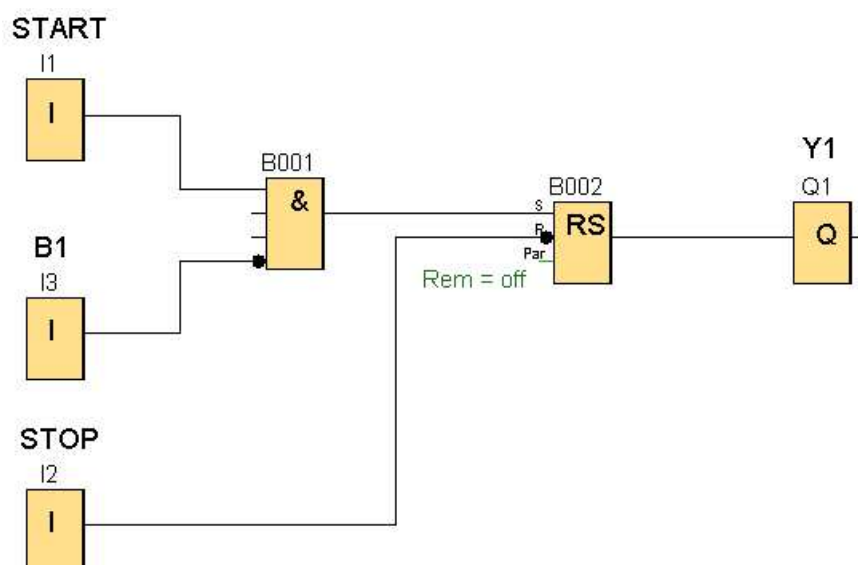
Siłownik jednostronnego działania wysuwa się (zasilana cewka elektrozaworu Y1) po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) START pod warunkiem, że siłownik jest całkowicie wsunięty, co jest wykrywane przez czujnik B1. Wsuniecie siłownika następuje po krótkotrwałym naciśnięciu przycisku monostabilnego (NC) STOP, Rys. 4.20.



Rys. 4.20 Realizacja programu – uruchomienie warunkowe

3. Ignorowanie polecenia „START”

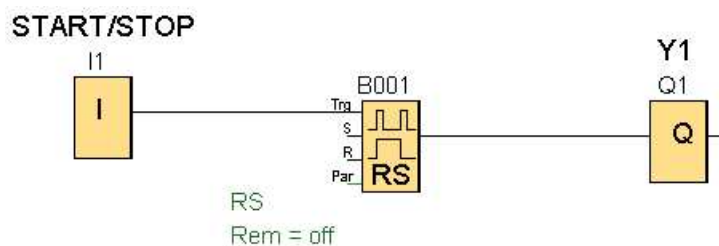
Siłownik jednostronnego działania wysuwa się (zasilana cewka elektrozaworu Y1) po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) START. Wciskanie przycisku START jest ignorowane, jeżeli czujnik B1 wykrywa w polu roboczym siłownika obecność człowieka. Wsuniecie siłownika następuje po krótkotrwałym naciśnięciu przycisku monostabilnego (NC) STOP, Rys. 4.21.



Rys. 4.21 Realizacja programu – ignorowanie polecenia 'START'

4. Załączanie i wyłączanie tym samym przyciskiem

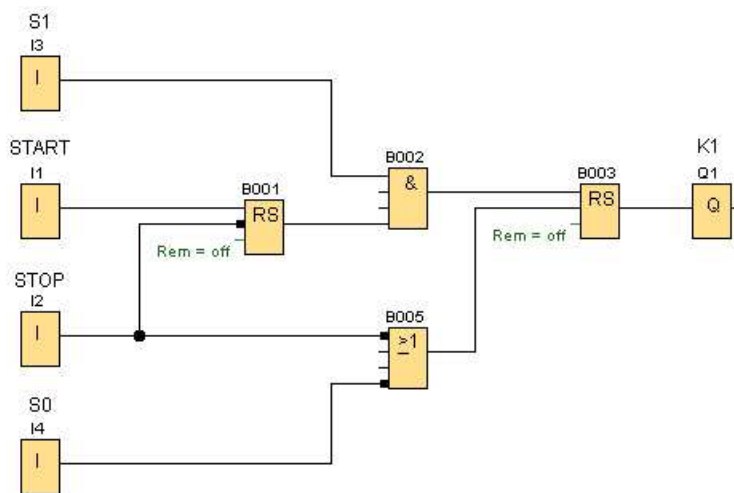
Siłownik jednostronnego działania wysuwa się (zasilana cewka elektrozaworu Y1) po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) START/STOP. Kolejne krótkotrwałe naciśnięcie przycisku START/STOP powoduje wsunięcie siłownika, Rys. 4.22.



Rys. 4.22 Realizacja programu – załączanie i wyłączanie pojedynczym przyciskiem

5. Załączenie urządzenia możliwe tylko w trybie PRACA

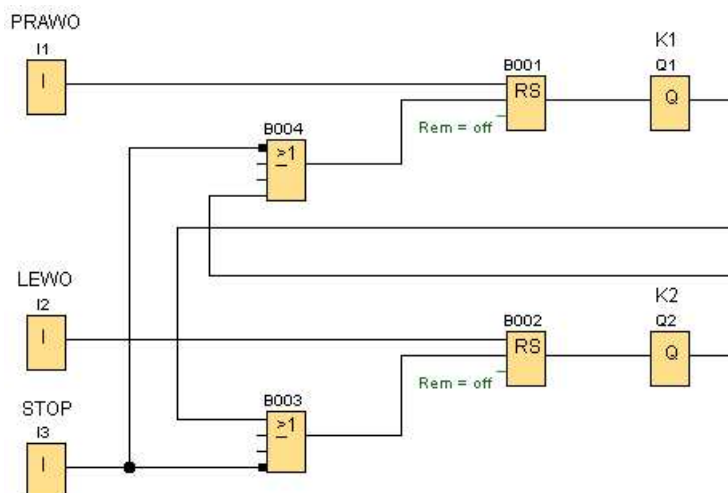
Mechanizm sterowniczy znajduje się w trybie PRACA po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) START. Wyłączenie trybu PRACA możliwe jest przez krótkotrwałe naciśnięcie przycisku monostabilnego (NC) STOP. Silnik uruchamia się (zasilana cewka stycznika K1) po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) S1 tylko wtedy, gdy mechanizm sterowniczy znajduje się w trybie PRACA. Wyłączenie silnika następuje po krótkotrwałym naciśnięciu przycisku monostabilnego (NC) S0 lub po wyłączeniu trybu PRACA, Rys. 4.23.



Rys. 4.23 Realizacja programu – załączenie w trybie praca

6. Blokada przeciwwzruciowa LEWO/PRAWO

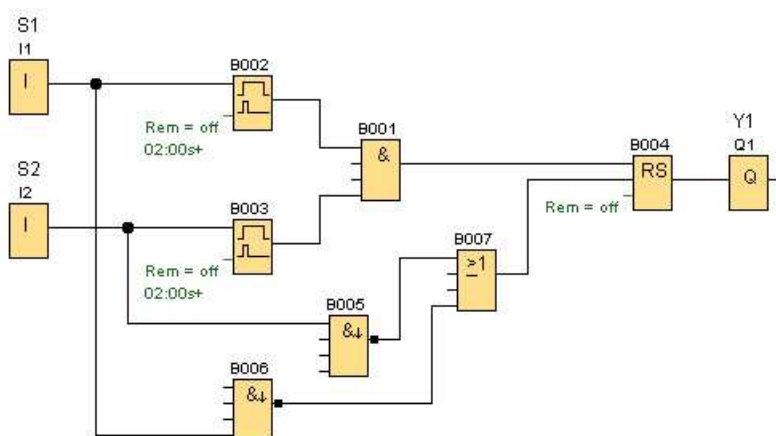
Prawe obroty silnika załącza się (zasilana cewka stycznika K1) przez krótkotrwałe naciśnięcie przycisku monostabilnego (NO) PRAWO, a lewe obroty silnika załącza się (zasilana cewka stycznika K2) przez krótkotrwałe naciśnięcie przycisku monostabilnego (NO) LEWO. Silnik wyłącza się po krótkotrwałym naciśnięciu przycisku monostabilnego (NC) STOP. Silnik posiada blokadę zabezpieczającą przed jednoczesnym zasilaniem cewek styczników K1 i K2, Rys. 4.24.



Rys. 4.24 Realizacja programu – blokada przeciwwzarciowa LEWO/PRAWO

7. Oburęczne załączenie

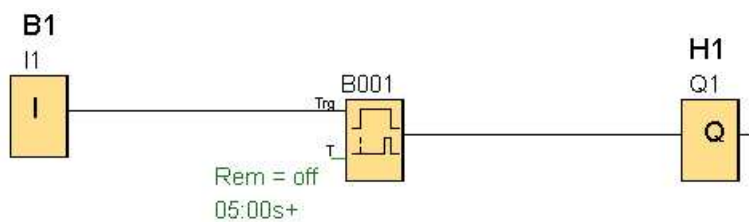
Prasa hydrauliczna załącza się (zasilana cewka elektrozaworu Y1) po jednoczesnym naciśnięciu przycisków monostabilnych (NO) S1 i S2. Dopuszczalny czas opóźnienia w przyciśnięciu przycisków wynosi 2 sekundy. Zwolnienie któregośkolwiek z przycisków po uruchomieniu prasy przerywa jej pracę, Rys. 4.25.



Rys. 4.25 Realizacja programu – dwuręczne załączenie

8. Opóźnienie załączenia 1

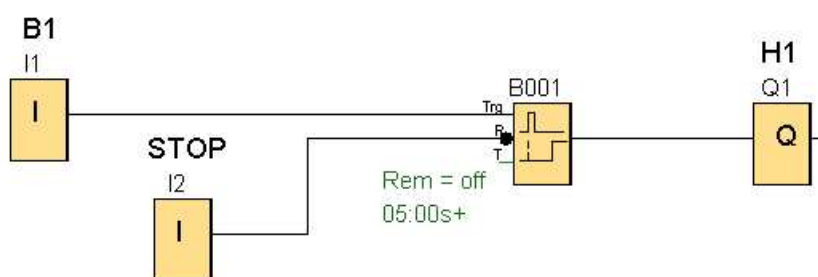
Pojawienie się obiektu przed czujnikiem B1 powoduje załączenie po upływie 5 sekund alarmu (lampki sygnalizacyjnej H1). Alarm nie załączy się jeżeli obiekt zniknie przed upływem 5 sekund. Załączony alarm wyłączy się samoczynnie po zniknięciu obiektu, Rys. 4.26.



Rys. 4.26 Realizacja programu – opóźnienie załączenia 1

9. Opóźnienie załączenia 2

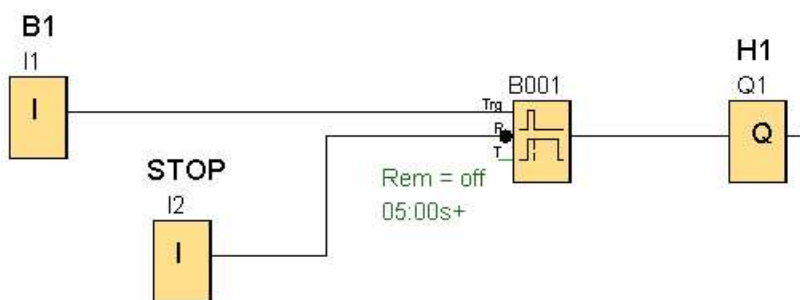
Pojawienie się obiektu przed czujnikiem B1 powoduje załączenie po upływie 5 sekund alarmu (lampki sygnalizacyjnej H1). Alarm załączy się nawet jeżeli obiekt zniknie przed upływem 5 sekund. Alarm można wyłączyć tylko poprzez krótkotrwałe naciśnięcie przycisku monostabilnego (NC) STOP, Rys. 4.27.



Rys. 4.27 Realizacja programu – opóźnienie załączenia 2

10. Opóźnienie wyłączenia

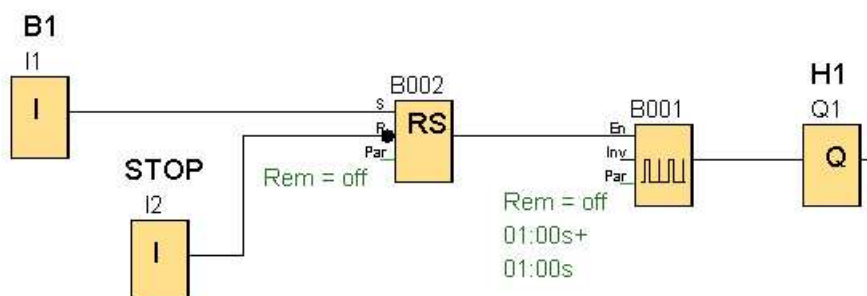
Pojawienie się obiektu przed czujnikiem B1 powoduje natychmiastowe załączenie alarmu (lampki sygnalizacyjnej H1). Alarm wyłączy się samoczynnie dopiero po 5 sekundach od zniknięcia obiektu. Alarm można też wyłączyć wcześniej poprzez krótkotrwałe naciśnięcie przycisku monostabilnego (NC) STOP, Rys. 4.28.



Rys. 4.28 Realizacja programu – opóźnienie wyłączenia

11. Miganie lampki sygnalizacyjnej

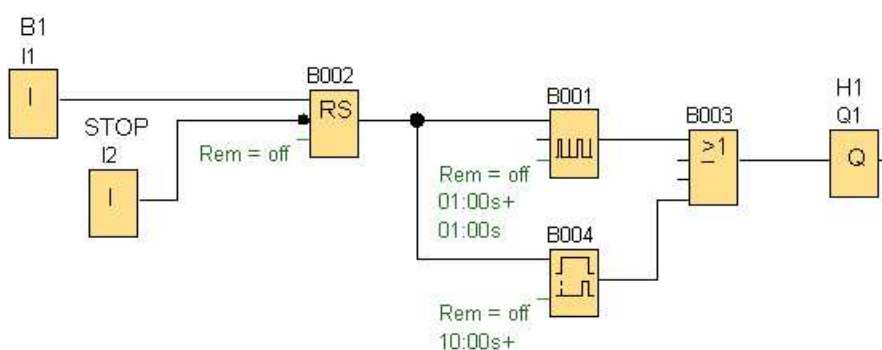
Pojawienie się obiektu przed czujnikiem B1 powoduje natychmiastowe załączenie alarmu - miga lampka sygnalizacyjna H1. Alarm pozostaje załączony nawet po zniknięciu obiektu. Alarm można wyłączyć tylko poprzez krótkotrwałe naciśnięcie przycisku monostabilnego (NC) STOP, Rys. 4.29.



Rys. 4.29 Realizacja programu – miganie lampki sygnalizacyjnej

12. Miganie i świecenie ciągłe lampki sygnalizacyjnej

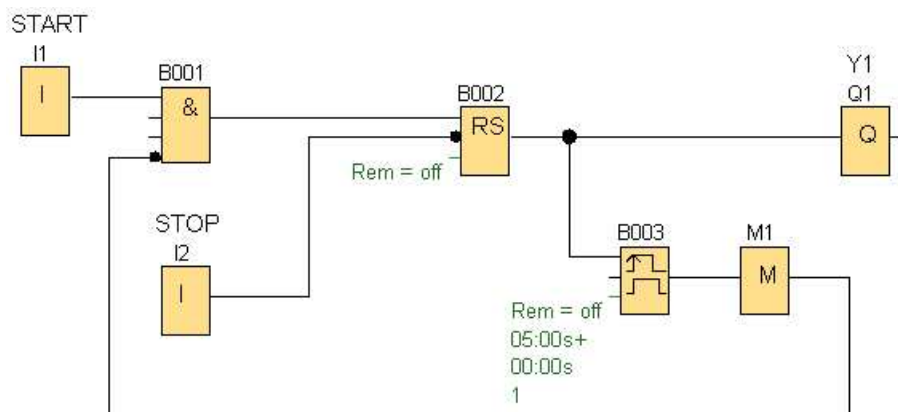
Pojawienie się obiektu przed czujnikiem B1 powoduje natychmiastowe załączenie alarmu - miga lampka sygnalizacyjna H1. Po 10 sekundach od załączenia alarmu lampka świeci światłem ciągłym. Alarm pozostaje załączony nawet po zniknięciu obiektu. Alarm można wyłączyć tylko poprzez krótkotrwałe naciśnięcie przycisku monostabilnego (NC) STOP, Rys. 4.30.



Rys. 4.30 Realizacja programu – miganie i świecenie lampki sygnalizacyjnej

13. Chwilowa blokada powtórnego załączenia 1

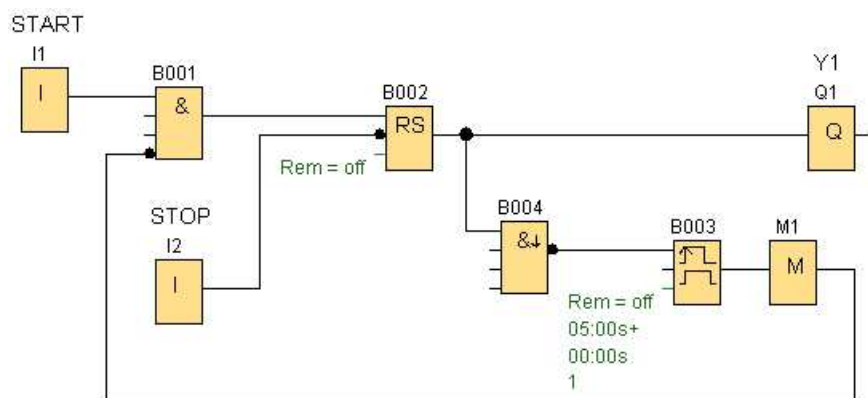
Siłownik jednostronnego działania wysuwa się (zasilana cewka elektrozaworu Y1) po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) START. Wsuniecie siłownika następuje po krótkotrwałym naciśnięciu przycisku monostabilnego (NC) STOP. Ponowne wysunięcie siłownika możliwe jest dopiero po 5 sekundach od rozpoczęcia wysuwania siłownika w poprzednim cyklu pracy, Rys. 4.31.



Rys. 4.31 Realizacja programu – chwilowa blokada powtórnego załączenia 1

14. Chwilowa blokada powtórnego załączenia 2

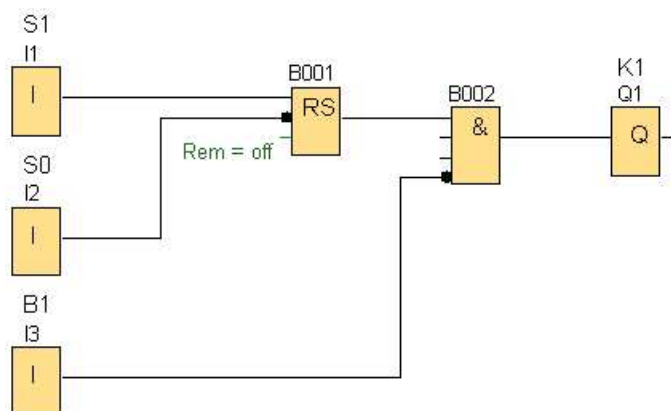
Siłownik jednostronnego działania wysuwa się (zasilana cewka elektrozaworu Y1) po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) START. Wsuniecie siłownika następuje po krótkotrwałym naciśnięciu przycisku monostabilnego (NC) STOP. Ponowne wysunięcie siłownika możliwe jest dopiero po 5 sekundach od rozpoczęcia wsuwania siłownika w poprzednim cyklu pracy, Rys. 4.32.



Rys. 4.32 Realizacja programu – chwilowa blokada powtórnego załączenia 2

15. Wstrzymanie pracy urządzenia

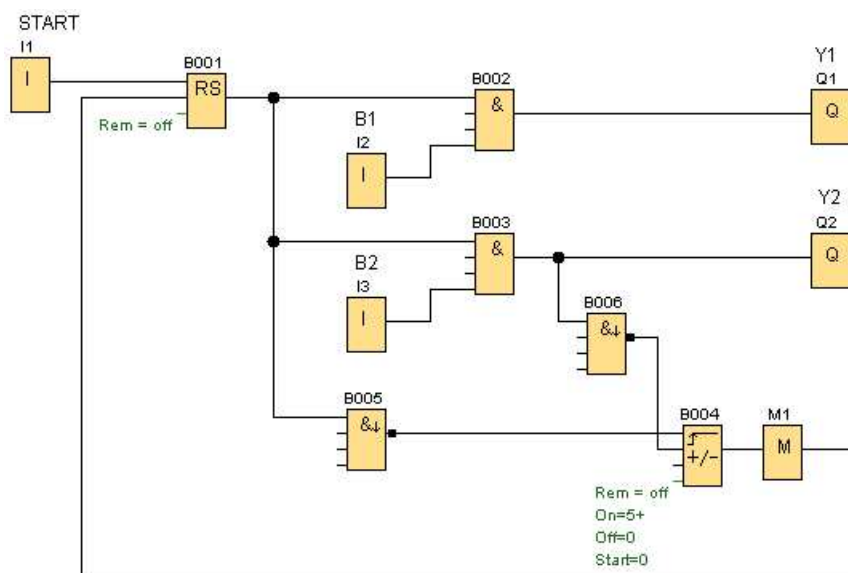
Silnik uruchamia się (zasilana cewka stycznika K1) po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) S1. Po wykryciu przez czujnik B1 obiektu w polu roboczym silnika, praca silnika zostaje wstrzymana. Silnik uruchomi się samoczynnie po opuszczeniu pola roboczego przez obiekt. Trwałe wyłączenie silnika następuje po krótkotrwałym naciśnięciu przycisku monostabilnego (NC) S0, Rys. 4.33.



Rys. 4.33 Realizacja programu – wstrzymanie pracy urządzenia

16. Zliczanie zdarzeń

Po krótkotrwałym naciśnięciu przycisku monostabilnego (NO) START siłownik dwustronnego działania przechodzi do trybu PRACA, w którym cyklicznie wysuwa się (zasilana cewka elektrozaworu Y1) i wsuwa się (zasilana cewka elektrozaworu Y2). Stany pełnego wsunięcia i wysunięcia tłoczyska siłownika wykrywane są czujnikami magnetycznymi B1 i B2. Po 5 cyklach pracy siłownika następuje wyłączenie trybu PRACA, zatrzymanie siłownika w pozycji wsuniętej i zresetowanie licznika zliczającego cykle pracy siłownika, Rys. 4.34.



Rys. 4.34 Realizacja programu – zliczanie zdarzeń

17. Budowa sumatora dwóch bitów z przeniesieniem

Rozwiązania rozpoczyna się poprzez opracowanie tabeli prawdy dla funkcji logicznej sumy dwóch bitów L1 i L2 i trzeciego bitu z przeniesienia C. Następnie opracowujemy tablice Karnaugh'a.

Tabela 4.3 Tabela prawdy dla sumatora dwóch bitów z przeniesieniem

Lp	I1	I2	C	Q1	C1
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

Tabela 4.4 Tablica Karnaugh'a dla sygnału Q1

C	/ I1 I2	00	01	11	10
0		0	1	0	1
1		1	0	1	0

Równanie odpowiedzi Q1 dla sumatora dwóch bitów z przeniesieniem jest następujące:

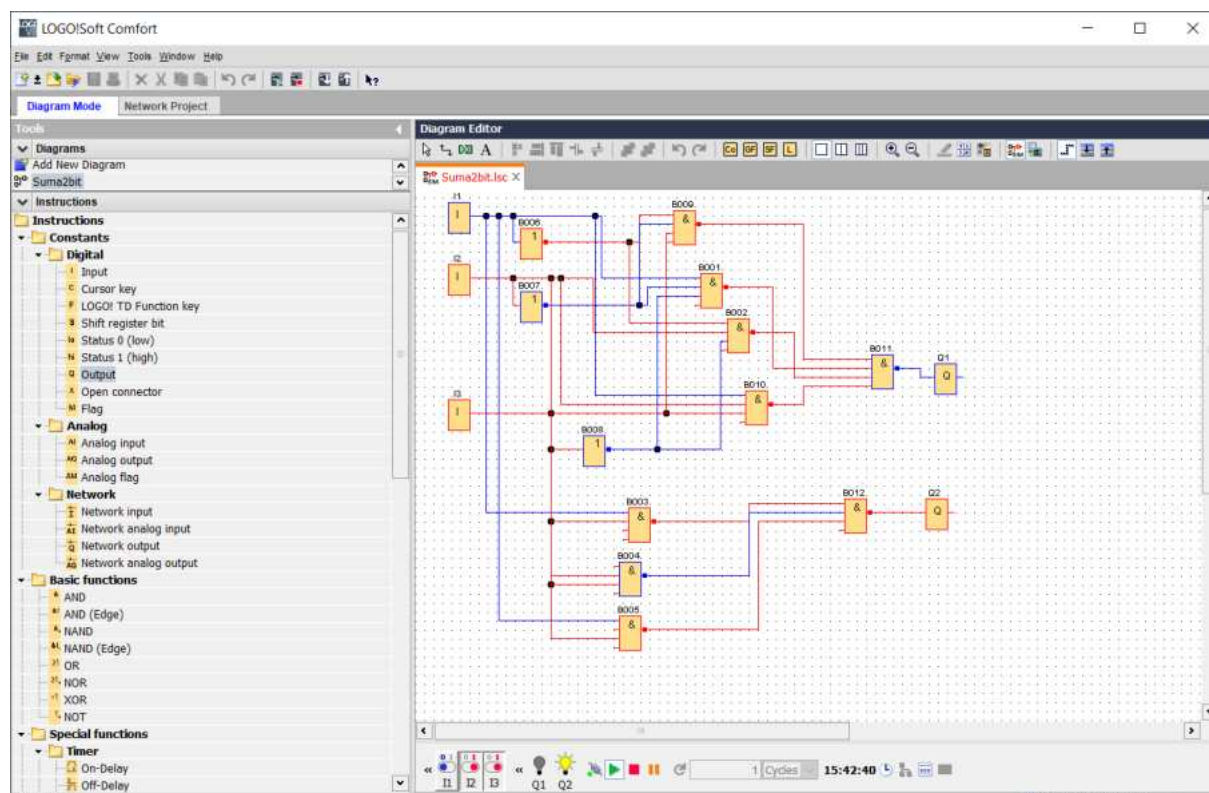
$$Q1 = \bar{I1} \cdot \bar{I2} \cdot C + \bar{I1} \cdot I2 \cdot \bar{C} + I1 \cdot \bar{I2} \cdot \bar{C} + I1 \cdot I2 \cdot C$$

Tabela 4.5 Tablica Karnaugh'a dla sygnału C1

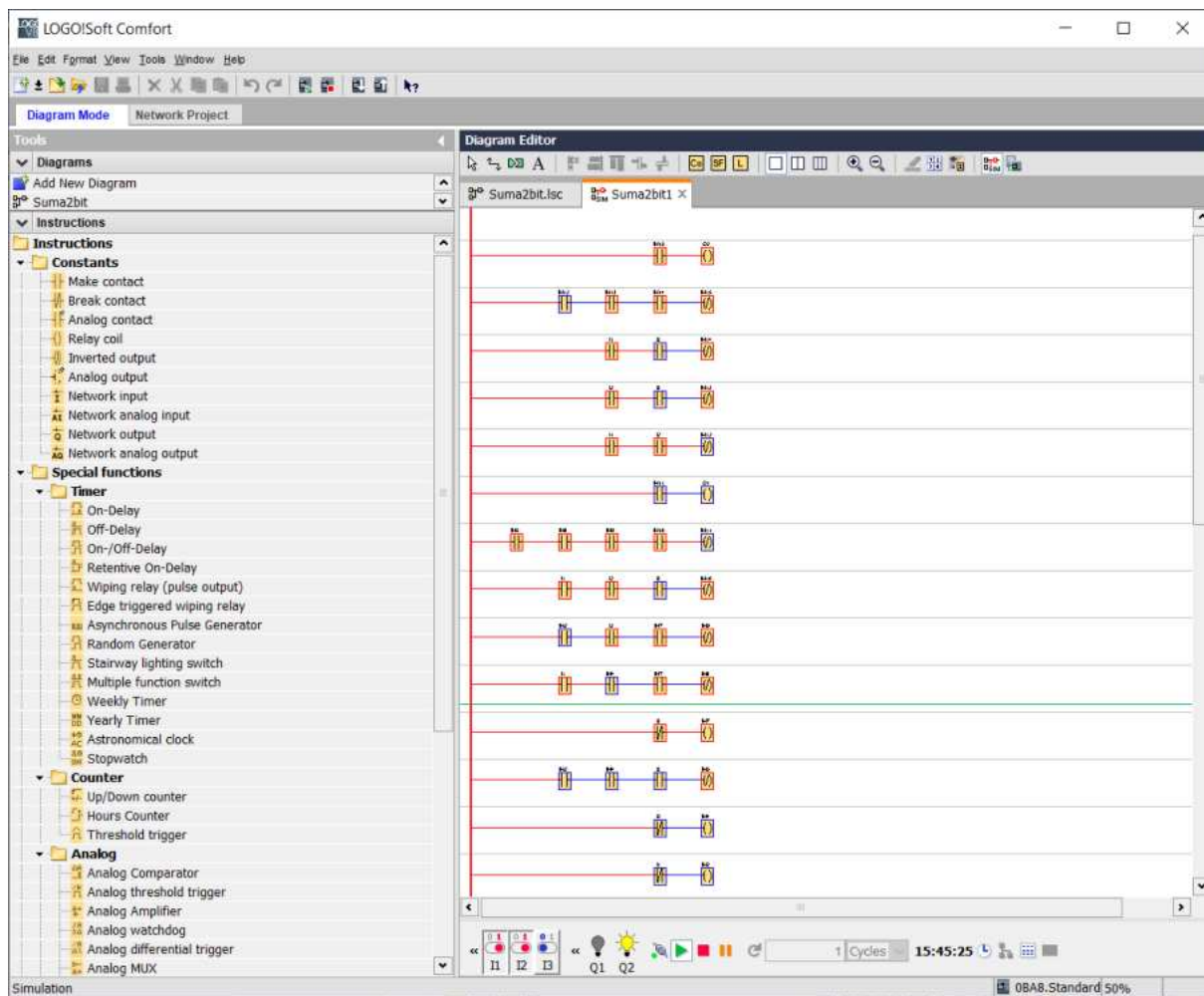
C	/ I1 I2	00	01	11	10
0		0	0	1	0
1		0	1	1	1

Równanie bitu przeniesienia C1 dla sumatora dwóch bitów z przeniesieniem jest następujące:

$$C1 = I1 \cdot I2 + I2 \cdot C + I1 \cdot C$$



Rys. 4.35 Program FBD, suma dwóch bitów z przeniesieniem, I1 i I2 – pojedyncze bity, I3 – bit z przeniesienia

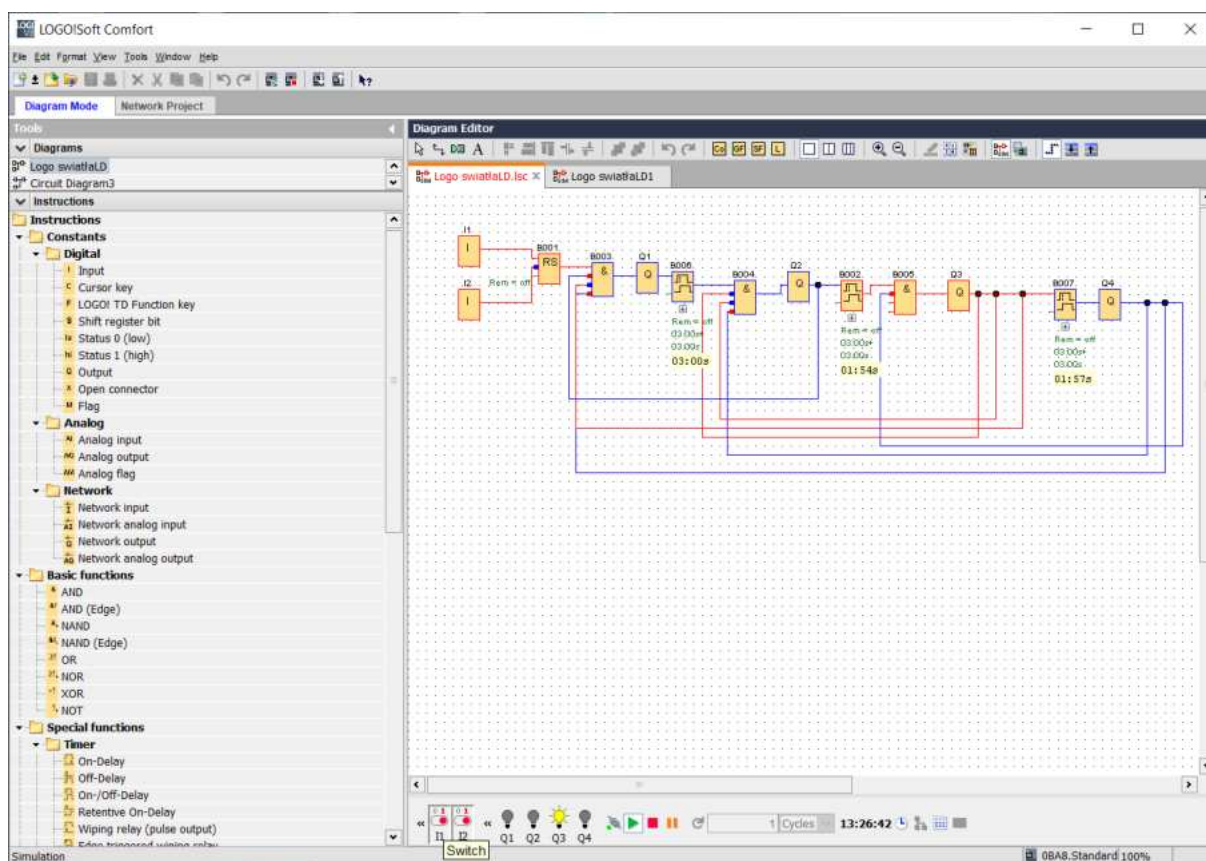


Rys. 4.36 Program LAD, suma dwóch bitów z przeniesieniem, I1 i I2 – pojedyncze bity, I3 – bit z przeniesienia

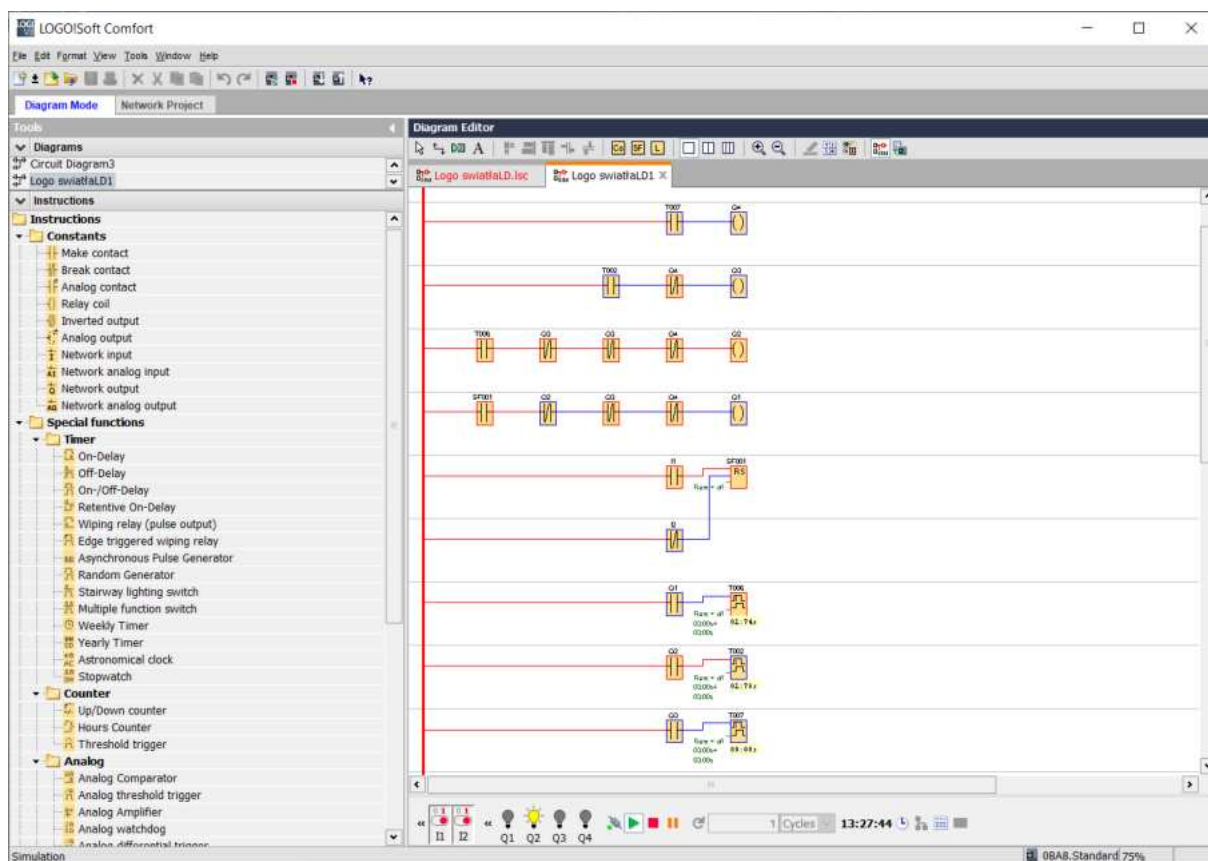
Realizację zadania na sterowniku LOGO! w wersji programu FBD przedstawia Rys. 4.35, w wersji programu LAD przedstawia Rys. 4.36.

18. Sekwencyjne załączanie świateł na skrzyżowaniu dróg

Program na sekwencyjne załączanie świateł drogowych w wersji bloków FBD przedstawia Rys. 4.37, w wersji programu LAD jest przedstawiony na Rys. 4.38.



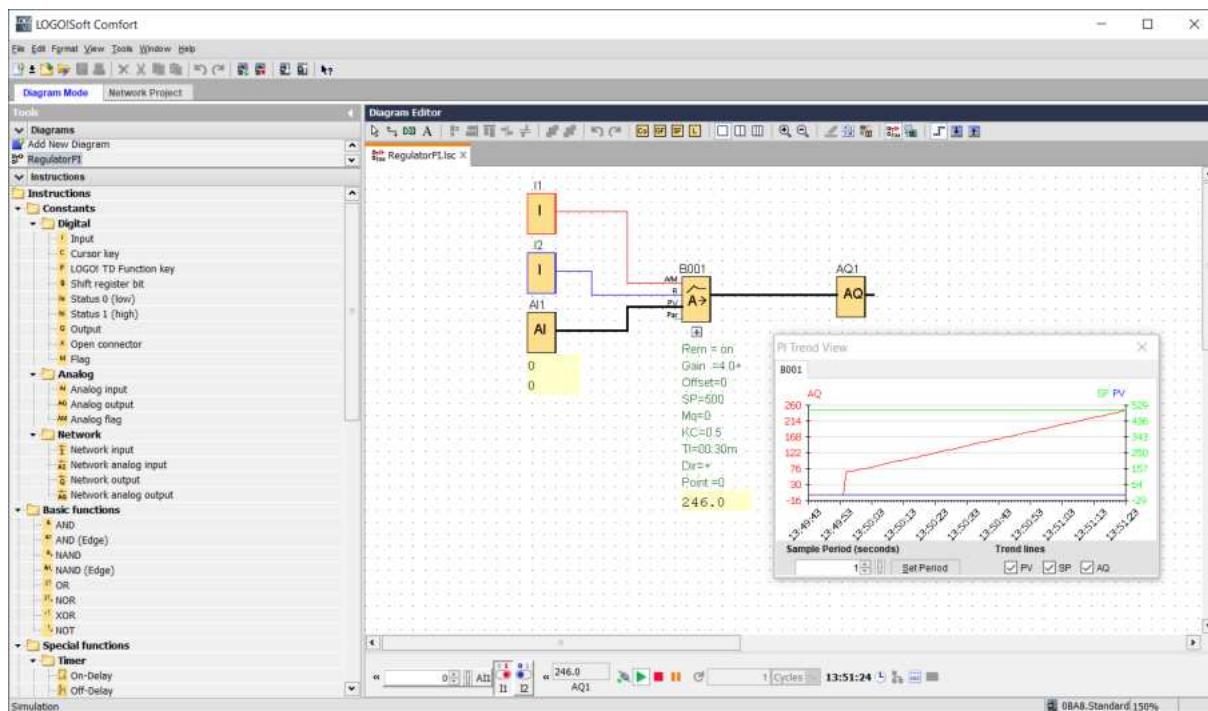
Rys. 4.37 Program na sekwencyjne załączanie świateł drogowych – bloki FBD



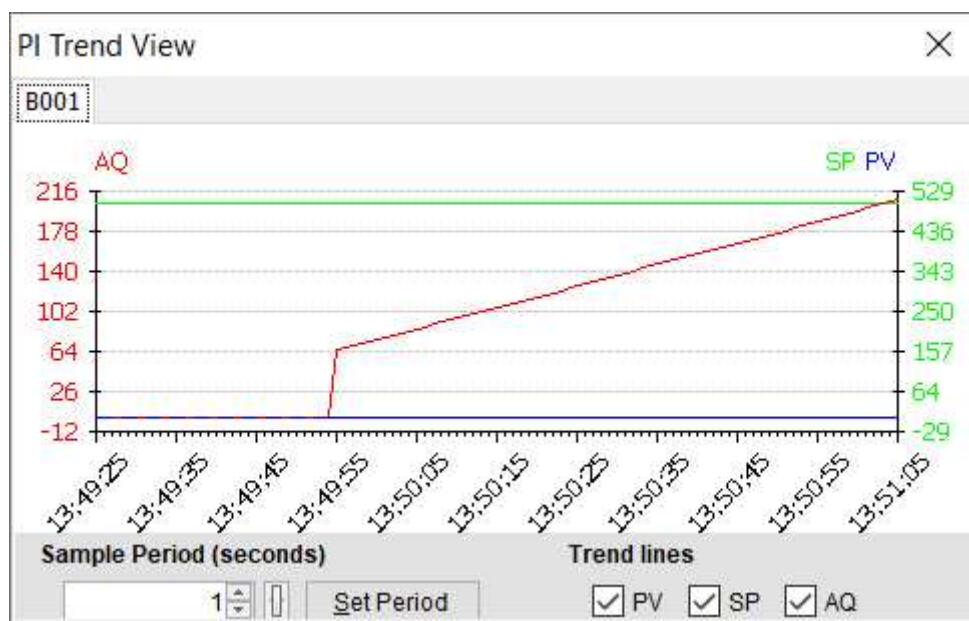
Rys. 4.38 Program na sekwencyjne załączanie świateł drogowych – program LAD

19. Programowanie regulatora PI

Program z regulatorem PI w wersji bloków FBD przedstawiono na Rys. 4.39.

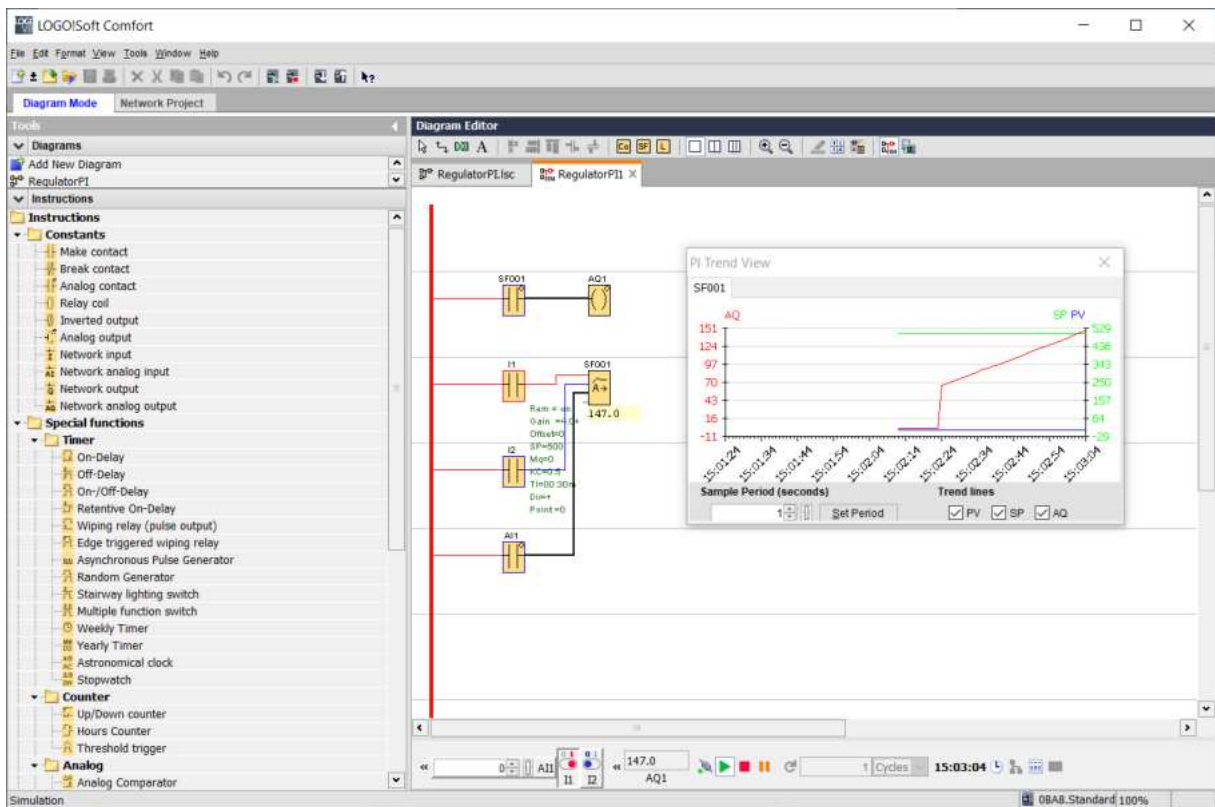


Rys. 4.39 Regulator PI na sterowniku LOGO! – bloki FBD



Rys. 4.40 Linia trendu wyjścia regulatora PI sterownika LOGO

Włączając symulację programu widoczny jest przebieg trendu odpowiedzi regulatora PI, Rys. 4.40



Rys. 4.41 Regulator PI na sterowniku LOGO – program LAD

Po przekształceniu programu na programowanie LAD program z regulatorem PI w wersji języka LAD przedstawiono na Rys. 4.41.

5. Programowanie sterownika GE-FANUC serii 90-30

Sterownik modułowy GE-Fanuc 90-30 przedstawiono na Rys. 5.1.



Rys. 5.1 Sterownik modułowy GE-Fanuc 90-30

5.1 Moduły sterownika GE-Fanuc 90-30

Sterownik modułowy jest układem do sterowania wybranym procesem w całości. Konfiguruje się go według zapotrzebowania. Jest ważnym elementem w procesie zarządzania produkcją.

Elementy sterownika modułowego:

- Płyta łączeniowa (ang. baseplate), zwana także kasetą (ang. rack), posiada gniazda (ang. slots) do podłączenia wybranych modułów,
- Moduły podstawowe: zasilacz (PS, ang. Power Supply) oraz moduł jednostki centralnej (CPU, ang. Central Processing Unit).
- Moduły wejść i wyjść cyfrowych (ang. Digital Input, Digital Output),
- Moduły wejść i wyjść analogowych (ang. Analog Input, Analog Output),
- Moduły komunikacyjne, do podłączenia sterownika do sieci lokalnej w określonym standardzie, np. Modbus, Profibus, ControlNet, Genius itp. lub do sieci Ethernet,

Inne moduły sterownika (moduły inteligentne i dodatkowe):

- Moduły szybkich liczników (HSC, ang. High-Speed Counter),
- Moduły pozycjonowania osi (APM, ang. Axis Positioning Module),
- Moduły wejściowe dla czujników temperatury,
- Moduły regulatora PID lub regulatory rozmyte,
- Moduły akwizycji kodu paskowego itd.

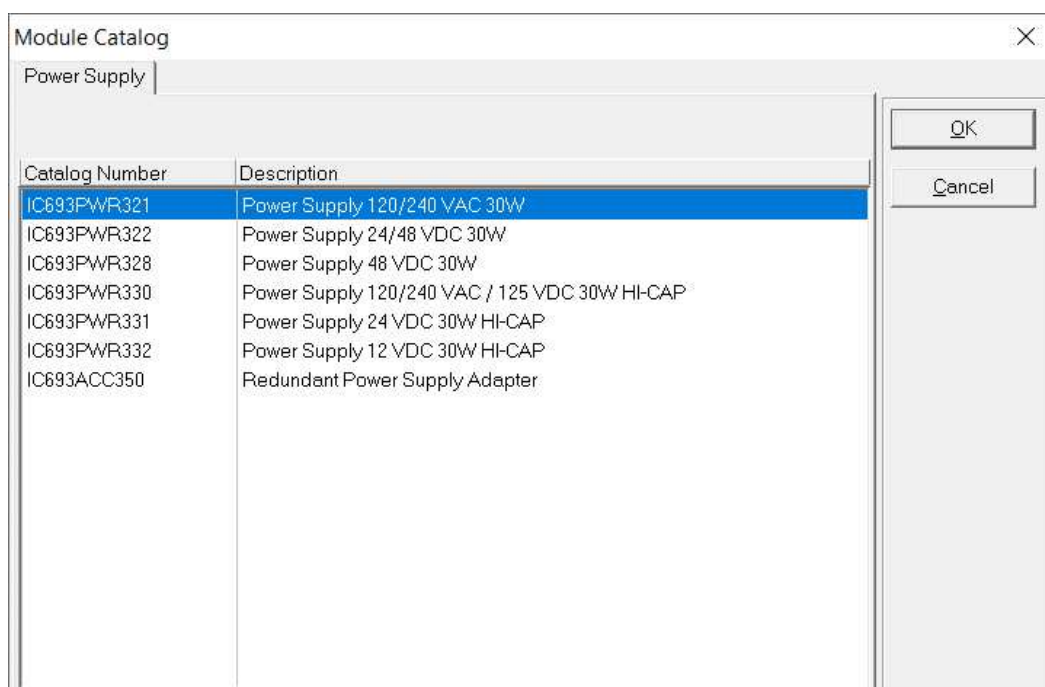
Zasilacz do sterownika GE-Fanuc 90-30

Sterowniki są produkowane w dwóch zasadniczych wersjach – z zasilaniem sieciowym 110V AC albo 230V AC lub z zasilaniem 24V DC. Przy zasilaniu sieciowym koniecznym elementem jest zasilacz (PS, ang. *Power Supply*), który jest wbudowany w sterownikach o budowie kompaktowej, lub stanowi odrębny podzespół w sterownikach o budowie

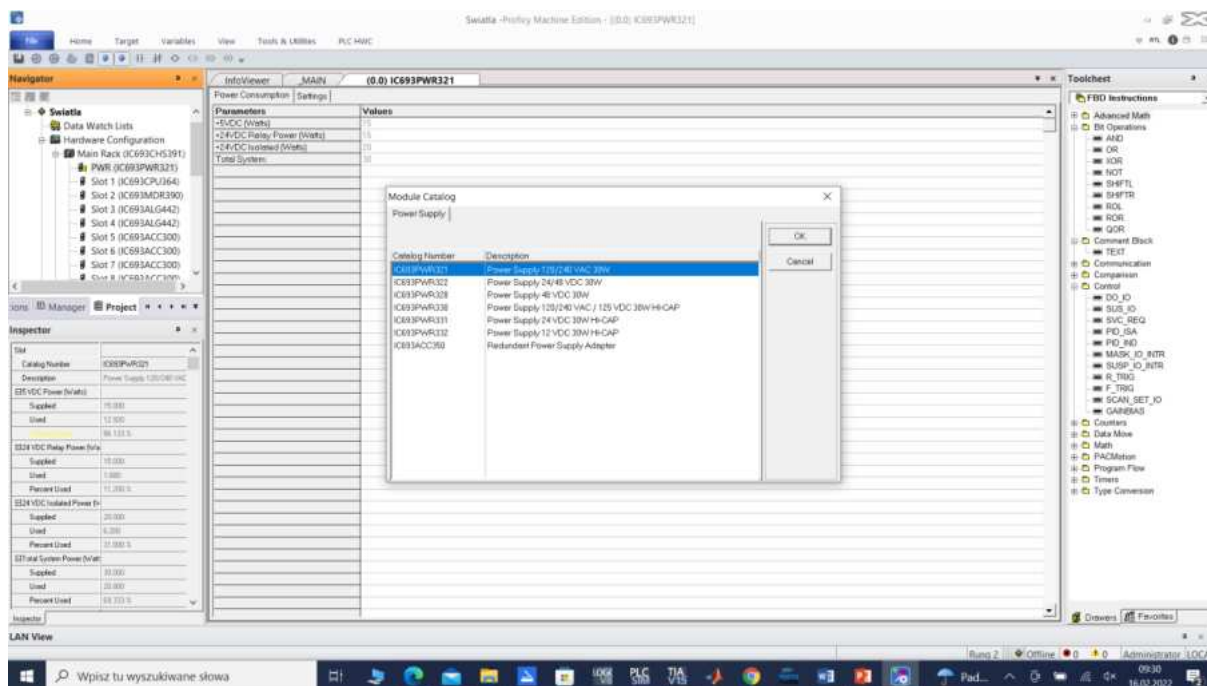
modułowej. Zasilacze modułowe (dedykowane) są specyficzne dla poszczególnych typów i rodzajów sterowników. Są wyposażone w nietypowe złącza i trudno jest zastępować je innymi – standardowymi urządzeniami. Czasami zasilacz stanowi integralną część sterownika, połączoną w taki sposób, że funkcjonowanie zestawu bez dedykowanego zasilacza nie jest możliwe.

Stosowane zasilacze charakteryzują się niewielkimi mocami i służą do zasilania jednostek centralnych oraz poszczególnych modułów sterownika. Nie jest zalecane wykorzystywanie zasilacza sterownika do zasilania innych podzespołów układu sterowania (czujniki, przetworniki, kontrolki itp.).

Katalog dostępnych zasilaczy sterownika GE-Fanuc przedstawia Rys. 5.2. Zasilacz zawsze montuje się porcie 'slot 0' płyty głównej 'Main Rack'. Dodawanie i konfigurowanie zasilacza pokazano na Rys. 5.3.



Rys. 5.2 Dostępne zasilacze sterownika GE-Fanuc 90 - 30



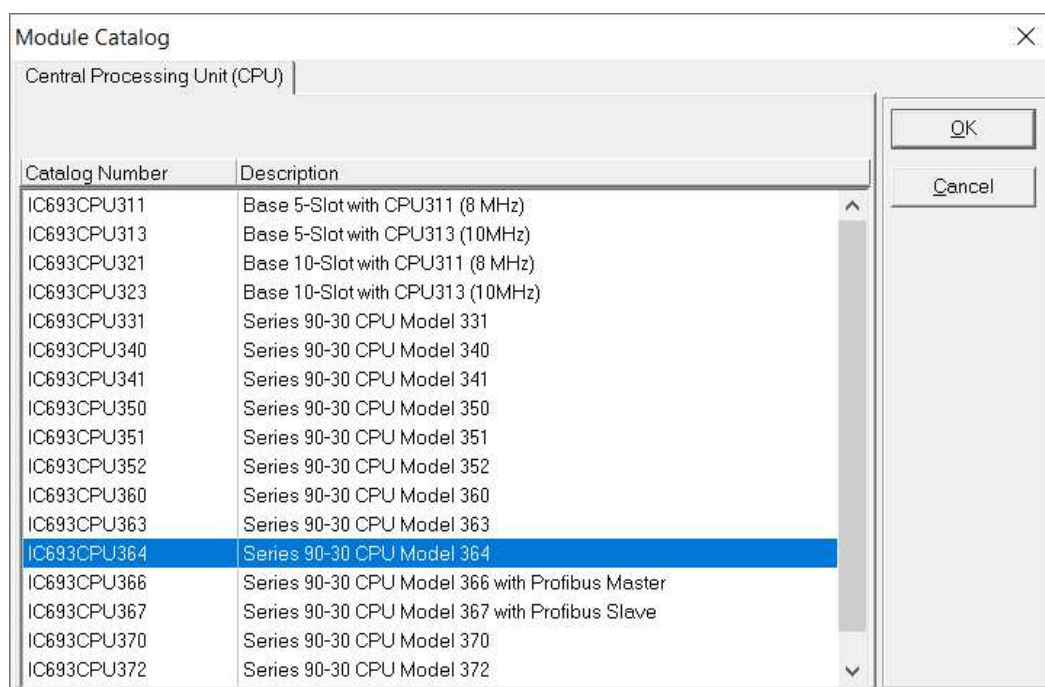
Rys. 5.3 Dodawanie zasilacza i konfiguracja sterownika GE-Fanuc 90 - 30

Jednostka centralna do sterownika GE-Fanuc 90-30

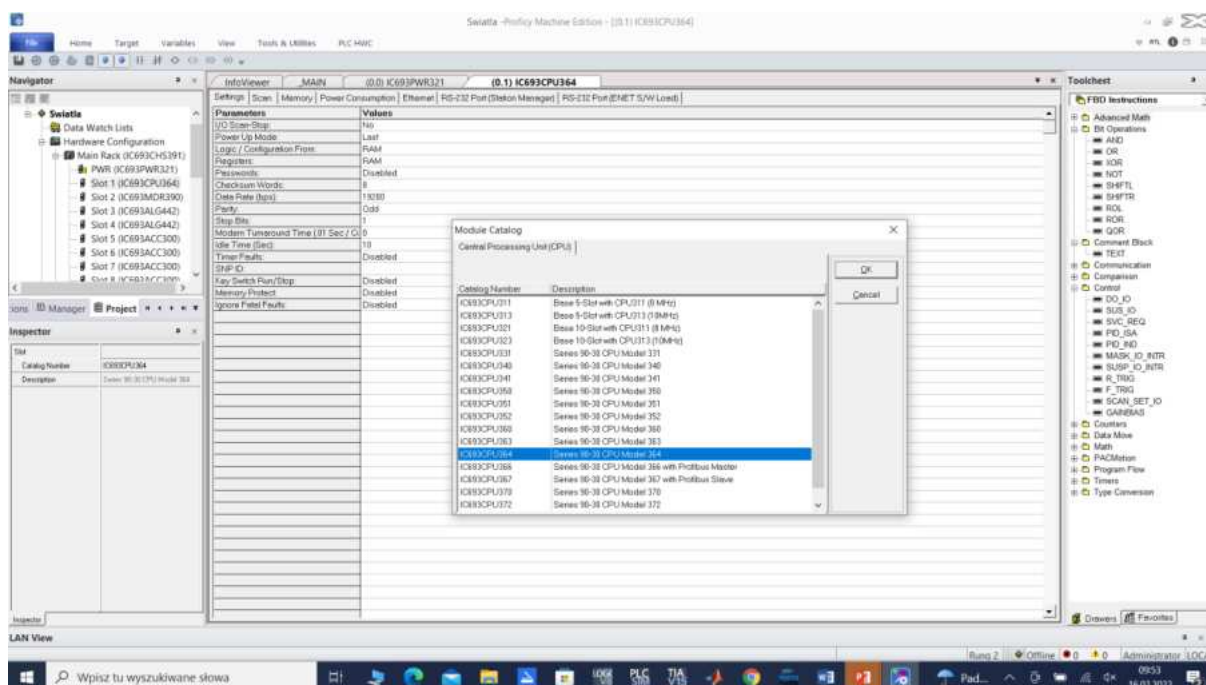
Jednostka centralna (**CPU**, ang. *Central Processing Unit*.) – jest to element, gdzie program jest ładowany, przechowywany i wykonywany, inna nazwa to: Jednostka Główna Przetwarzania (MPU – *Main Processing Unit*).

Parametry jednostki centralnej można scharakteryzować poprzez: wielkość pamięci, szybkość działania, napięcie zasilania, możliwość pracy w systemie *Master – Slave*, rodzaj oprogramowania, liczba dostępnych procedur i bloków funkcyjnych w oprogramowaniu, możliwość pracy w sieci i zaimplementowane protokoły komunikacyjne – porty wymiany danych, liczba dozwolonych modułów rozszerzających, możliwość rozszerzania pamięci.

Katalog dostępnych procesorów sterownika GE-Fanuc 90-30 przedstawia Rys. 5.4. Jednostkę centralną zawsze montuje się porcie 'slot 1' płyty głównej 'Main Rack'. Dodawanie i konfigurowanie jednostki CPU pokazano na Rys. 5.5. Pracę ze sterownikiem zaczyna się zawsze od wgrania programu poprzez port szeregowy RS-232. Możliwe jest przejście na port Ethernet po wcześniejszym wprowadzeniu adresu IP.



Rys. 5.4 Dostępne jednostki CPU sterownika GE-Fanuc 90-30



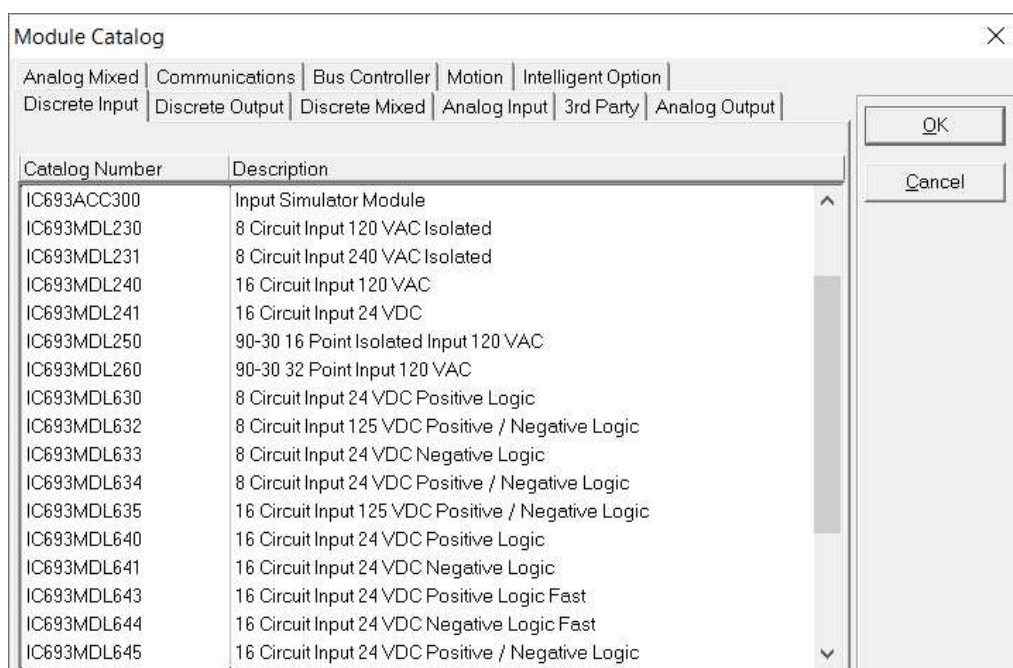
Rys. 5.5 Dodawanie jednostki CPU i konfiguracja sterownika GE-Fanuc 90-30

Moduł wejść cyfrowych

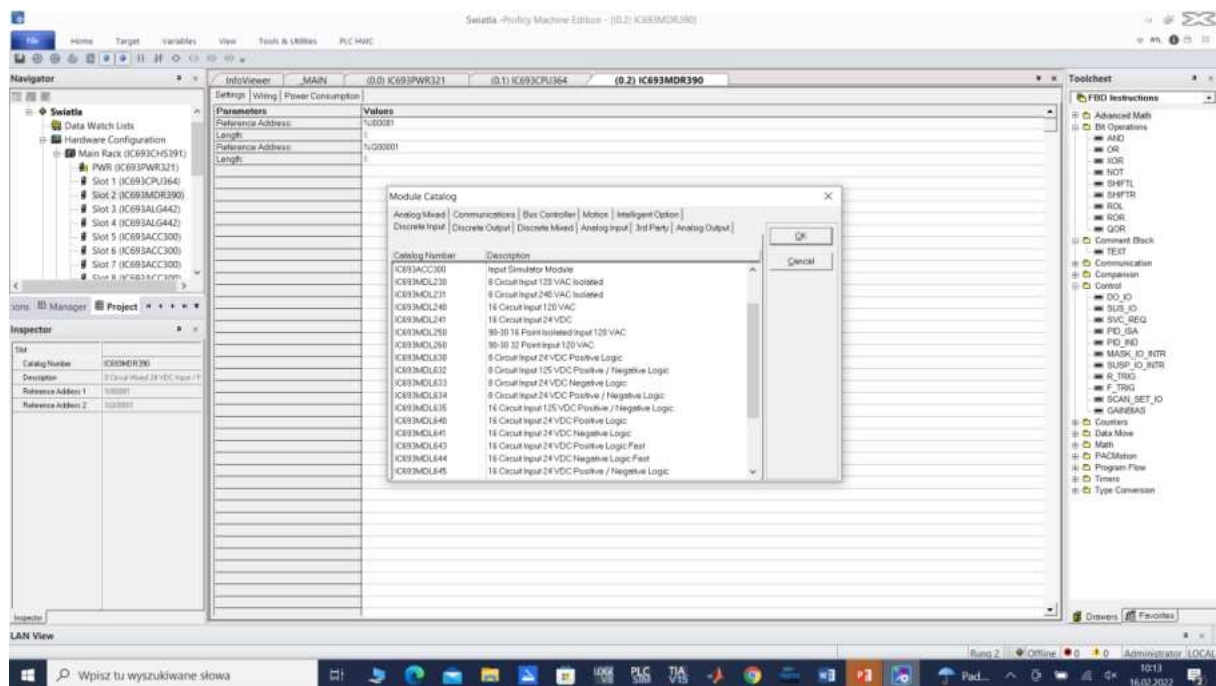
Blok wejść cyfrowych (ang. Digital *Input*) to elementy umożliwiające wprowadzanie sygnałów wejściowych dwustanowych 0 lub 1 bez określenia wartości. Poszczególne wejścia mają określone oznaczenia, zarówno naniesione na obudowie, jak też są jednoznacznie przyporządkowane, tak że w programie możliwe jest ich precyzyjne adresowanie.

Katalog dostępnych modułów wejść cyfrowych GE-Fanuc 90-30 przedstawia Rys. 5.6. Dowolne moduły mogą być montowane w porcie od 'slot 2' i wyżej płyty głównej 'Main Rack' i kolejnych płyt do 'Rack 7'. W taki sposób uzyskuje się szerokie możliwości sterownika. Producent udostępnia szeroką listę modułów cyfrowych, analogowych, mieszanych itp.

Dodawanie i konfigurowanie modułów pokazano na Rys. 5.7. Istotne w konfiguracji jest poznanie adresów wejść/wyjść danego modułu.



Rys. 5.6 Dostępne moduły wejść cyfrowych sterownika GE-Fanuc 90-30



Rys. 5.7 Dodawanie modułów i konfiguracja wejść/wyjść sterownika GE-Fanuc 90-30

Moduł wejść analogowych

Blok wejść analogowych (ang. *Analog Input*) to elementy umożliwiające wprowadzanie sygnałów wejściowych o charakterze ciągłym i odczytanie wartości np. temperatury, ciśnienia, przepływu, itp. Do jednostki centralnej wszystkie sygnały muszą dochodzić jako cyfrowe. Wobec tego sygnały analogowe muszą zostać zamienione na postać cyfrową w przetwornikach A/C, zmienne typu integer.

Sygnałami wejściowymi dla wejścia analogowego są najczęściej:

- napięcie 0 do 10V DC, -10 do 10V DC oraz napięcie pochodzące z czujników termoelektrycznych np.: Ni-Cr;
- prąd: 0 do 20 mA lub 4 do 20 mA DC z czujników rezystancyjnych np. Pt100 itp.

Moduły wejść analogowych zawierają w sobie przetworniki, które dostosowują dostarczany na wejście sygnał do parametrów sterownika według długości słowa cyfrowego. Funkcjonowanie wejść analogowych jest możliwe dzięki zastosowaniu w sterownikach przetworników A/C (analogowo-cyfrowy) o odpowiedniej rozdzielczości:

Przetwornik 8-bitowy, zakres 0-255

dla sygnału 0-10V oznacza to dokładność 39mV

dla sygnału 4-20mA oznacza to dokładność 62.5μA

Przetwornik 12-bitowy, zakres 0-4095

dla sygnału 0-10V oznacza to dokładność 2.44mV

dla sygnału 4-20mA oznacza to dokładność 3.9μA

Przetwornik 16 bitowy – jeden bit znakowy plus 15 bitów, zakres 0-32767
dla sygnału 0-10V oznacza to dokładność 0.3mV
dla sygnału 4-20mA oznacza to dokładność 0.488μA

Moduł komunikacyjny

Blok elementów komunikacyjnych służy do podłączenia sterownika do innych urządzeń to jest programatora lub urządzeń (Device) sieci lokalnej w określonym standardzie, np. Modbus, Profibus, ControlNet, Genius itp. lub do sieci Ethernet. Sterowniki mają w oprogramowaniu wewnętrznym (systemowym) zaimplementowane określone procedury komunikacyjne oraz charakter działania (tryb Multimaster lub Master–Slave). Zewnętrznym elementem modułu są złącza komunikacyjne, najczęściej standardu RS 232, RS 485, RJ 45.

Wyjścia cyfrowe (ang. *Digital Output*)

Wyjścia przekaźnikowe - zapewniają pełną separację galwaniczną wewnętrznych układów sterownika i obwodów wyjściowych. Przekazniki charakteryzują się stosunkowo niewielką częstotliwością włączeń oraz ograniczoną żywotnością wynoszącą około 1 mln cykli łączeniowych.

Wyjścia tranzystorowe - także zapewniają separację galwaniczną wewnętrznych układów sterownika i obwodów wyjściowych. Można je stosować jedynie przy zasilaniu obwodów wyjściowych prądem stałym. Zazwyczaj stosuje się napięcie 24 V. Obciążalność wyjść tranzystorowych jest niewielka – około 0,5 A Wyjścia tranzystorowe charakteryzują się wysoką częstotliwością włączeń oraz dużą żywotnością. Często sterowniki mają kilka rodzajów wyjść tranzystorowych różniących się dopuszczalną częstotliwością np. do 1 kHz oraz do 20 kHz. Tego rodzaju wyjścia przeznaczone są do układów sterowania charakteryzujących się ciągłą pracą przy dużej liczbie przełączeń, np. sterowanie silnikami krokowymi.

Wyjścia analogowe (ang. *Analog Output*)

Moduły wyjść analogowych są niezbędne w układach regulacji oraz do sterowania pracą analogowych elementów i urządzeń wykonawczych (np. sterowanie położeniem przepustnicy, sterowanie położeniem zaworu, sterowanie mocą grzałki itp.).

Moduły wyjść analogowych są zazwyczaj dostosowane do określonego zakresu sygnałów (np. prądowy 0-20 mA lub 4-20 mA, czy napięciowy 0-10 V, lub +/- 10 V). Spotyka się także wyjścia dedykowane do konkretnych urządzeń wykonawczych. Obciążalność wyjść analogowych jest niewielka.

Moduły wyjść analogowych AO (ang. *analog output*) dostarczają sygnał wyjściowy w postaci sygnału analogowego. Następuje to dzięki zastosowaniu przetwornika cyfrowo-analogowego DC/AC (*DAC- Digital-to-analog - converter*) w układzie modułu wyjściowego sterownika.

Sterownik GE-Fanuc 90-30 na stanowisku laboratoryjnym składa się z następujących komponentów na płycie głównej:

- 'slot 0' zasilacz o numerze producenta IC693PWR321
- 'slot 1' jednostka CPU o numerze producenta IC693CPU364
- 'slot 2' moduł wejść/wyjść cyfrowych mieszny o numerze producenta IC693MDR390
- 'slot 3' moduł wejść/wyjść analogowych mieszny o numerze producenta IC693ALG442
- 'slot 4' moduł wejść/wyjść analogowych mieszny o numerze producenta IC693ALG442
- 'slot 5' symulator wejść cyfrowych o numerze producenta IC693ACC300
- 'slot 6' symulator wejść cyfrowych o numerze producenta IC693ACC300

- 'slot 7' symulator wejść cyfrowych o numerze producenta IC693ACC300
- 'slot 8' symulator wejść cyfrowych o numerze producenta IC693ACC300
- 'slot 9' pusty
- 'slot 10' pusty

Obsługa wejść i wyjść wymaga właściwej interpretacji w programowaniu sterownika, Tabela 5.1 oraz Tabela 5.2.

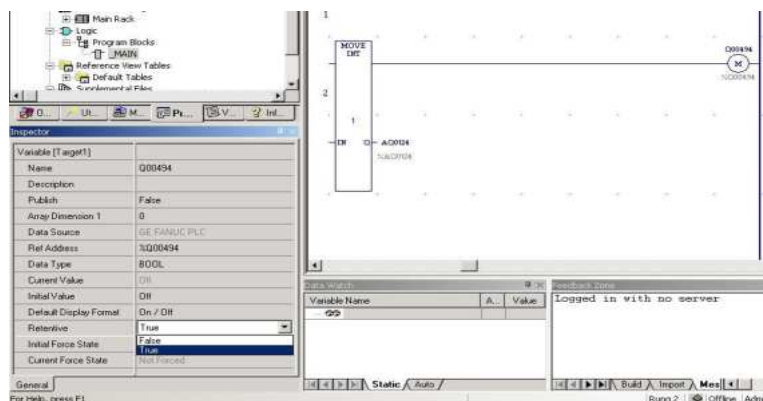
Tabela 5.1 Oznaczenia wejść, wyjść i zmiennych w programie sterownika GE-FANUC 90 - 30

Typ	Opis
%I	Wejście dyskretne, zawartość obszaru pamięci pokazywany w tabeli stanu wejść.
%Q	Wyjście dyskretne, zawartość obszaru pamięci pokazywany w tabeli stanu wyjść, adresy typu %Q mogą posiadać pamięć stanu lub nie, w zależności od charakteru ich wykorzystywania w programie.
%AI	Wejście analogowe, oznacza rejestr wejścia analogowego, po symbolu podawany jest adres rejestru np. %AI0016, zmienna przechowuje wartość wejścia analogowego
%AQ	Wyjście analogowe, oznacza rejestr wyjścia analogowego, po symbolu podawany jest adres rejestru np. %AI0026, zmienna przechowuje wartość wyjścia analogowego
%R	Zmienna 16-bitowa oznaczająca rejestr sterownika, w którym można przechowywać dane programu i wyniki obliczeń, po symbolu podawany jest adres rejestru np. %R0246
%M	Zmienna wewnętrzna (pomocnicza) programu sterującego, po symbolu podawany jest adres pamięci np. %M0124, adresy typu %M mogą posiadać pamięć stanu lub nie, w zależności od charakteru ich wykorzystywania w programie.
%T	Zmienna tymczasowa (tracąca swój stan po zaniku zasilania lub zatrzymaniu-uruchomieniu sterownika) do zapisu danych, po symbolu podawany jest adres pamięci np. %T0024, zmienne typu %T nie mogą być wykorzystywane w przekaznikach z pamięcią.
%S	Zmienne systemowe posiadające ściśle określone znaczenie, tylko do odczytu, adresy typu %S mogą być wykorzystane jako parametry wejściowe funkcji lub bloków funkcyjnych.
%G	Zmienne globalne, obszar pamięci wykorzystywany przez dane globalne, wartość danych przechowywanych w pamięci %G jest pamiętana po wyłączeniu zasilania, adresy typu %G mogą być wykorzystywane z stykami i przekaznikami z pamięcią, ale nie mogą być wykorzystywane w przypadku przekazników bez pamięci.

Zmienne posiadają pamięć stanu, jej wartość jest przechowywana po wyłączeniu sterownika. Sterowniki serii 90 przechowują:

- część logiczną programu sterującego,
- tabele błędów działania i diagnostyczne,
- zablokowane wartości zmiennych,
- wartości zmiennych rejestrowych (typu %R, %AI, %AQ),
- wartości zmiennych dyskretnych (typu %I, %S, %G, bity - znaczniki błędów działania)
- wartości zmiennych dyskretnych typu %M i %Q w przypadku, gdy są one przypisane przekaznikom z pamięcią, wartości zmiennych %T nie są zapamiętywane.

Zmienne %Q i %M domyślnie nie występują z pamięcią (są NONRETENTIVE). Opcja ta musi zostać zmieniona na RETENTIVE w oknie INSPEKTORA, Rys. 5.8.



Rys. 5.8 Opcja RETENTIVE dla zmiennych %Q i %M dla sterownika GE-Fanuc 90 - 30

Występują dwa podstawowe typy zmiennych:

- dyskretne (1 bit), przeznaczone do opisu styków i przekaźników,
- analogowe (słowo 16 bitowe), przeznaczone jako parametry bloków funkcyjnych i sygnałów analogowych.

Tabela 5.2 Typy danych sterownika GE-FANUC 90 - 30

Typ	Nazwa	Opis
BOOL	Boolean	Dana zajmująca najmniejszą komórkę pamięci, przyjmuje wartość 0 lub 1
BYTE	Bajt	Dana zawierająca 8 bitów, wartość całkowita w zakresie 0-255
INT	Liczba całkowita ze znakiem	Liczba całkowita ze znakiem zajmująca 16 bitów w zakresie od -32768 do +32767, dwa bajty, 16 bit znakowy
DINT	Liczba całkowita podwójnej precyzji ze znakiem	Liczba całkowita podwójnej precyzji ze znakiem zajmująca 32 bity w zakresie od -2 147 483 648 do +2 147 483 647, cztery bajty, 32 bit znakowy
WORD	Słowo	Słowo złożone z 16 bitów, dwa bajty bez znaku
BCD	Format BCD (Four-Digit binary Coded Decimal)	Czterocyfrowa liczba dziesiętna zakodowana w formacie BCD zajmująca 16 bitów pamięci, dwa bajty, każda z czterech cyfr danej liczby jest zakodowana w czterech bitach i może reprezentować cyfrę z zakresu od 0 do 9, zakres wartości od 0 do 9999 z przeznaczeniem do prezentacji na wyświetlaczu
REAL	Liczba rzeczywista, zmiennoprzecinkowa, mantysa – cecha (wykładnik)	Liczba rzeczywista ze znakiem zajmująca 32 bity w zakresie od $\pm 1.401298E-45$ do $\pm 3.402823E+38$, cztery bajty, 32 bit znakowy
TIME	Czas	Zmienna czasu w formacie d-dni, h-godziny, m-minuty, s-sekundy, ms-milisekundy

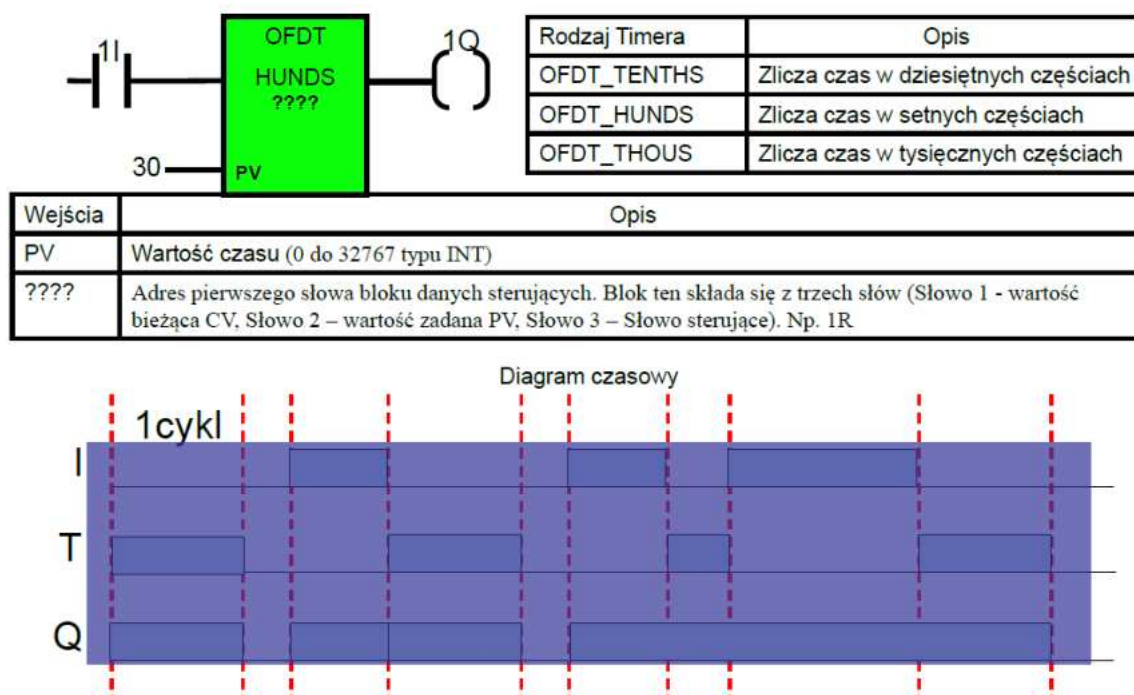
Tabela 5.3 przedstawia zmienne systemowe dostępne w programowaniu GE-FANUC 90 – 30.

Tabela 5.3 Wybrane zmienne systemowe sterownika GE-FANUC 90 - 30

Zmienna	Nazwa	Definicja
%S0001	FST_SCN	Jeżeli bieżący cykl jest pierwszym cyklem pracy sterownika, ustawiana jest wartość 1.
%S0002	LST_SCN	Jeżeli bieżący cykl jest ostatnim cyklem pracy sterownika, ustawiana jest wartość 0.
%S0003	T_10MS	Podstawa czasu generatora sygnału prostokątnego: 0.01 s.
%S0004	T_100MS	Podstawa czasu generatora sygnału prostokątnego: 0.1 s.
%S0005	T_SEC	Podstawa czasu generatora sygnału prostokątnego: 1.0 s.
%S0006	T_MIN	Podstawa czasu generatora sygnału prostokątnego: 1.0 minuta.
%S0007	ALW_ON	Styk zawsze zamknięty.
%S0008	ALW_OFF	Styk zawsze otwarty.
%S0009	SY_FULL	Jeżeli tabela błędów działania sterownika jest wypełniona całkowicie, ustawiana jest wartość 1. Wartość zmiennej jest ustawiana na 0, gdy z tabeli błędów zostanie usunięta pozycja lub cała zawartość tabeli zostanie wymazana.
%S0010	IO_FULL	Jeżeli tabela błędów działania układów wejść/wyjść jest wypełniona całkowicie, ustawiana jest wartość 1. Wartość zmiennej jest ustawiana ponownie na 0, gdy z tabeli zostanie usunięta część informacji lub cała zawartość tabeli zostanie wymazana.
%S0014	PLC_BAT	Wartość 1 informuje o uszkodzeniu baterii w jednostce centralnej. Zmienna ta uaktualniana jest jeden raz w ciągu cyklu. Zmienna ta jest obsługiwana wyłącznie przez 23- i 28-punktowe sterowniki Micro.
%S0022	USR_SW	Wartość tej zmiennej informuje o stanie przełącznika trybu jednostki centralnej Run/Stop: 1 = Run/On 0 = Stop/Off
%SC0015	SFT_FLT	Wartość ustawiana na 1 w momencie pojawienia się błędu programowego. Ustawiana na 0, gdy obie tabele błędów działania nie zawierają komunikatów o błędach.

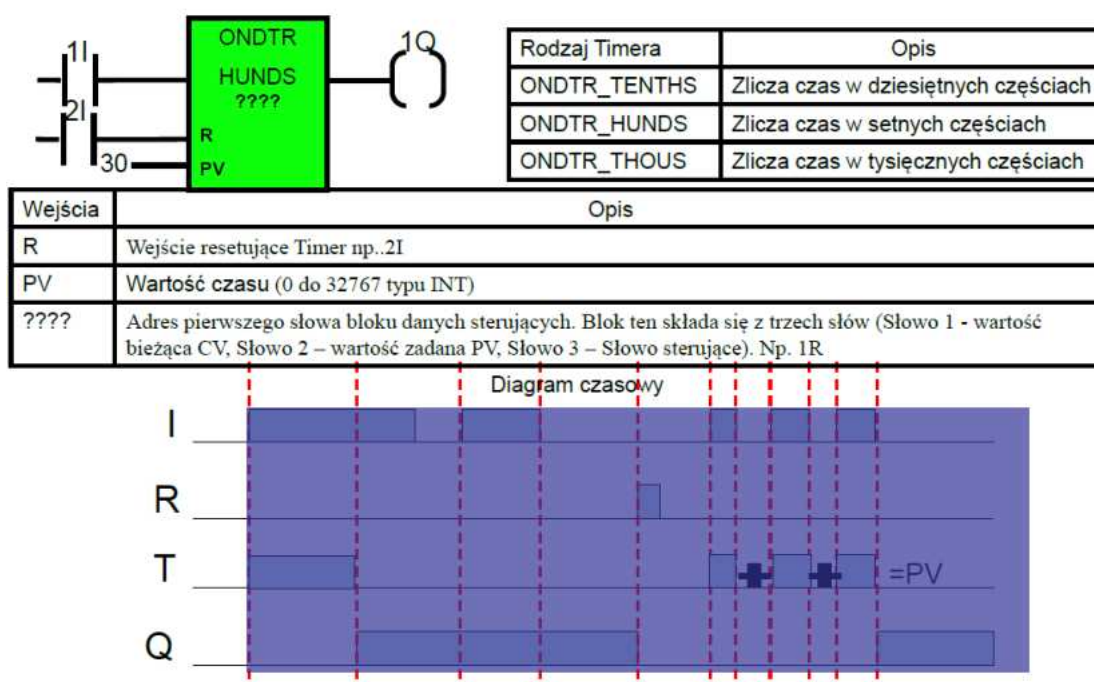
5.2 Timery GE-Fanuc 90 - 30

1. Timer GE Fanuc OFDT –opóźnienie wyłączenia, Rys. 5.9



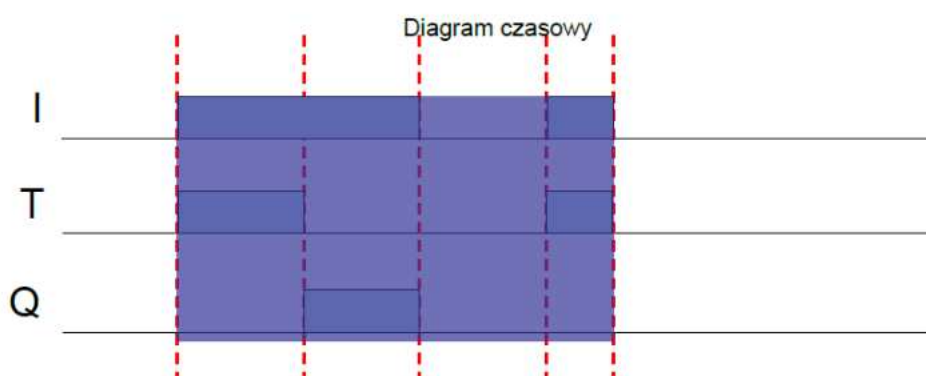
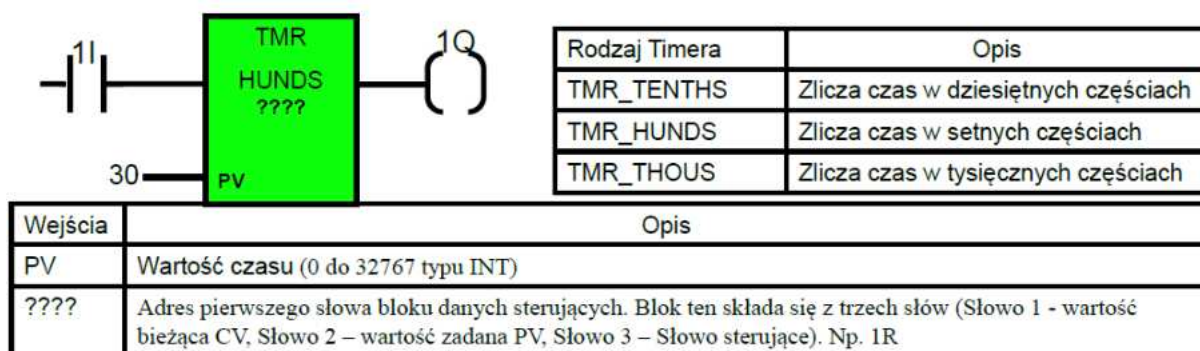
Rys. 5.9 Timer OFDT opóźnienie wyłączenia dla sterownika GE-Fanuc 90 - 30

2. Timer GE Fanuc ONDTR –opóźnienie załączenia z pamięcią, Rys. 5.10



Rys. 5.10 Timer ONDTR opóźnienie załączenia z pamięcią dla sterownika GE-Fanuc 90 - 30

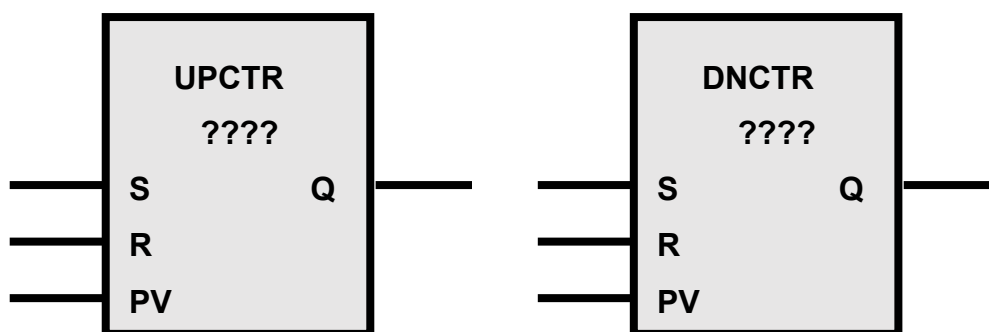
3. Timer GE Fanuc TMR –opóźnienie załączenia, Rys. 5.11



Rys. 5.11 Timer TMR opóźnienie załączenia dla sterownika GE-Fanuc 90 - 30

5.3 Liczniki sterownika GE-Fanuc 90-30

Na sterowniku są dostępne liczniki zliczające w górę i w dół, Rys. 5.12.



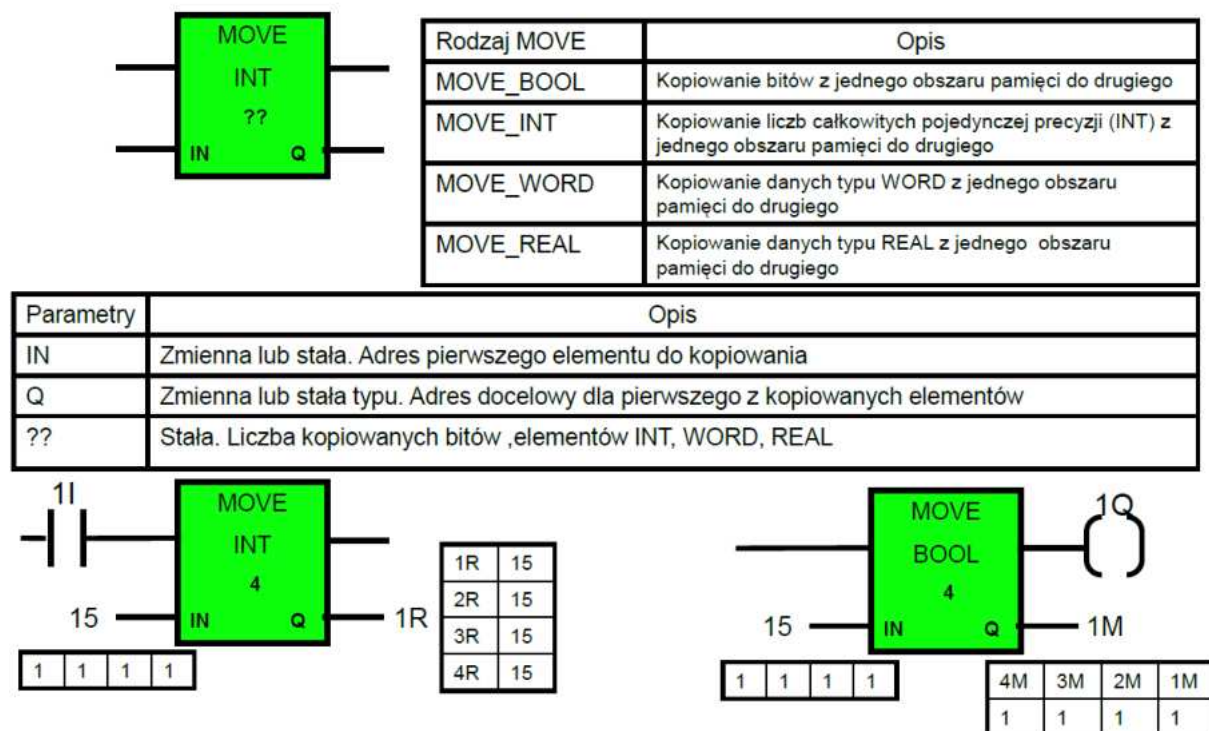
Rys. 5.12 Liczniki UPCTR i DNCTR

Tabela 5.4 Parametry liczników

UPCTR – licznik zliczający w górę	DNCTR – licznik zliczający w dół
S – wejście zliczające w górę lub w dół w zależności od licznika (na wejściu musi być styk o adresie np. 1I)	
R – wejście zerujące stan licznika zliczającego w górę (na wejściu musi być styk o adresie np. 2I)	
R – wejście ustawiające stan licznika zliczającego w dół na wartość PV (na wejściu musi być styk o adresie np. 2I)	
PV –wartość (0 do 32767 typu INT) po przekroczeniu której zapala się wyjście Q dla UPCRT np. 5	
PV –wartość (0 do 32767 typu INT) od której zaczyna liczyć licznik DNCRT np. 25	
Q - wyjście licznika, ma stan logicznej jedynki jeżeli wartość licznika jest większa lub równa PV dla UPCRT	
Q - wyjście licznika, ma stan logicznej jedynki jeżeli wartość licznika jest równa zero dla DNCRT	
???? - Adres pierwszego słowa bloku danych sterujących. Blok ten składa się z trzech słów (Słowo 1 - wartość bieżąca CV, Słowo 2 – wartość zadana PV, Słowo 3 – Słowo sterujące). Np. 1R a następny licznik powinien mieć adres 4R	

5.4 Przesyłanie danych Move na sterowniku GE Fanuc 90-30

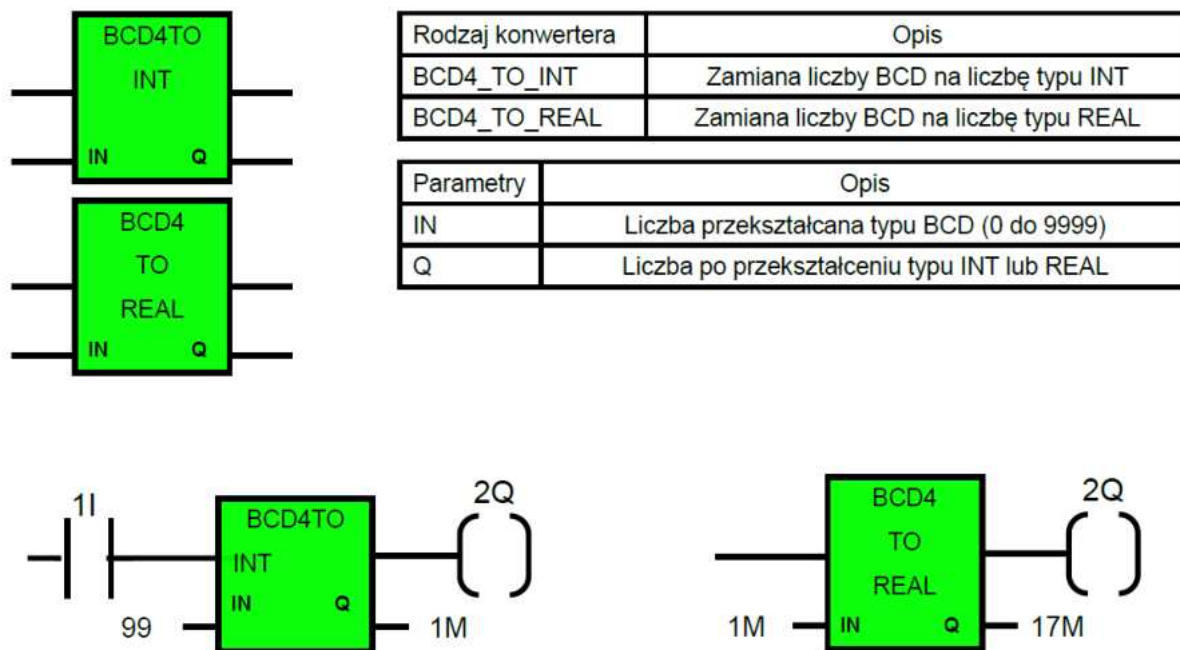
Przesyłanie danych Move –GE Fanuc, Rys. 5.13



Rys. 5.13 Przesyłanie danych Move dla sterownika GE-Fanuc 90 - 30

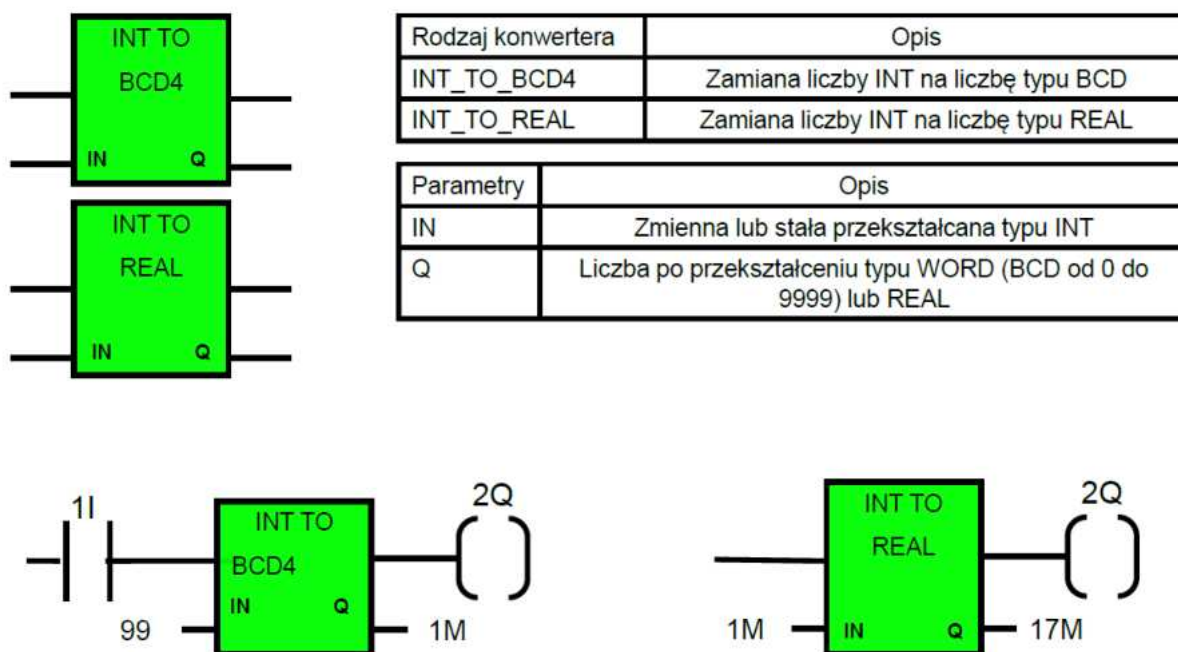
5.5 Konwertery sterownika GE Fanuc 90-30

1. Zamiana kodu BCD do typu integer i real, Rys. 5.14.



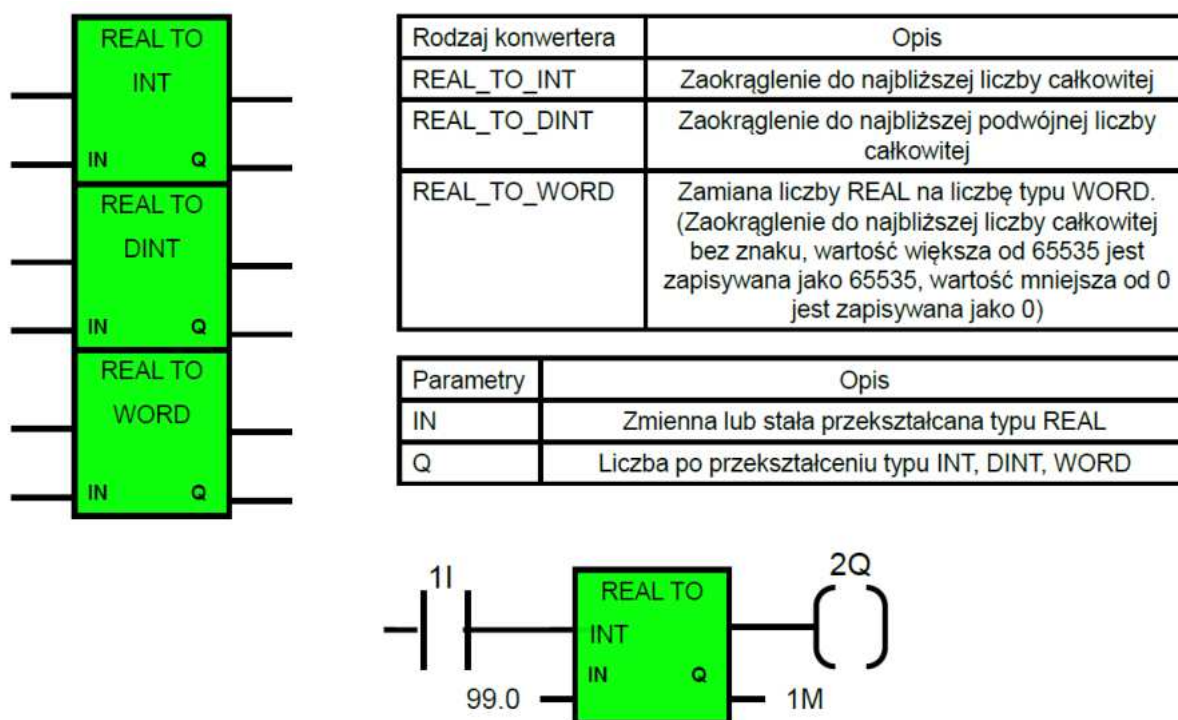
Rys. 5.14 Zamiana kodu BCD dla sterownika GE-Fanuc 90 - 30

2. Zamiana typu integer do kodu BCD i real, Rys. 5.15



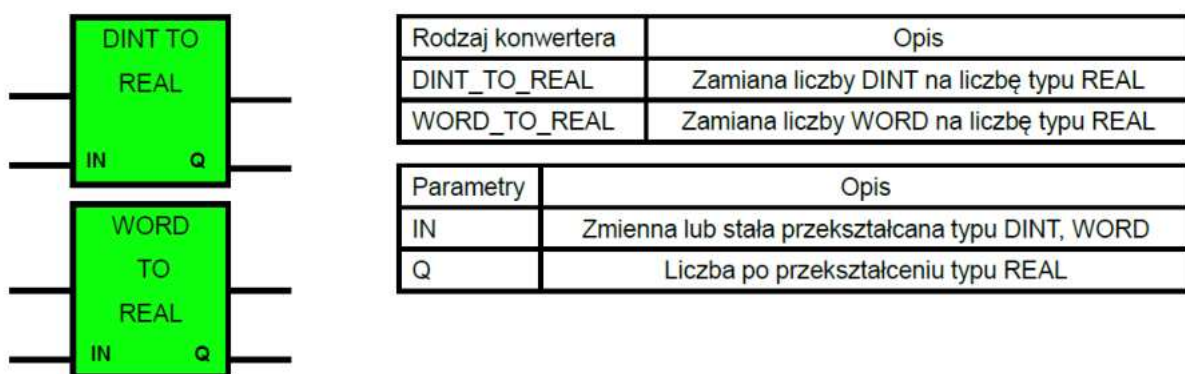
Rys. 5.15 Zamiana typu integer dla sterownika GE-Fanuc 90 - 30

3. Zamiana typu real do typu integer i word, Rys. 5.16



Rys. 5.16 Zamiana typu real dla sterownika GE-Fanuc 90 - 30

4. Zamiana typu DInteger do typu real, Rys. 5.17.



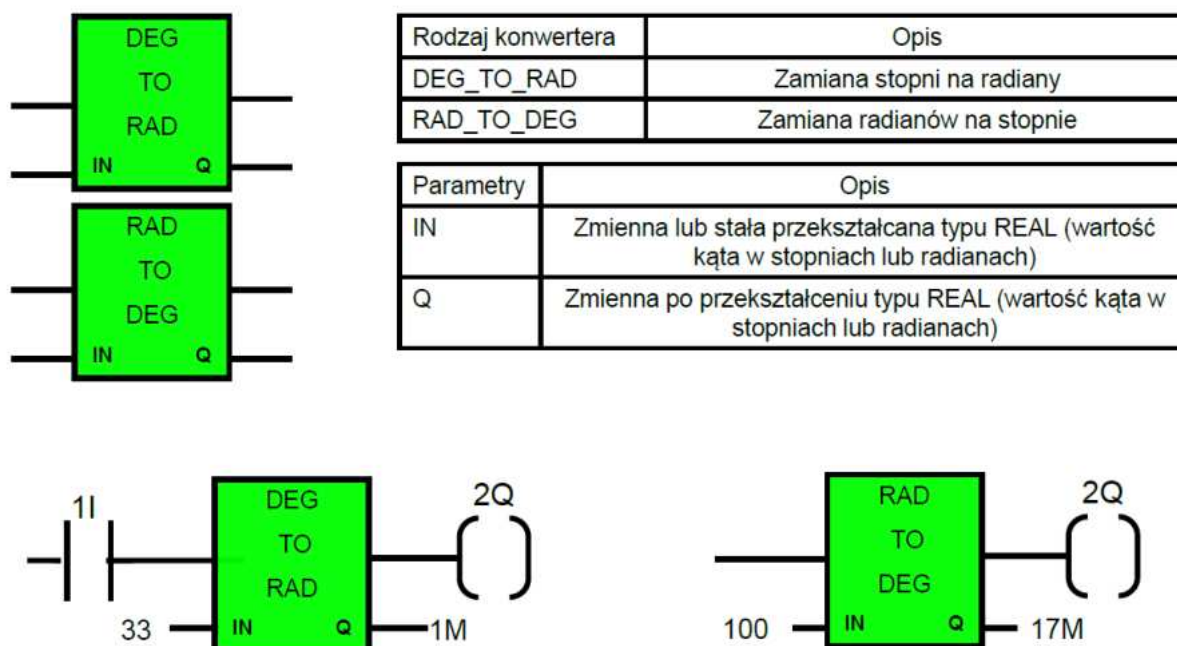
Rys. 5.17 Zamiana typu dinteger dla sterownika GE-Fanuc 90 - 30

5. Zamiana typu real do typu integer z usunięciem części poprzecinku, Rys. 5.18



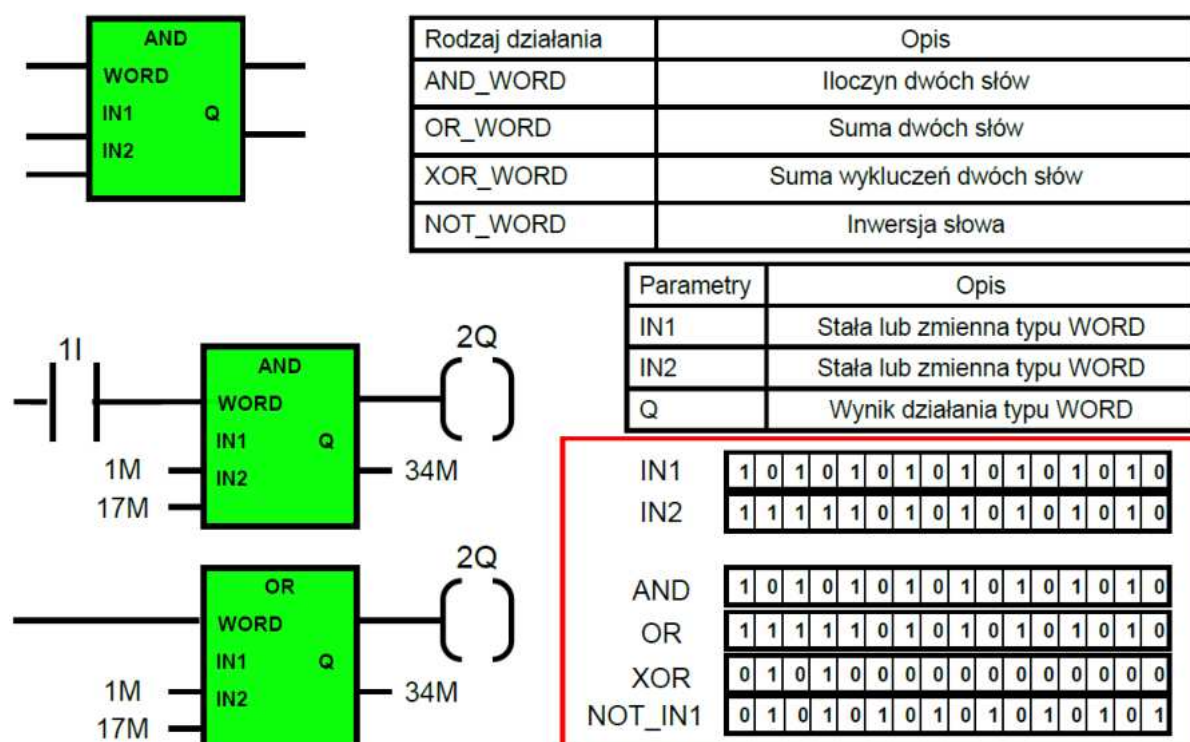
Rys. 5.18 Zamiana typu real z zaokrągleniem dla sterownika GE-Fanuc 90 - 30

6. Zamiana stopni na radiany i odwrotnie, Rys. 5.19.



Rys. 5.19 Zamiana stopni na radiany i odwrotnie dla sterownika GE-Fanuc 90 - 30

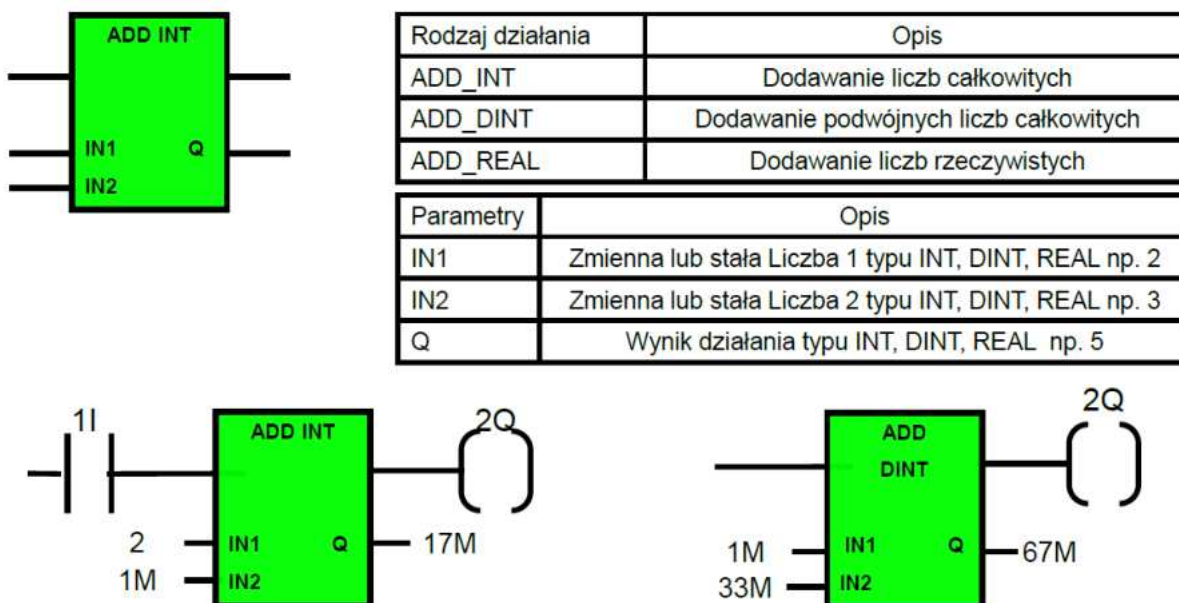
5.6 Operacje logiczne na słowach sterownika GE Fanuc 90-30



Rys. 5.20 Podstawowe funkcje logiczne dla sterownika GE-Fanuc 90 - 30

5.7 Podstawowe funkcje matematyczne na sterowniku GE Fanuc 90-30

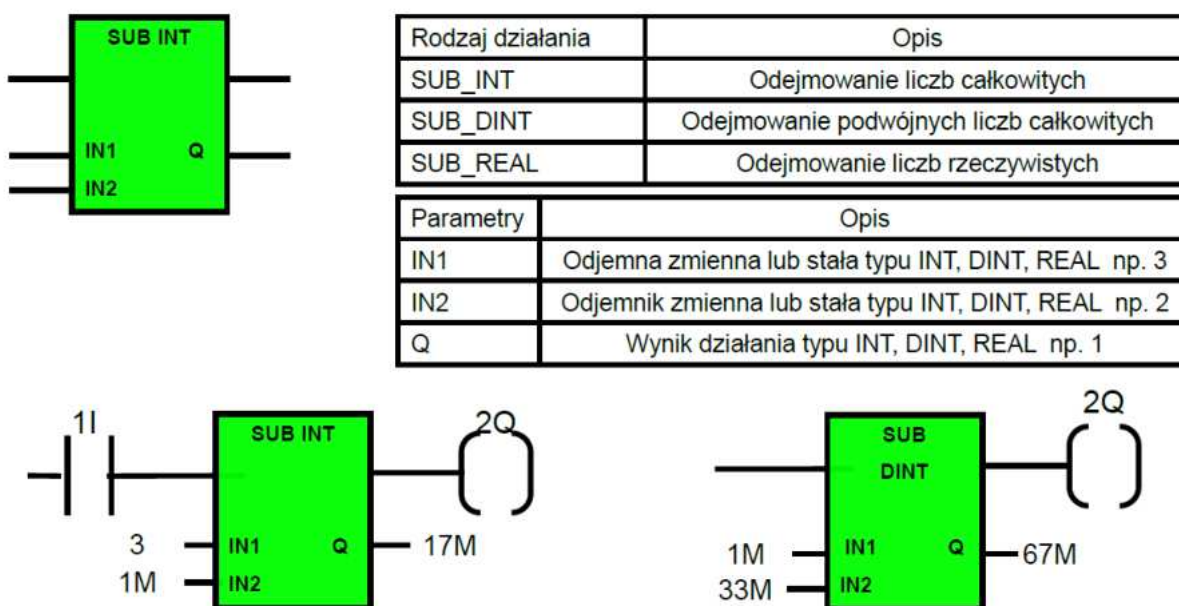
1. Funkcja matematyczna ADD, Rys. 5.21



Rys. 5.21 Funkcja dodawania liczb całkowitych ADD dla sterownika GE-Fanuc 90 - 30

Sygnał wyjściowy jest wysyłany, jeżeli w czasie wykonywania operacji dodawania nie przekroczono dopuszczalnego zakresu wartości. W przypadku przekroczenia dopuszczalnego zakresu wartości, wynik jest równy najwyższej, dopuszczalnej wartości i posiada poprawny znak, ale nie jest wysyłany sygnał wyjściowy.

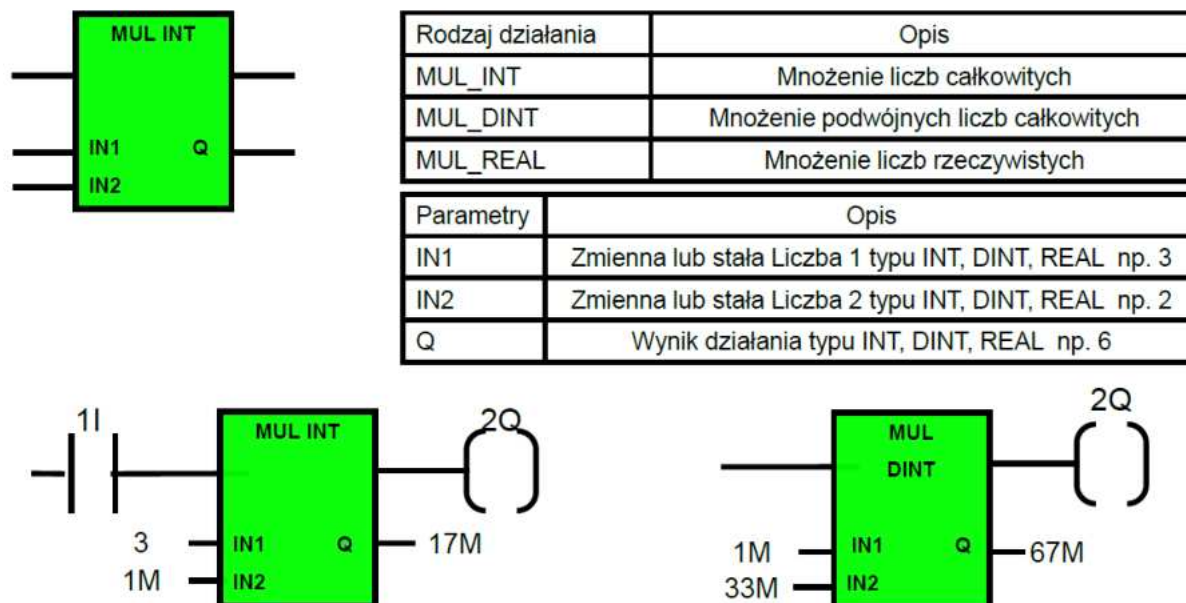
2. Funkcja matematyczna SUB, Rys. 5.22



Rys. 5.22 Funkcja odejmowania liczb całkowitych SUB dla sterownika GE-Fanuc 90 - 30

Sygnal wyjściowy jest wysyłany, jeżeli w czasie odejmowania nie przekroczono zakresu dopuszczalnych wartości lub nie wystąpił inny błąd. W przypadku przekroczenia dopuszczalnego zakresu wartości, wynik jest równy najwyższej, dopuszczalnej wartości i posiada poprawny znak, ale nie jest wysyłany sygnał wyjściowy.

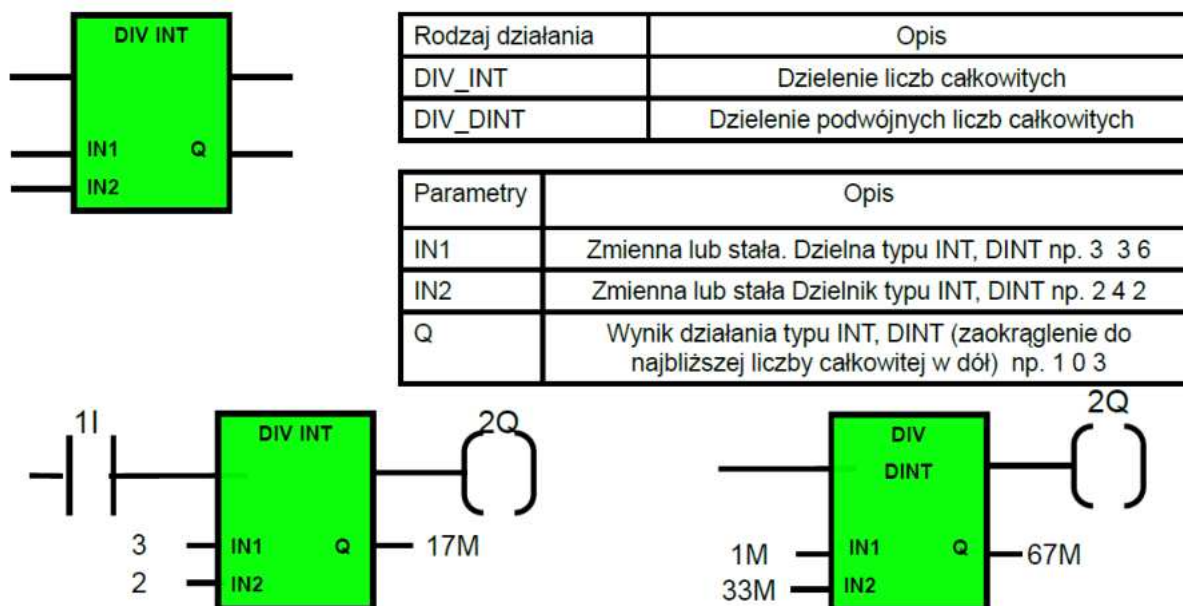
3. Funkcja matematyczna MUL, Rys. 5.23



Rys. 5.23 Funkcja mnożenia liczb całkowitych MUL dla sterownika GE-Fanuc 90 - 30

Sygnal wyjściowy jest wysyłany, jeżeli w czasie mnożenia nie przekroczono zakresu dopuszczalnych wartości lub nie wystąpił inny błąd. W przypadku przekroczenia dopuszczalnego zakresu wartości, wynik jest równy najwyższej, dopuszczalnej wartości i posiada poprawny znak, ale nie jest wysyłany sygnał wyjściowy.

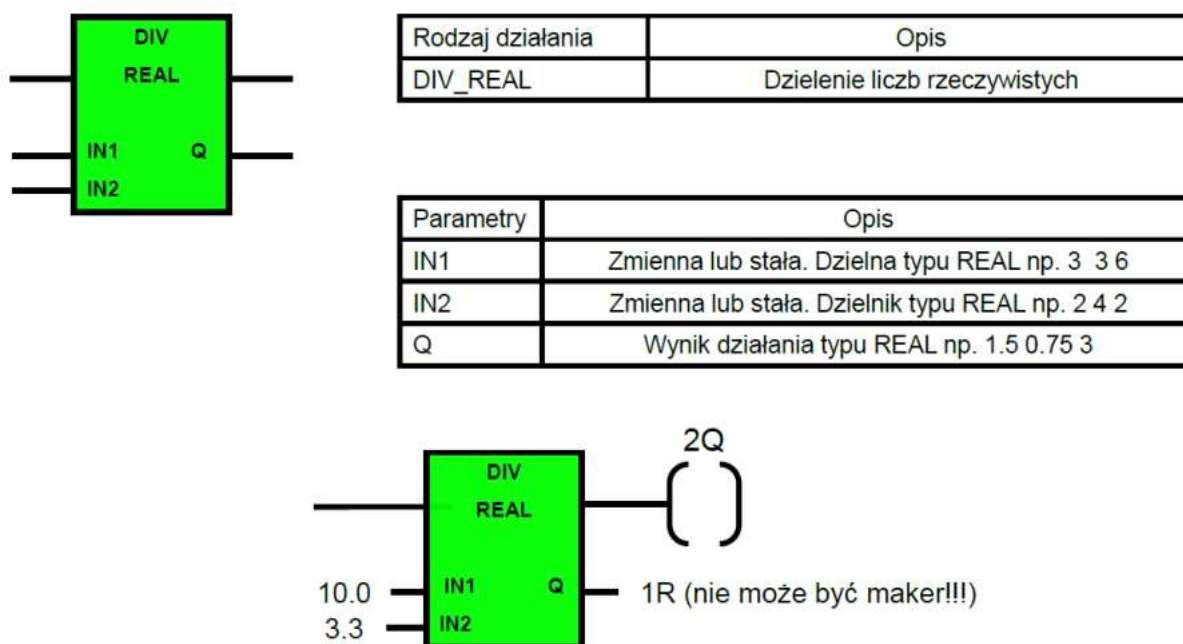
4. Funkcja matematyczna DIV, Rys. 5.24



Rys. 5.24 Funkcja dzielenia liczb całkowitych DIV dla sterownika GE-Fanuc 90 - 30

Sygnal wyjściowy jest wysyłany, jeżeli w czasie wykonywania dzielenia nie przekroczono dopuszczalnego zakresu wartości i nie dokonano niedozwolonej operacji. W przypadku przekroczenia dopuszczalnego zakresu wartości, wynik jest równy najwyższej, dopuszczalnej wartości i posiada poprawny znak, ale nie jest wysyłany sygnał wyjściowy.

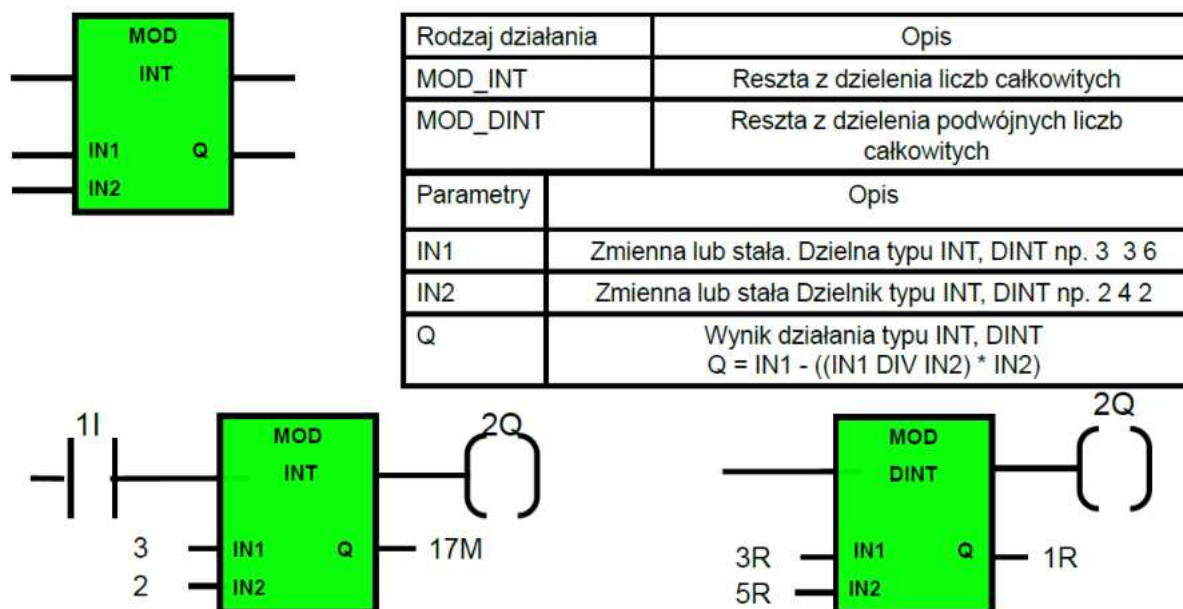
5. Funkcja matematyczna DIV, Rys. 5.25



Rys. 5.25 Funkcja dzielenia liczb rzeczywistych DIV dla sterownika GE-Fanuc 90 - 30

Wszystkie trzy parametry muszą być tego samego typu. Znak liczby wynikowej jest zawsze taki sam jak znak parametru wejściowego IN1. Sygnal wyjściowy jest przesyłany zawsze po doprowadzeniu sygnału wejściowego chyba, że nastąpi próba dzielenia przez zero. W takim wypadku sygnał wyjściowy nie jest wysyłany.

6. Funkcja matematyczna MOD – reszta z dzielenia całkowitego, Rys. 5.26

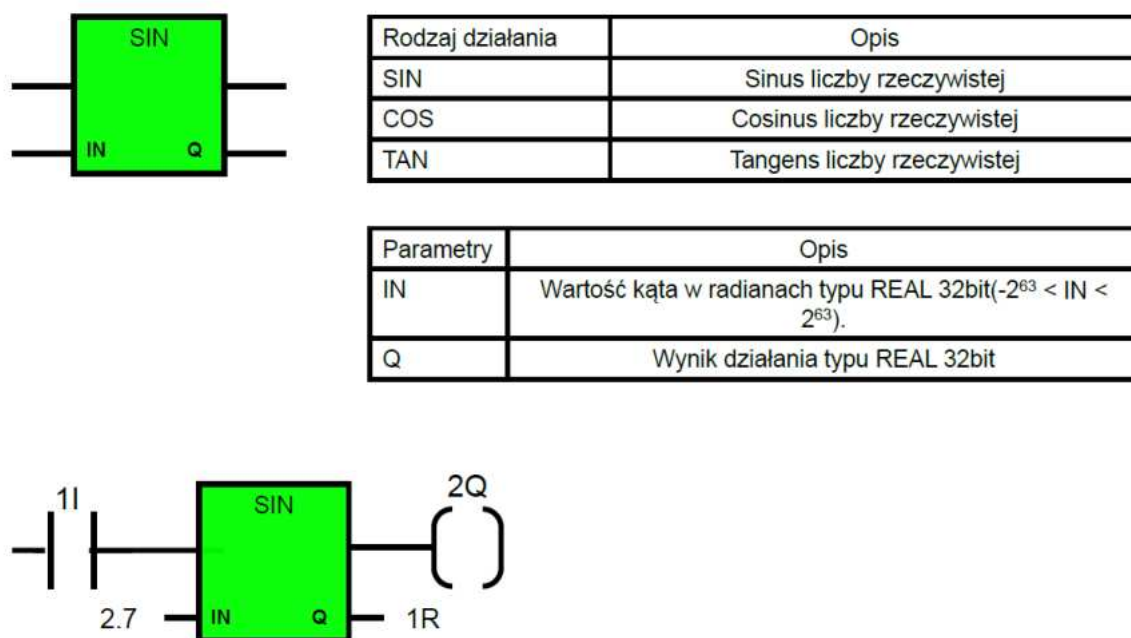


Rys. 5.26 Funkcja dzielenia liczb całkowitych MOD (reszta z dzielenia) dla sterownika GE-Fanuc 90 - 30

Wszystkie trzy parametry muszą być tego samego typu. Znak liczby wynikowej jest zawsze taki sam jak znak parametru wejściowego IN1. Sygnał wyjściowy jest przesyłany zawsze po doprowadzeniu sygnału wejściowego chyba, że nastąpi próba dzielenia przez zero. W takim wypadku sygnał wyjściowy nie jest wysyłany.

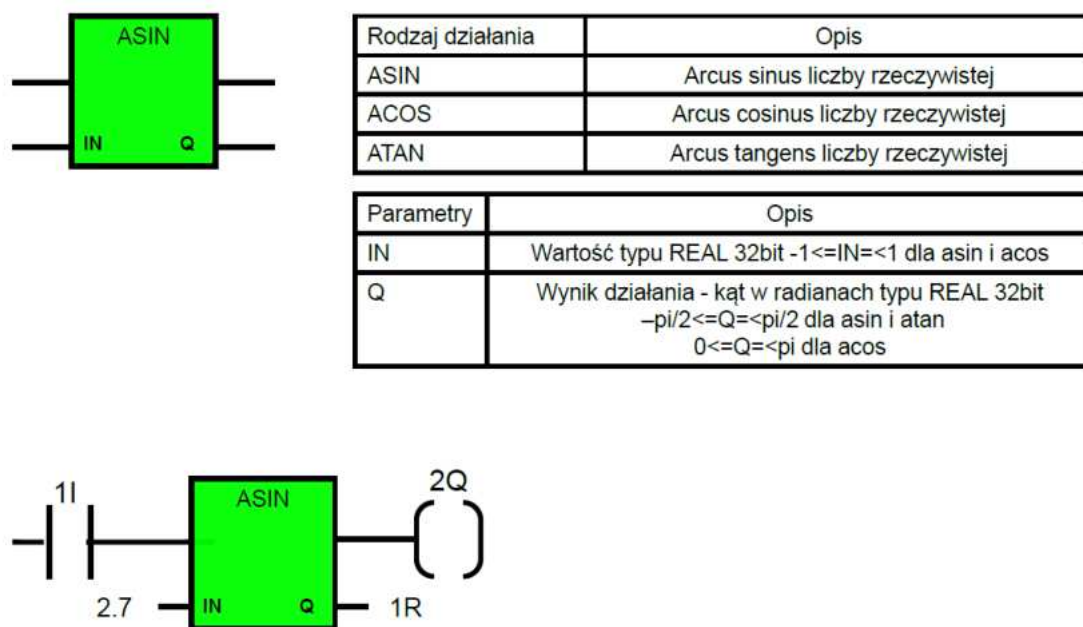
5.8 Rozszerzone funkcje matematyczne –GE Fanuc

1. Funkcja trygonometryczna sinus SIN, Rys. 5.27



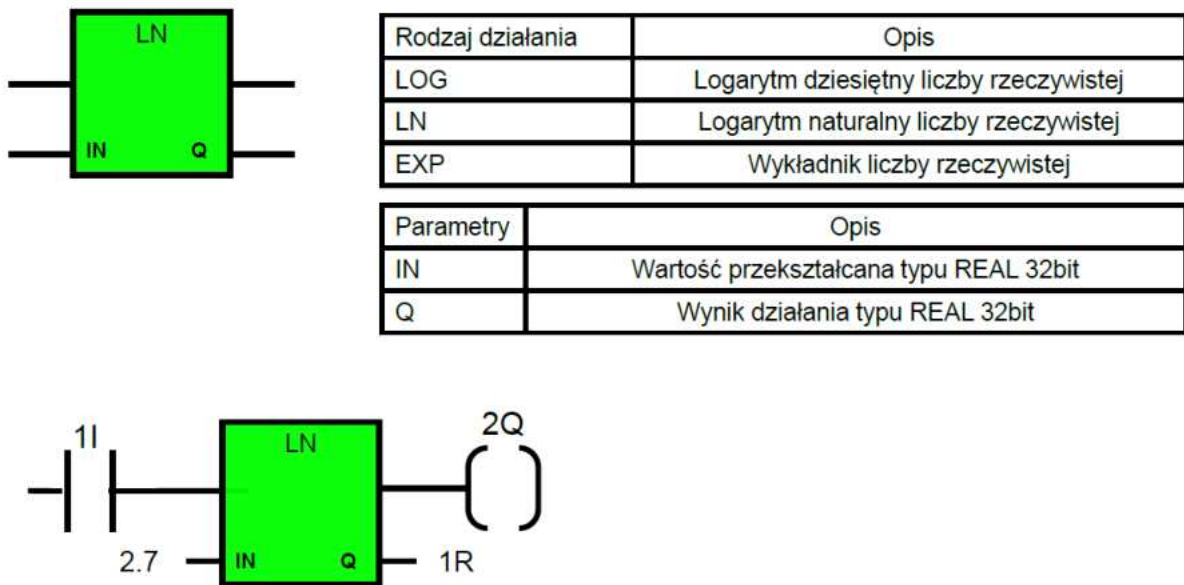
Rys. 5.27 Funkcja trygonometryczna SIN dla sterownika GE-Fanuc 90 - 30

2. Funkcja trygonometryczna odwrotna arcsinus ASIN, Rys. 5.28



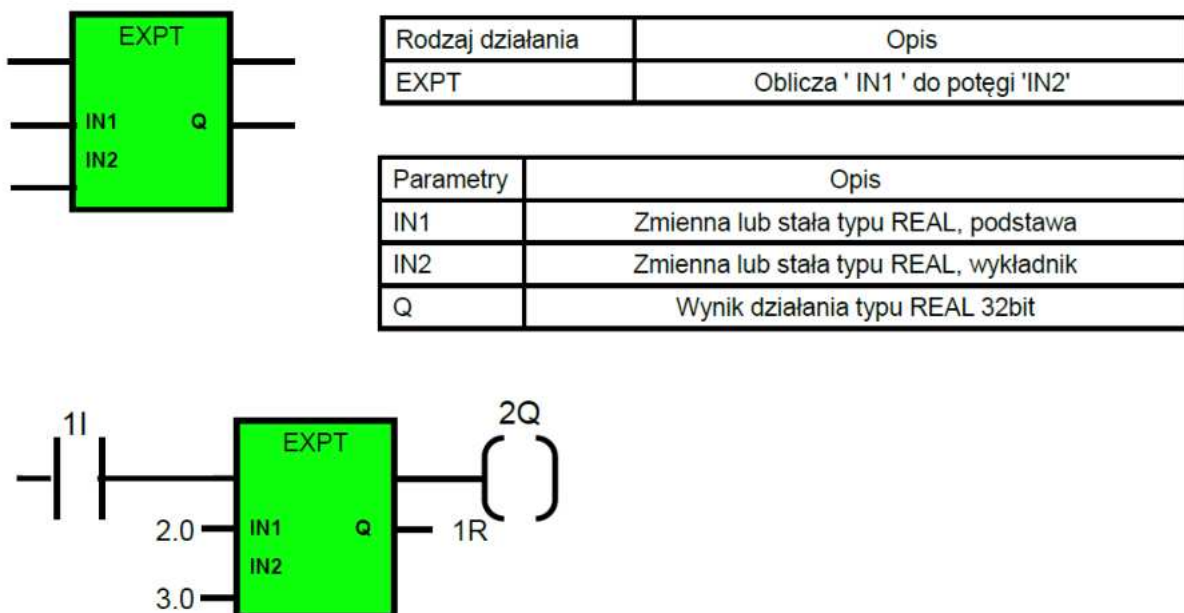
Rys. 5.28 Funkcja trygonometryczna odwrotna ASIN dla sterownika GE-Fanuc 90 - 30

3. Funkcja logarymiczna LN, Rys. 5.29



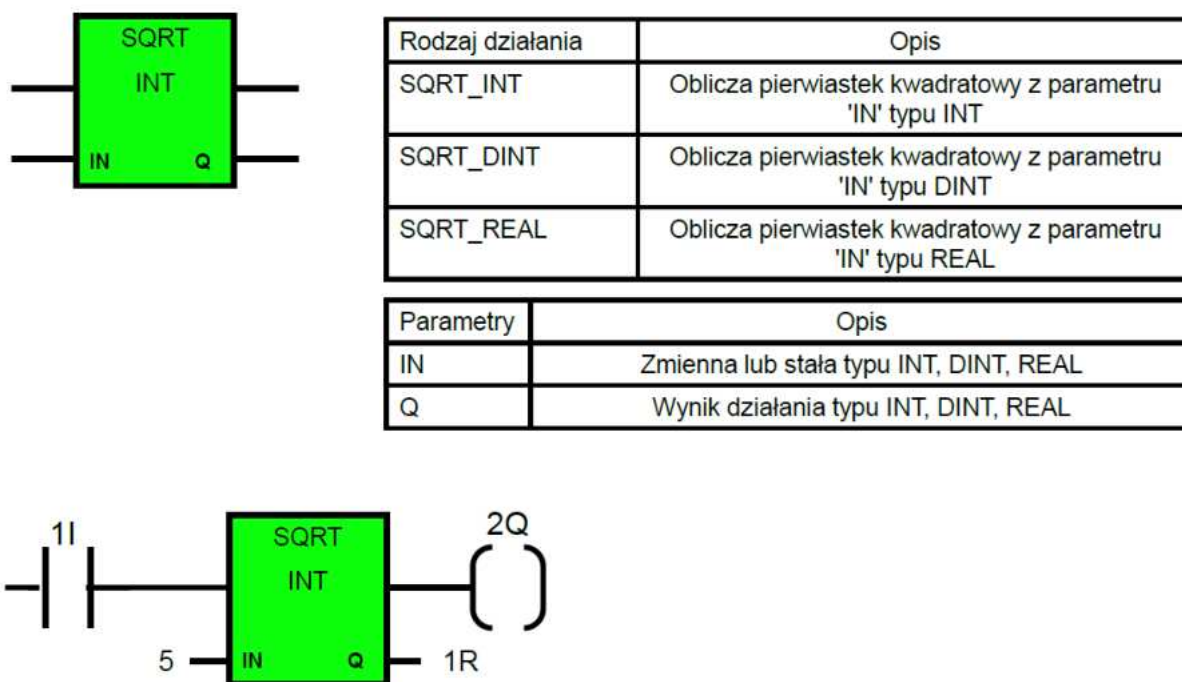
Rys. 5.29 Funkcja matematyczna logarytm LN dla sterownika GE-Fanuc 90 - 30

4. Funkcja wykładnicza EXPT, Rys. 5.30



Rys. 5.30 Funkcja matematyczna wykładnicza EXPT dla sterownika GE-Fanuc 90 - 30

5. Funkcja pierwiastek kwadratowy SQRT, Rys. 5.31



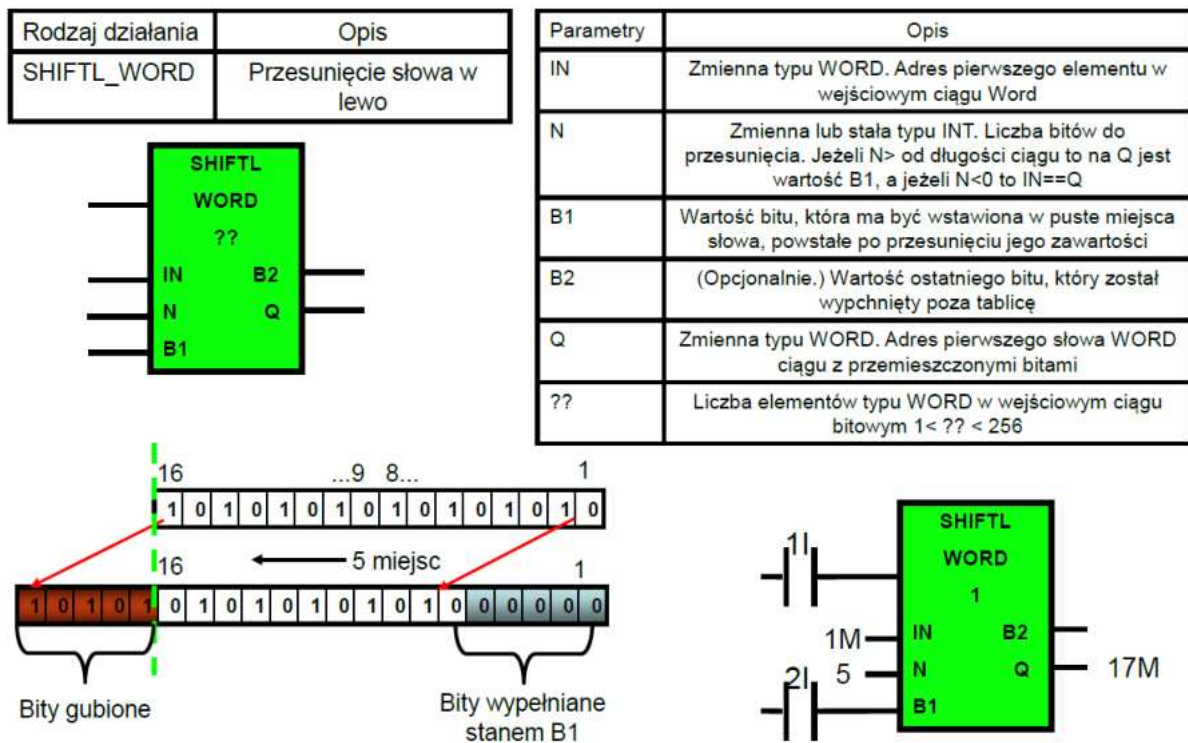
Rys. 5.31 Funkcja matematyczna pierwiastek kwadratowy SQRT dla sterownika GE-Fanuc 90 - 30

6. Skalowanie wartości SCALE, Rys. 5.32



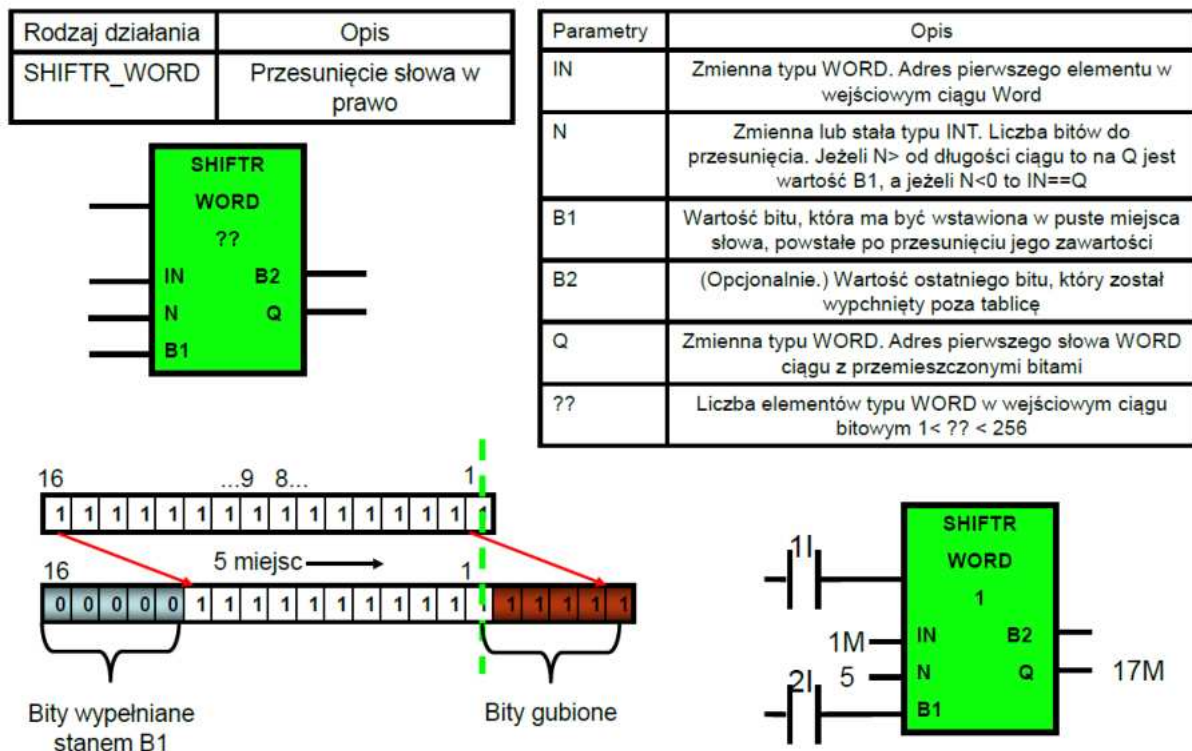
Rys. 5.32 Funkcja skalowanie wartości SCALE dla sterownika GE-Fanuc 90 - 30

7. Funkcje przesuwania w lewo SHIFTL, Rys. 5.33



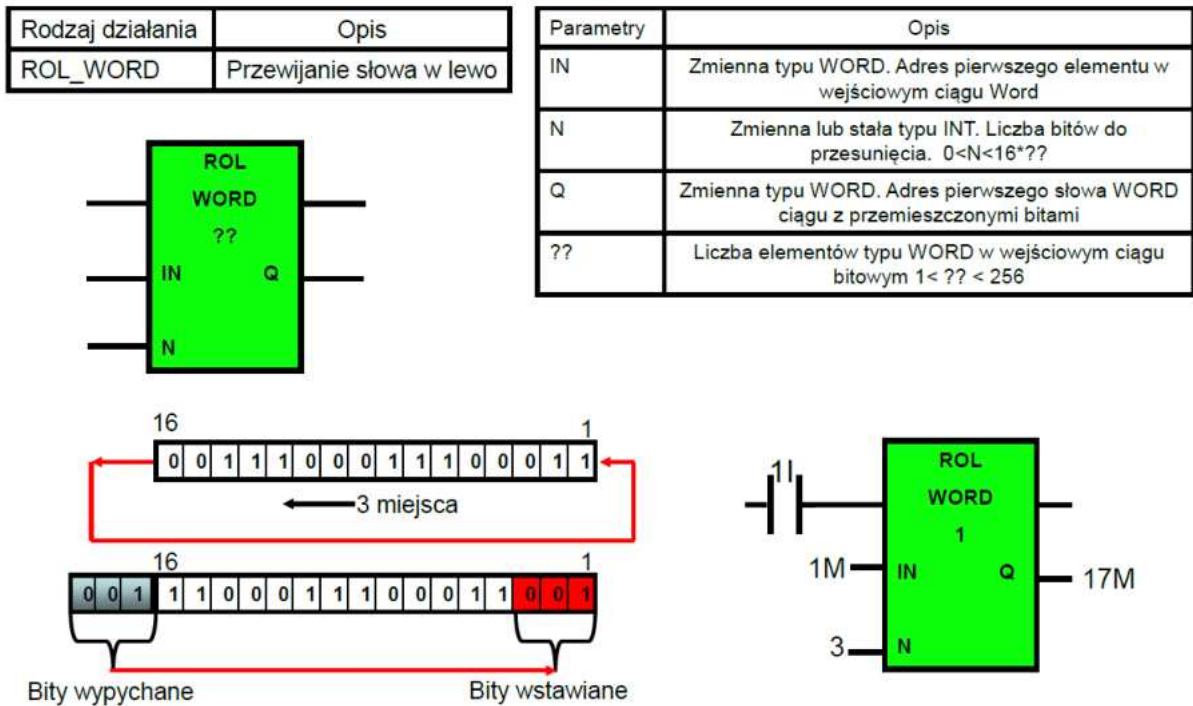
Rys. 5.33 Funkcja przesuwania w lewo SHIFTL dla sterownika GE-Fanuc 90 - 30

8. Funkcje przesuwania w prawo SHIFTR, Rys. 5.34



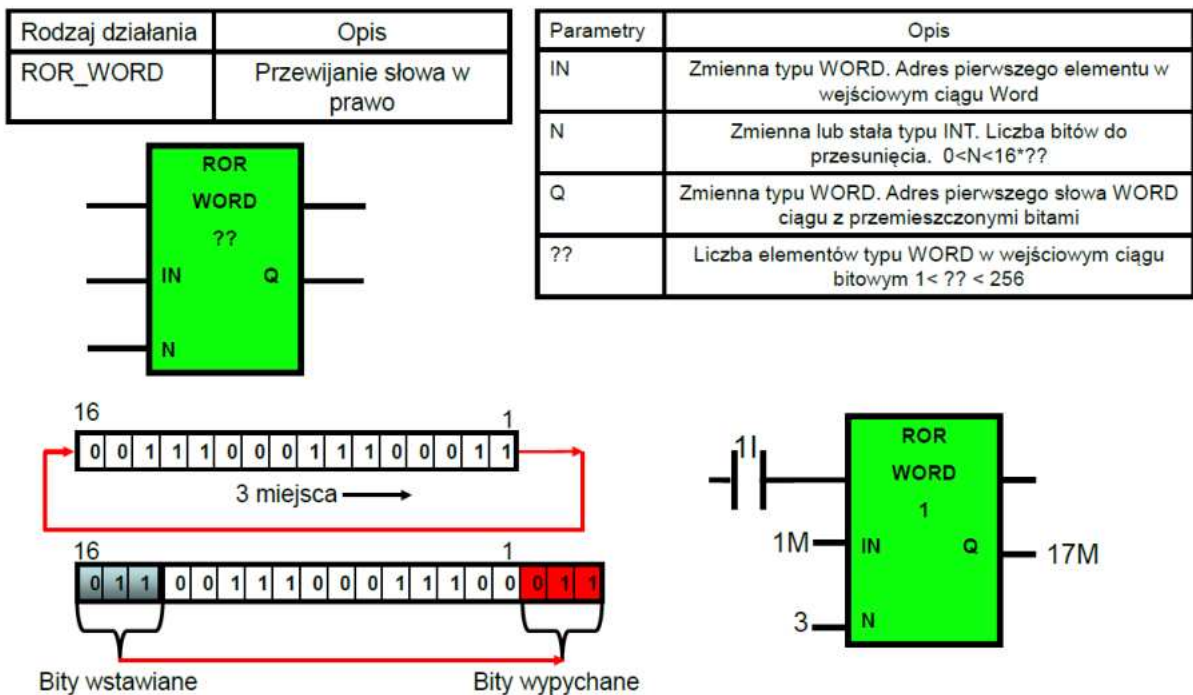
Rys. 5.34 Funkcja przesuwania w prawo SHIFTR dla sterownika GE-Fanuc 90 - 30

9. Funkcje rotacji w lewo ROL, Rys. 5.35



Rys. 5.35 Funkcja rotacji w lewo ROL dla sterownika GE-Fanuc 90 - 30

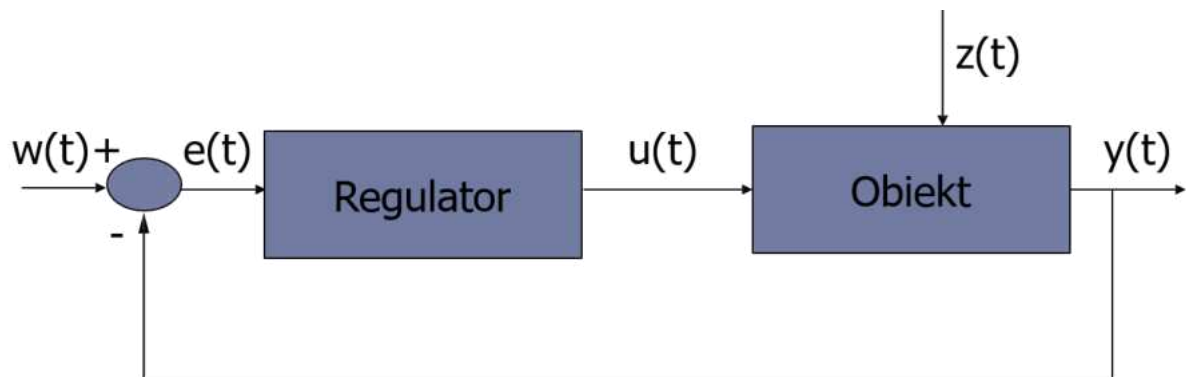
10. Funkcje rotacji w prawo ROR, Rys. 5.36



Rys. 5.36 Funkcja rotacji w prawo ROR dla sterownika GE-Fanuc 90 - 30

5.9 Regulacja PID na sterowniku GE-Fanuc 90-30

Klasyczny układ regulacji przedstawiono na Rys. 5.37.

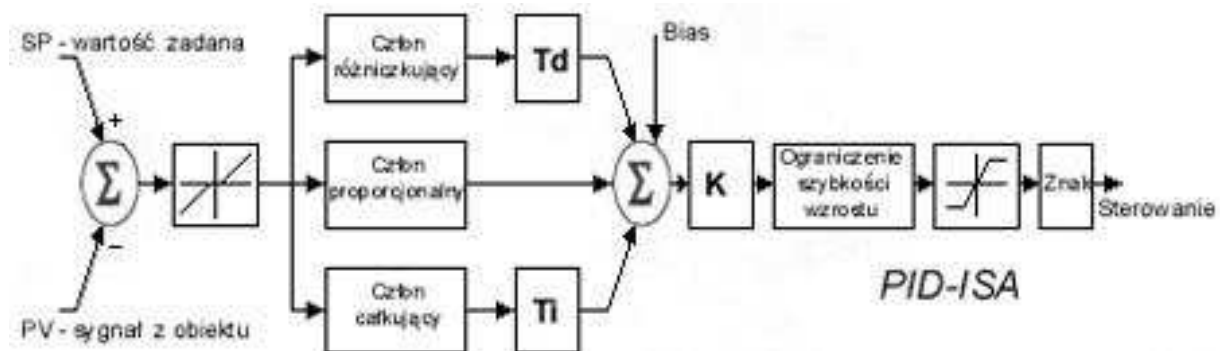


Rys. 5.37 Prosty układ regulacji, $y(t)$ – wielkość wyjściowa (regulowana), $u(t)$ – sygnał sterujący, $w(t)$ – wartość zadana, $e(t)$ – uchyb sterowania, $z(t)$ - zakłócenie

W programowaniu sterowników GE-Fanuc dostępne są dwie wersje regulatorów PID:

- regulator PID_ISA o równaniu standardowym, Rys. 5.38

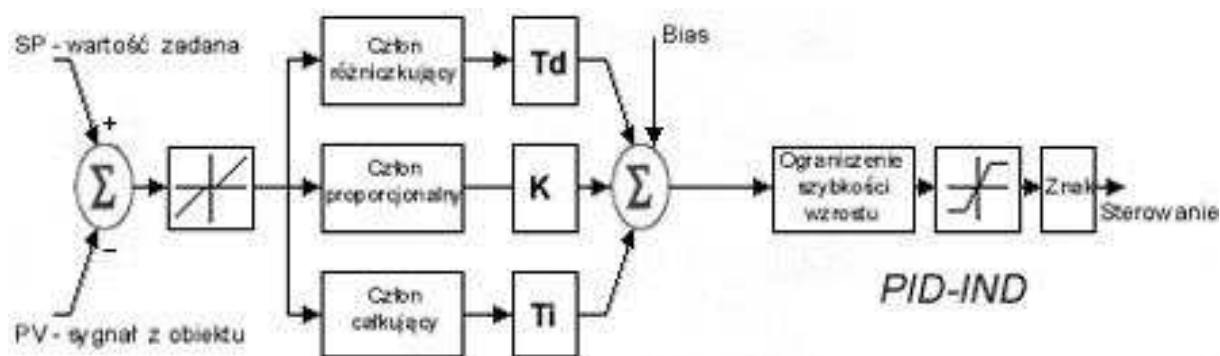
$$y(t) = k_p \left(e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right)$$



Rys. 5.38 Schemat blokowy regulatora standardowego PID-ISA dla sterownika GE-Fanuc 90 - 30

- regulator PID_IND o funkcjach rozdzielonych, Rys. 5.39

$$y(t) = k_p e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt}$$



Rys. 5.39 Schemat blokowy regulatora o funkcjach rozdzielonych PID-IND dla sterownika GE-Fanuc 90 - 30

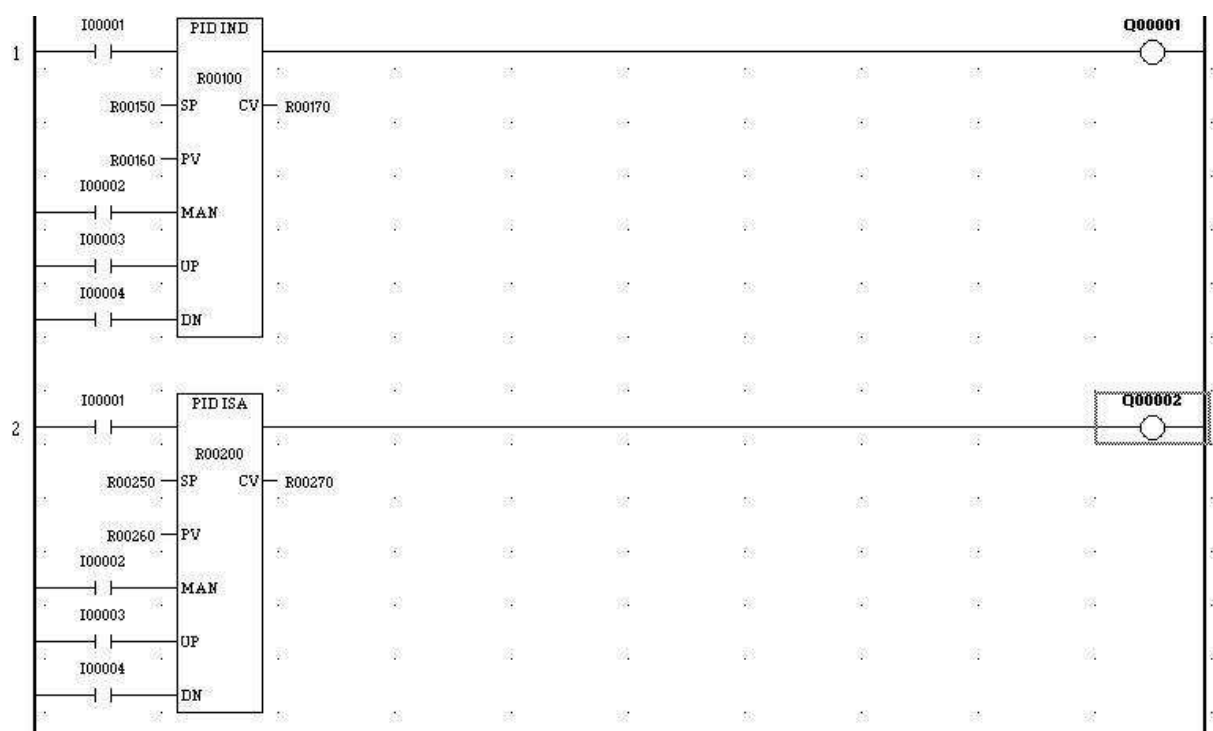
gdzie:

K_p – wzmacnienie regulatora

T_i – akcja całkująca regulatora

T_d – akcja różniczkująca regulatora

Bloki regulatorów w edytorze przedstawia Rys. 5.40.



Rys. 5.40 Regulator PID na sterowniku GE-Fanuc 90-30 w wersji standardowej i zmienionej – program LAD

Tabela 5.5 Parametry regulatora PID

Parametr	Opis
Enable	Sygnał wejściowy zezwolenia. Gdy do bloku funkcyjnego dopływa sygnał, wykonywany jest algorytm PID standardowy lub o niezależnych wyrazach
SP	Wartość zadana wielkości regulowanej (punkt pracy regulatora). Po porównaniu parametrów wielkości regulowanej PV z wielkością zadaną SP blok PID dobiera tak sygnał sterujący CV, aby parametry PV i SP miały taką samą wartość (uchyb zerowy).
PV	Wielkość regulowana, której wartość odczytywana jest ze sterowanego procesu, często jako zmienna typu %AI.
MAN	Parametr MAN o wartości 1 powoduje przełączenie regulatora w ręczny tryb pracy (MANUAL). Wartość tego parametru równa 0 powoduje przełączenie regulatora w automatyczny tryb pracy.
UP	Parametr mający znaczenie tylko w ręcznym trybie pracy. Wartość tego parametru równa 1 powoduje zwiększenie wartości sygnału wyjściowego, wartość 0 nie powoduje żadnego działania.
DN	Parametr mający znaczenie tylko w ręcznym trybie pracy. Wartość tego parametru równa 1 powoduje zmniejszenie wartości sygnału wyjściowego, wartość 0 nie powoduje żadnego działania.
Adres tablicy RefArray	Adres pierwszego z rejestrów, w których przechowywane są wewnętrzne parametry regulatora (parametry użytkownika i parametry wewnętrzne). Obszar ten zajmuje 40 rejestrów typu %R, które nie mogą być wykorzystywane w innym celu.
OK	Sygnał wyjściowy, wysyłany po poprawnym wykonaniu funkcji.
CV	Sygnał sterujący (wyjściowy) procesu, często jest zmienna typu %AQ.

Tabela 5.6 Rejestr regulatora PID – 40 słów w sterowniku GE-Fanuc 90-30

Rejestr	Parametr	Jednostki programowania	Zakres wartości
%Ref+0000	Loop numer – numer układu regulacji	Liczba całkowita bez znaku	0 do 255 – wyłącznie na potrzeby informowania użytkownika
%Ref+0001	Algorithm – rodzaj algorytmu	Wartość ustawiana przez sterownik tylko do odczytu	Nie konfigurowalny
%Ref+0002	Sample Period – okres próbkowania	10 milisekund	0 (każdy cykl) do 65535 (10.9 min). Dla sterowników 90-30 należy wprowadzić co najmniej 1 – szczegółowy opis parametrów bloku funkcyjnego PID
%Ref+0003	Dead Band – górna granica strefy nieczułości	Jednostki bezwymiarowe PV	0 do 32000, wartości ujemne niedozwolone
%Ref+0004	Dead Band – dolna granica strefy nieczułości	Jednostki bezwymiarowe PV	-32000 do 0 wartości dodatnie niedozwolone
%Ref+0005	Proportional Gain Kp – współczynnik wzmocnienia proporcjonalnego	0.01 CV%/PV%	0 do 327.67%/%

%Ref+0006	Derivative Gain Kd – współczynnik wzmocnienia różniczkowego	0.01 sekundy	0 do 327.67 sekundy
%Ref+0007	Integral Rate Ki – współczynnik wzmocnienia całkowego	powtórzeń/1000 sek.	0 do 32.767 powtórzeń/sek.
%Ref+0008	CV Bias/Output Offset – przesunięcie punktu pracy	Jednostki bezwymiarowe parametru CV	-32000 do +32000 wartość dodawana do wyjścia bloku całkującego
%Ref+0009	Upper Clamp – górna granica wartości sygnału wyjściowego	Jednostki bezwymiarowe parametru CV	-32000 do +32000 (>%Ref+10) wartość graniczne sygnału wyjściowego
%Ref+0010	Lower Clamp – dolna granica wartości sygnału wyjściowego	Jednostki bezwymiarowe parametru CV	-32000 do +32000 (<%Ref+9) wartość graniczne sygnału wyjściowego
%Ref+0011	Minimum Slew Time – minimalny czas narastania sygnału wyjściowego	sekundy/pełne przemieszczenie	0 (brak) do 32000sek. do przesunięcia o 32000 CV
%Ref+0012	Config Word - parametr konfiguracyjny	Wykorzystywanych jest 5 bitów	Bit 0 do 2 informują o sposobie obliczania uchybu, polaryzacji sygnału wyjściowego CV i sposobie różniczkowania
%Ref+0013	Manual Command – sygnał sterujący w trybie ręcznym	Jednostki bezwymiarowe parametru CV	Nadąża za parametrem CV w trybie automatycznym lub określa wartość CV w trybie ręcznym
%Ref+0014	Control Word – słowo sterujące	Zapisywany przez sterownik, o ile bit 1 jest <>1	Jeżeli wartość bitu 1 wynosi 0, wartość ta zapisywana jest przez sterownik – szczegółowy opis parametrów bloku funkcyjnego PID
%Ref+0015	Internal SP – punkt pracy regulatora SP	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny
%Ref+0016	Internal CV – sygnał ustawiający CV	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny
%Ref+0017	Internal PV – wartość wielkości regulowanej PV	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny
%Ref+0018	Output – polaryzacja sygnału ustawiający CV	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny
%Ref+0019	Diff Term Storage – dane robocze dotyczące bloku różniczkującego	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny

%Ref+0020 i %Ref+0021	Int Term Storage – dane robocze dotyczące bloku całkującego	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny
%Ref+0022	Slew Term Storage – dane robocze dotyczące prędkości narastania sygnału wyjściowego	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny
%Ref+0023 %Ref+0024 %Ref+0025	Clock – zegar wewnętrzny, czas który upłynął od momentu ostatniego wykonania algorytmu	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny
%Ref+0026	Y remainder Storage – rejestr do przechowywania wartości Y, wewnętrzna zmienna sterownika	Wartość ustawiana przez sterownik, tylko do odczytu	Nie konfigurowalny
%Ref+0027	Lower Range for SP i PV – dolna granica zakresu wartości parametrów SP i PV	Jednostki bezwymiarowe PV	-32000 do +32000 (>%Ref+28) do wyświetlania
%Ref+0028	Upper Range for SP i PV – górna granica zakresu wartości parametrów SP i PV	Jednostki bezwymiarowe PV	-32000 do +32000 (<%Ref+27) do wyświetlania
%Ref+0029 %Ref+0034	Zarezerwowane do użytku wewnętrznego	Tylko do odczytu	Nie konfigurowalny
%Ref+0035 %Ref+0039	Zarezerwowane do użytku wewnętrznego	Tylko do odczytu	Nie konfigurowalny

5.10 Nastawy regulatora PID:

Współczynnik wzmocnienia proporcjonalnego Kp (rejestr 5) - wartość całkowita, oznaczana symbolem Kp (w wersji ISA), określająca zmianę sygnału wyjściowego CV odpowiadającą zmianie uchybu o 100 jednostek bezwymiarowych PV. Parametr ten wyświetlany jest jako 0.00 %/%, z domyślnie przyjmowanymi dwoma miejscami dziesiętnymi.

Przykładowo parametr Kp równy 450 będzie wyświetlany jako 4.50, a powodowane przez niego zmiana sygnału nastawiającego będzie wynosić $Kp \cdot Uchyb / 100$ lub $450 \cdot Uchyb / 100$.

Współczynnik wzmocnienia członu całkującego Ki (rejestr 7) - wartość całkowita ze znakiem określająca częstotliwość całkowania, jeżeli Uchyb ma stałą wartość równą 1 jednostce bezwymiarowej PV. Parametr ten wyświetlany jest w formacie 0.000 powtórzeń/ sek.

Przykładowo, jeżeli parametr Ki ma wartość 1400, wyświetlany będzie jako 1.400 powtórzeń/ sek., a powodowana przez niego zmiana sygnału wyjściowego bloku PID wynosi $Ki \cdot Uchyb \cdot dt$, czyli

$1.400 \cdot 20 \cdot 50 / 1000$, dla uchybu o wartości 20 jednostek bezwymiarowych wartości regulowanej PV i czasu trwania cyklu sterownika 50 milisekund.

Czas różniczkowania Kd (rejestr 6) - wartość całkowita określająca czas różniczkowania w setnych częściach sekundy. Wprowadzana w jednostkach 10 milisekund, a wyświetlana w formacie 0.00 sekund. Przykładowo, jeżeli parametr Kd ma wartość 120, będzie on wyświetlany jako 1.20 sek., a powodowana przez niego zmiana sygnału wyjściowego bloku PID będzie wynosić $Kd \cdot \text{przyrost uchybu} / \text{przyrost czasu}$, czyli $120 \cdot 4 / 3$, jeżeli uchyb zmieni się o 4 jednostki bezwymiarowe PV w ciągu 30 milisekund. Parametr Kd może zostać wykorzystany do przyśpieszenia wolnej odpowiedzi regulatora, ale jest bardzo wrażliwy na zakłócenia wielkości regulowanej PV.

Okno PID Tuning regulatora GE-Fanuc 90-30 przedstawia Rys. 5.42 umożliwia dogodne wprowadzenie nowych parametrów do tabeli rejestrów.

The screenshot shows the 'PID - Project Values' window. It includes the following elements:

- Loop No:** 0
- Manual Command:** 32055
- Control Mode:**
 - ☒ Enable
 - ☐ Up
 - ☐ Override
 - ☐ Down
 - ☐ Manual
- Tuning Parameters:**
 - Proportional:** 20.00 % / %
 - Integral:** 0.100 rep/sec
 - Derivative:** 0.03 sec
- SP/PV Range:** 0
- SP Value:** 40
- Status Indicators:** SP% (0), PV% (0), CV (99%)
- Other Parameters:**
 - Sample Period:** 0.01 sec
 - Dead Band Upper:** 0
 - Upper Clamp (+):** 32000
 - Error Term:** PV-SP
 - Derivative Action:** Error
 - Bias:** 0
 - Dead Band Lower:** 0
 - Lower Clamp (-):** 0
 - Min Slew Time:** 0 sec
 - Output Polarity:** POS
 - Clamp Integral:** ☐
- Buttons:** Update Project, Close, Update Controller, Help >>

Rys. 5.42 Okno doboru parametrów regulatora PID w sterowniku GE-Fanuc

Znaczenie parametrów:

Enable - włączyć

Śledzi wejście enable do instrukcji i jest używane do określenia, czy instrukcja jest aktywna, czy nie.

Override - nadpisanie

Umożliwia zdalne sterowanie instrukcją PID. Gdy bit override jest włączony, instrukcja jest wykonywana na podstawie bieżących wartości manual, up i down, czyli kolejnych trzech pól. Gdy obejście jest wyłączone, wartości manual, up i down są ustawiane na wartości zdefiniowane przez wejścia dyskretne instrukcji.

Bias - wprowadza przesunięcie wyjścia o stałą wartość.

Dead Band Upper/Lower

Wartości całkowite określające górną i dolną granicę strefy nieczułości. Jeśli nie jest wymagana strefa nieczułości, ustaw te dwie wartości na 0. Jeśli ustawiasz granice strefy nieczułości to:

- jeśli błąd PID (SP - PV) lub (PV - SP) mieści się w granicach strefy nieczułości, obliczenia wyjścia regulatora PID będą obliczane z błędem zerowym.
- jeśli ustawisz limity strefy nieczułości, górna strefa nieczułości musi być większa niż 0, a dolna strefa nieczułości musi być mniejsza niż 0; w przeciwnym razie instrukcja PID nie będzie realizowana.
- należy pozostawić limity strefy nieczułości ustawione na 0, dopóki nie zostaną ustawione lub dostrojone wzmocnienia pętli PID, po dostrojeniu może być pożądane dodanie strefy nieczułości, aby uniknąć niewielkich zmian wyjściowych CV z powodu niewielkich zmian błędu, w celu zmniejszenia zużycia mechanicznego elementu wyjściowego.

Upper Clamp and Lower Clamp

Liczby całkowite określające górną i dolną wartość wyjścia CV. Parametry są używane do definiowania limitów na podstawie fizycznych limitów wyjścia CV. Górny limit musi mieć bardziej dodatnią wartość niż Dolny limit; w przeciwnym razie pętla PID nie może działać. Instrukcja PID ma zabezpieczenie przed zerowaniem, aby zmodyfikować wartość całkowania po osiągnięciu maksimum CV. Górny limit jest zwykle ustawiony na +32 767, aby wyjście CV mogło przyjąć pełny zakres wartości.

Error Term – wybiera sposób obliczania błędu

Uchyb = (SP - PV) lub (PV-SP) pozwala wybrać pracę regulatora o kierunku rosnącym lub odwrotnym.

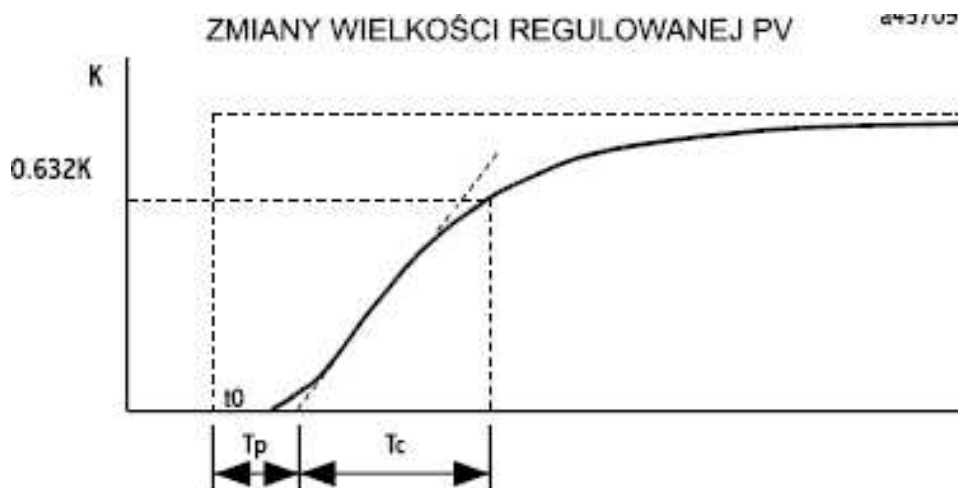
Min Slew Time - minimalny czas narastania

Minimalna liczba sekund, przez którą wyjście CV przechodzi od 0 do pełnego skoku 100% lub 32 767 zliczeń CV. Jest to odwrotny limit szybkości określający, jak szybko można zmienić wyjście CV. Jeśli wartość jest dodatnia, CV nie może zmienić się więcej niż 32 767 CV Counts Delta Time (sekundy) podzielone przez minimalny czas narastania. Na przykład, jeśli okres próbkowania wynosi 2,5 sekundy, a minimalny czas narastania wynosi 500 sekund, CV nie może zmienić się więcej niż $32\,767 \cdot 2,5 / 500$ lub wartość 163 skoku CV na jeden cykl PID. Podobnie jak w przypadku wyjścia CV, istnieje funkcja zapobiegająca zwijaniu się, która reguluje wartość całkowania w przypadku przekroczenia limitu szybkości CV. Jeśli Minimalny czas narastania wynosi 0, nie ma limitu szybkości zmian CV. Minimalny czas narastania należy ustawić na 0 podczas dostrajania lub regulacji wzmocnienia pętli regulatora PID.

Output Polarity – znak wyjścia regulatora PID

5.11 Strojenie regulatora PID

Parametry regulatora należy dobrać do parametrów obiektu regulacji, które wyznacza się z charakterystyki skokowej obiektu według Rys. 5.43. Wyznacza się ją praktycznie.



Rys. 5.43 Odczyt parametrów odpowiedzi obiektu – charakterystyka skokowa

Na podstawie krzywej zmiany wielkości regulowanej PV można wyznaczyć następujące parametry procesu:

Tabela 5.7 Parametry dynamiczne charakterystyki skokowej obiektu regulacji

K	Wzmocnienie w otwartym układzie sterowania procesem = końcowa zmiana wielkości regulowanej PV / do zmiany sygnału sterującego CV w chwili t_0 , zwrócić uwagę na brak indeksu przy współczynniku K.
T_p	Czas opóźnienia procesu lub czas po t_0 , po upływie którego wartość PV zacznie się zmieniać.
T_c	Stała czasowa procesu pierwszego rzędu, czas wymagany aby po czasie T_p wielkość regulowana PV uzyskała 63.3% swojej wartości końcowej.

Po wyznaczeniu trzech parametrów modelu procesu K, T_p i T_c można wstępnie oszacować wartość współczynników wzmocnienia dla zamkniętego układu regulacji PID. Przedstawiona poniżej metoda, ma za zadanie zapewnienie dobrej reakcji systemu przy wzmocnieniach dających stosunek amplitud na wyjściu 1/4. Stosunek amplitud jest to stosunek drugiej wartości szczytowej do pierwszej wartości szczytowej odpowiedzi zamkniętego układu regulacji. Kolejno wyznaczamy:

1. Obliczyć szybkość reakcji regulatora:

$$R = K/T_c$$

2. Jeżeli regulator pracuje jedynie jako regulator proporcjonalny, policzyć współczynnik wzmocnienia wyrazu proporcjonalnego

$$K_p = \frac{1}{R \cdot T_p} = \frac{T_c}{R \cdot T_p}$$

3. Jeżeli regulator pracuje jako proporcjonalny i całkujący, nastawy wyznacza się z równania:

$$K_p = \frac{0.9}{R \cdot T_p} = \frac{0.9 \cdot T_c}{R \cdot T_p}$$

$$K_i = 0.3 \frac{K_p}{T_p}$$

4. Jeżeli regulator pracuje jako proporcjonalny, całkujący i różniczkujący, nastawy wyznacza się z równania:

$$K_p = \frac{G}{R \cdot T_p}$$

gdzie G ma wartość z zakresu od 1.2 do 2.0

$$K_i = 0.5 \frac{K_p}{T_p}$$

$$K_d = 0.5 \cdot K_p \cdot T_p$$

5. Sprawdź, czy okres próbkowania ma wartość z zakresu $(T_p + T_c)/10$ do $(T_p + T_c)/1000$.

Po wyznaczeniu początkowych wartości współczynników wzmocnienia, można je przekonwertować na parametry użytkownika, będące liczbami całkowitymi. W celu uniknięcia problemów ze skalowaniem, wzmocnienie procesu K powinno być obliczane jako zmiana w jednostkach bezwymiarowych wielkości regulowanej PV podzielonej przez skokową zmianę wielkości wyjściowej CV w jednostkach bezwymiarowych CV, a nie w jednostkach inżynierskich parametrów PV czy CV. Również czas powinien być podawany w sekundach. Po wyznaczeniu współczynników K_p, K_i, i K_d, współczynniki K_p i K_d można pomnożyć przez 100 i wprowadzić je jako wartości całkowite, a współczynnik K_i należy pomnożyć przed wprowadzeniem do tabeli parametrów użytkownika %RefArray przez 1000.

Inna metoda 'Idealnego strojenia' ma za zadanie zapewnić najlepszą odpowiedź na zmiany wartości zadanej SP, opóźnionej wyłącznie przez opóźnienie procesu T_p lub czas martwy:

$$K_p = \frac{2 \cdot T_c}{3 \cdot K \cdot T_p}$$

$$K_i = T_c$$

$$K_d = \frac{K_i}{4}$$

jeżeli wykorzystywany jest składnik różniczkujący.

Bazując na własnym doświadczeniu, można obserwować charakterystykę odpowiedzi układu regulacji. Stosownie do obserwowanych zmian korygujemy kolejne nastawy regulatora według podanych punktów:

1. Ustawić wartość wszystkich parametrów regulatora na 0, a następnie górną i dolną wartość sygnału sterującego na odpowiednio maksymalną i minimalną oczekiwaną wartość. Ustawić okres próbkowania na wartość szacowanej globalnej stałej czasowej procesu podzielonej przez liczbę z przedziału od 0 do 100.
2. Wykonać blok w trybie ręcznym a następnie wprowadzić do parametru Sygnał sterujący w trybie ręcznym (%Ref +13) różne wartości w celu sprawdzenia, czy wartość wyjściowa CV

dochodzi do górnej i dolnej wartości granicznej. Zanotować wartości wielkości regulowanej PV dla wybranych wartości wielkości sterującej CV, a następnie przypisać je do wartości zadanej SP.

3. Ustawić małe wzmocnienie K_p , przykładowo $100 \cdot \text{Maksymalna wartość CV} / \text{Maksymalna wartość PV}$, po czym przejść do trybu ręcznego. Zmienić skokowo wartość SP o 2 do 10% wartości maksymalnej wielkości regulowanej PV i obserwować odpowiedź wielkości regulowanej PV. Zwiększyć wartość współczynnika wzmocnienia K_p , jeżeli odpowiedź wielkości regulowanej PV jest zbyt wolna lub też zmniejszyć ją, jeżeli zmiany wartości regulowanej PV są zbyt duże (występują silne oscylacje), bez dochodzenia do stanu stabilności.
4. Po znalezieniu odpowiedniego współczynnika wzmocnienia K_p , zwiększyć wartość K_i w celu znalezienia przeregulowania tłumionego do wartości ustalonej w ciągu 2 do 3 cykli oscylacji. Może to wymagać zmniejszenia współczynnika K_p . Spróbować również zmniejszyć skokową wielkość zmian oraz wprowadzić inne wartości sygnału sterującego CV.
5. Po znalezieniu odpowiednich współczynników K_p i K_i , spróbować wprowadzić współczynnik K_d w celu szybszego uzyskiwania odpowiedzi na parametry wejściowe, jednak bez powstawania oscylacji. Współczynnik K_d nie jest konieczny, nie sprawdza się w każdym procesie oraz nie powinien być wykorzystywany w przypadku częstych zakłóceń sygnału wielkości regulowanej PV.
6. Sprawdzić współczynnik wzmocnienia dla różnych wartości zadanych SP, oraz jeżeli jest to potrzebne, określić strefę nieczułości i minimalny czas narastania sygnału wyjściowego. W przypadku niektórych procesów konieczne może być ustawienie bitów słowa konfiguracyjnego w celu zmiany znaku uchybu lub polaryzacji.

6. Programowanie sterownika Siemens SIMATIC S7-1200

Zestaw sterownik Siemens SIMATIC S7-1200 przedstawiono na **Błąd! Nie można odnaleźć**



Rys. 6.1 Zestaw Sterownika Siemens SIMATIC S7-1200

źródła odwołania..

Sterownik S7-1200 składa się z modułu procesora, zintegrowanego zasilacza, obwodów wejściowych oraz obwodów wyjściowych umieszczonych w zwartej obudowie, co łącznie tworzy bardzo wydajny sterownik PLC. Wyposażony jest w port PROFINET umożliwiający komunikację poprzez sieć PROFINET/Ethernet. Dostępne są również moduły komunikacyjne pozwalające na łączność poprzez interfejsy RS485 i RS232.

Sterownik jest układem do sterowania dowolnym procesem produkcyjnym. Konfiguruje się go z modułów, które dopasowują wejścia/wyjścia do planowanej linii produkcyjnej. Ważnym elementem w sterowanie jest programowanie sterownika.

Oprogramowanie Totally Integrated Automation (TIA Portal) dostarcza narzędzi do zarządzania i konfigurowania wszystkich urządzeń w systemie, takich jak sterowniki PLC i urządzenia HMI. Elementem składowym TIA Portal jest STEP 7 Basic pozwalający programować sterowniki S7-1200 w językach programowania LAD, FBD i inne. TIA Portal zawiera również narzędzia do podłączania i konfigurowania urządzeń HMI.

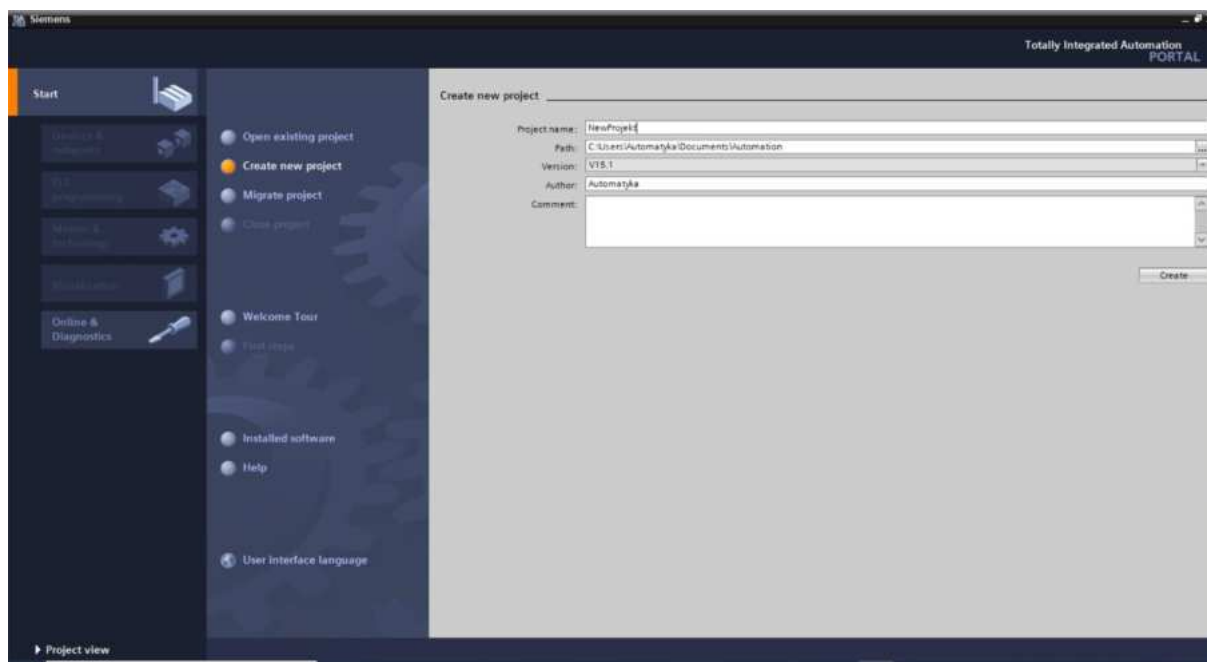
6.1 Tworzenie projektu w programie TIA PORTAL V15

1. Uruchomienie programu TIA PORTAL V15, Rys. 6.2.



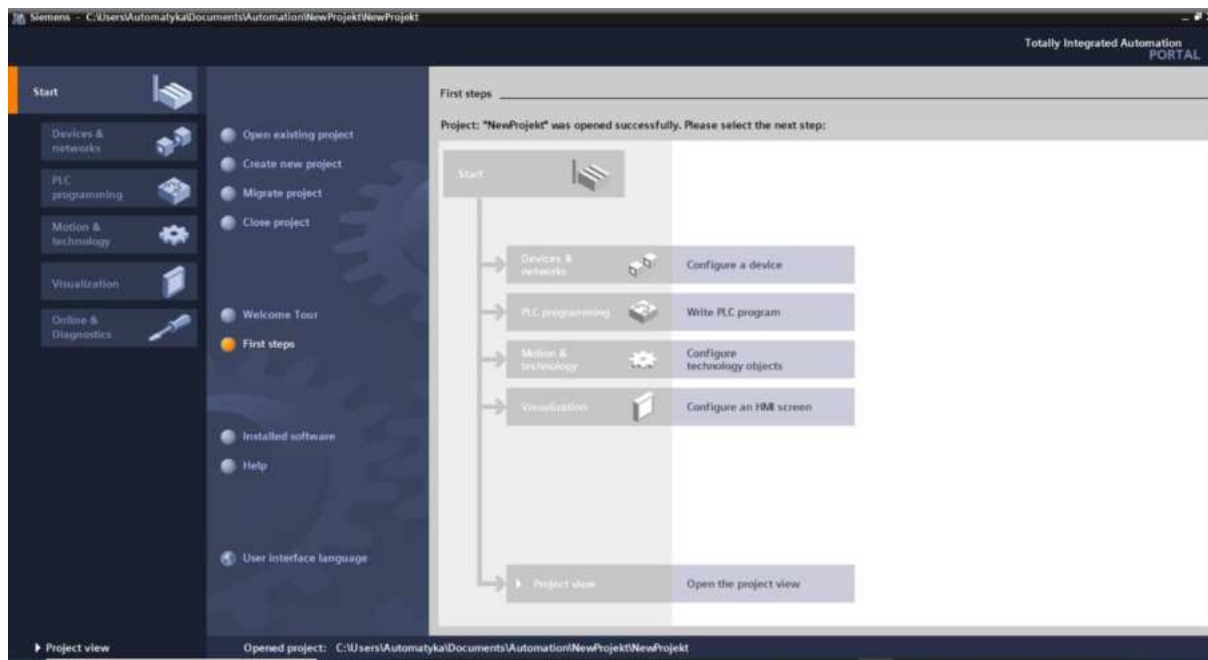
Rys. 6.2 Start programu TIA Portal

2. Menu programu - wybierz punkt tworzenie nowego programu, a następnie nadaj mu nazwę oraz ścieżkę z katalogiem docelowym. Wskazane jest oddzielny katalog do pojedynczego projektu, Rys. 6.3.



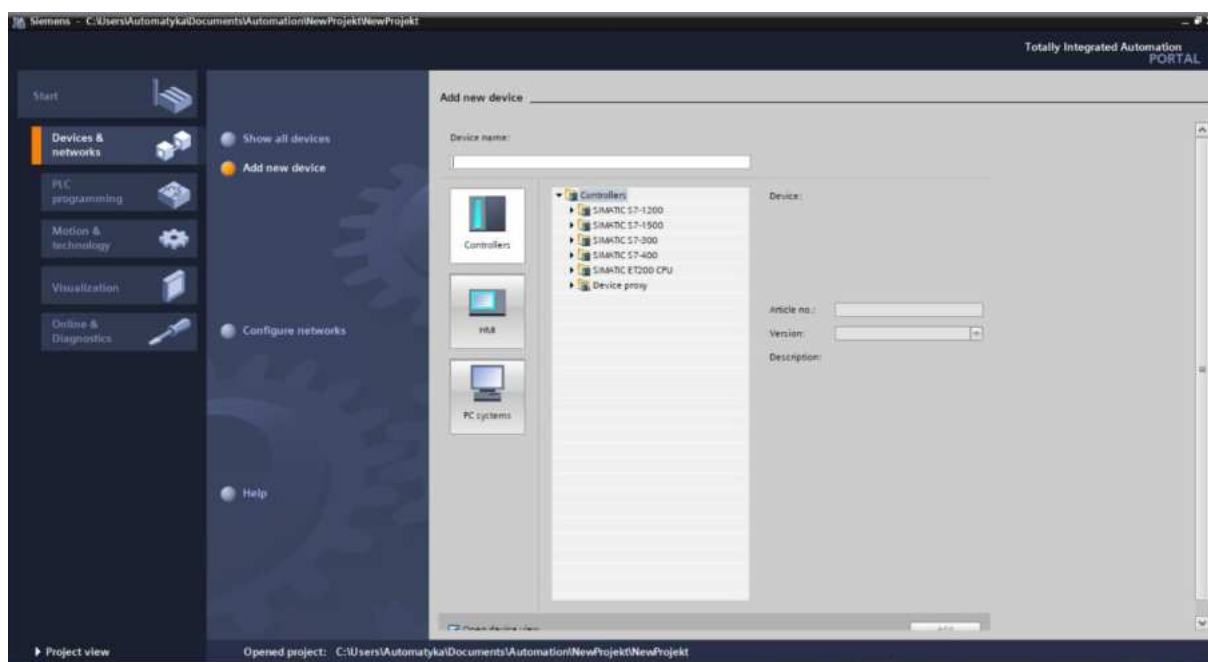
Rys. 6.3 Strona startowa programu TIA Portal

3. Kliknij tworzenie projektu, a następnie skonfiguruj urządzenie, Rys. 6.4.



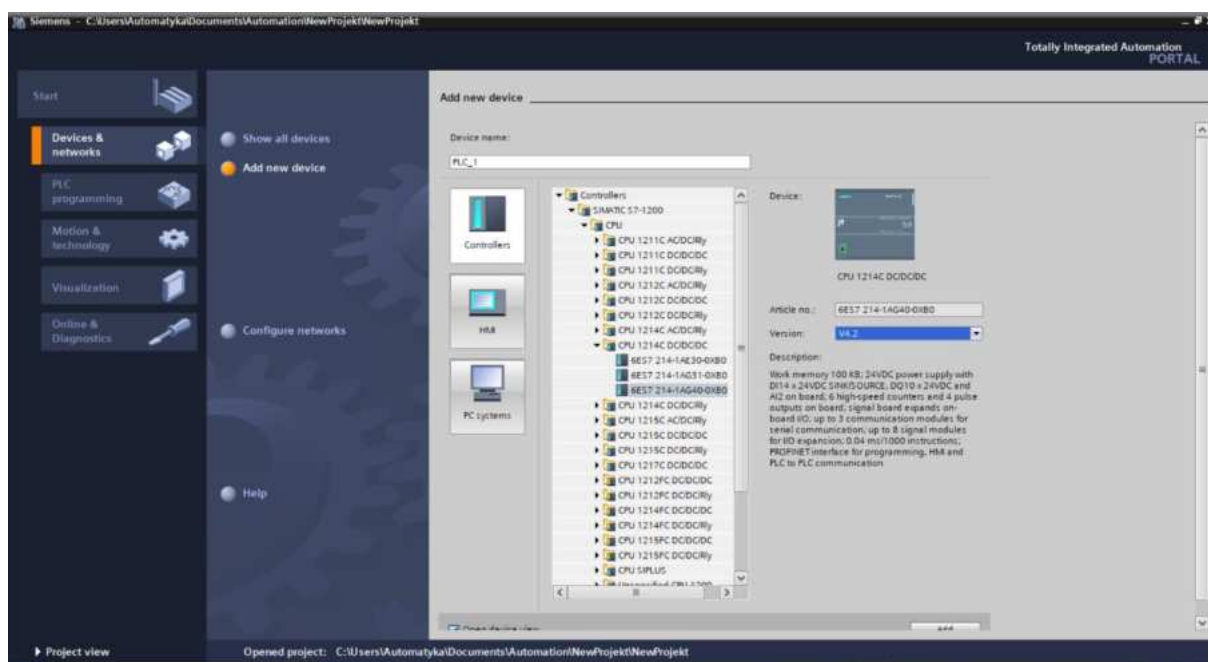
Rys. 6.4 Strona z menu tworzenia projektu i konfiguracji sterownika

4. Wybierz punkt – dodaj nowe urządzenie, Rys. 6.5.



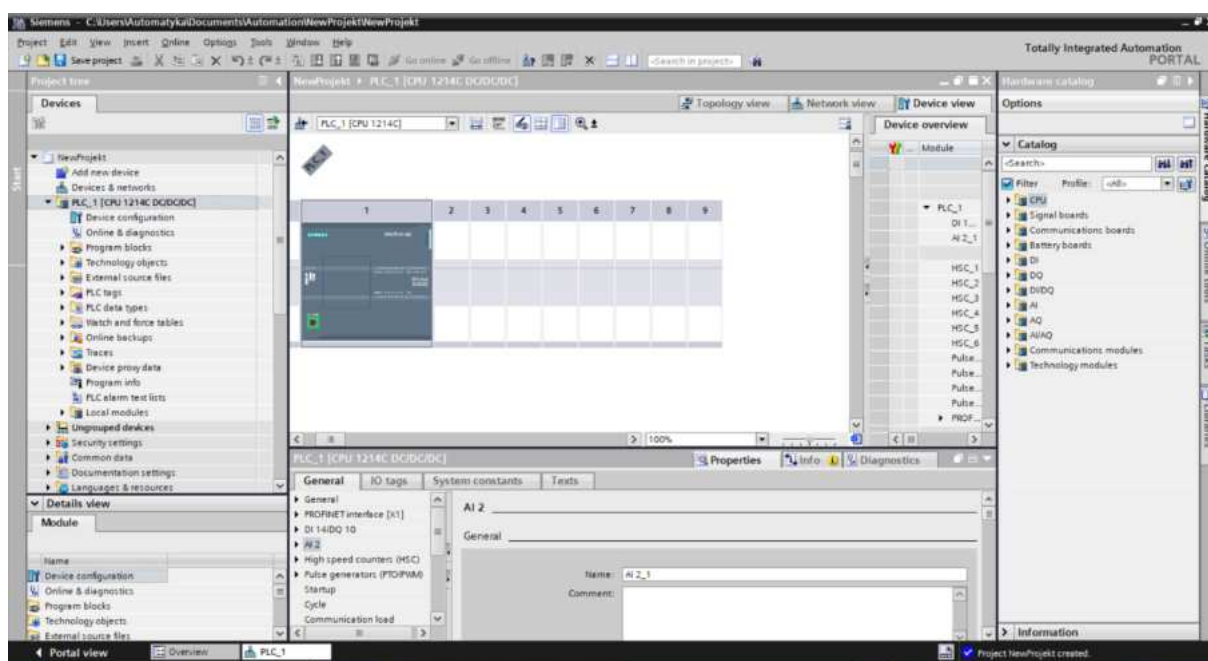
Rys. 6.5 Strona menu z wyborem sterownika

5. Dostępny jest sterownik z procesorem **CPU 1214C DC/DC/DC**, wersja **6ES7 214-1AG40-0XB0**, ostatecznie zatwierdzamy przez 'dodaj', Rys. 6.6.



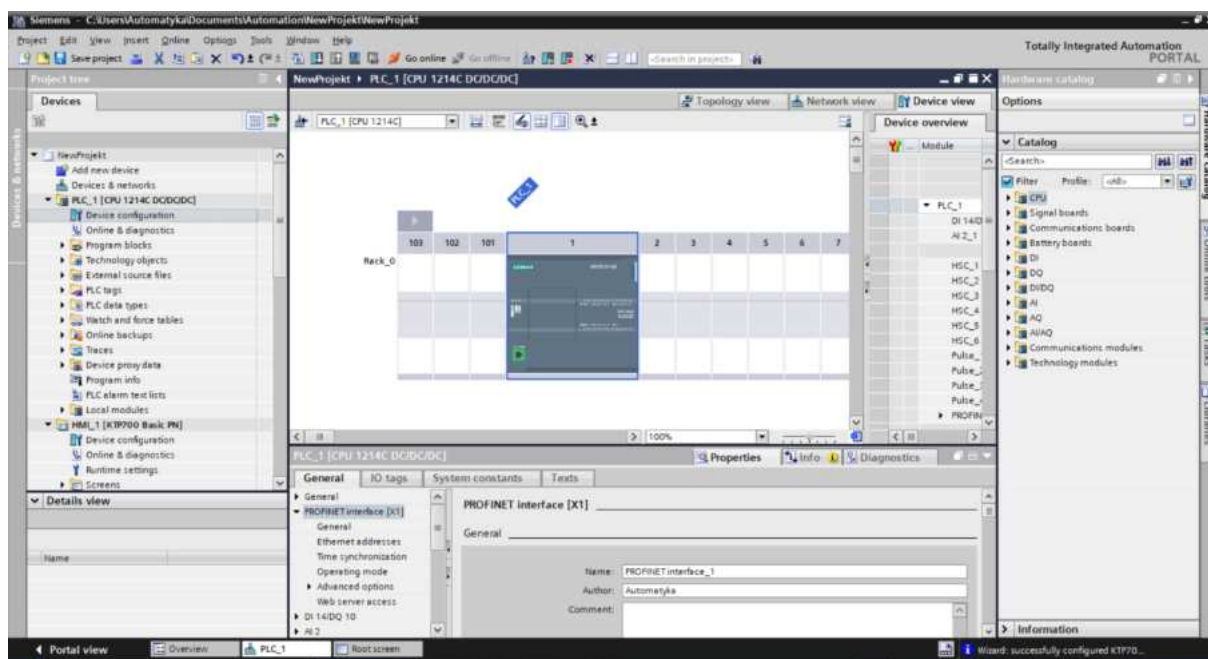
Rys. 6.6 Strona z wyborem sterownika, dostępny w laboratorium

6. Program przechodzi do pełnego ekranu programu i możliwe jest rozpoczęcie programowania, Rys. 6.7.



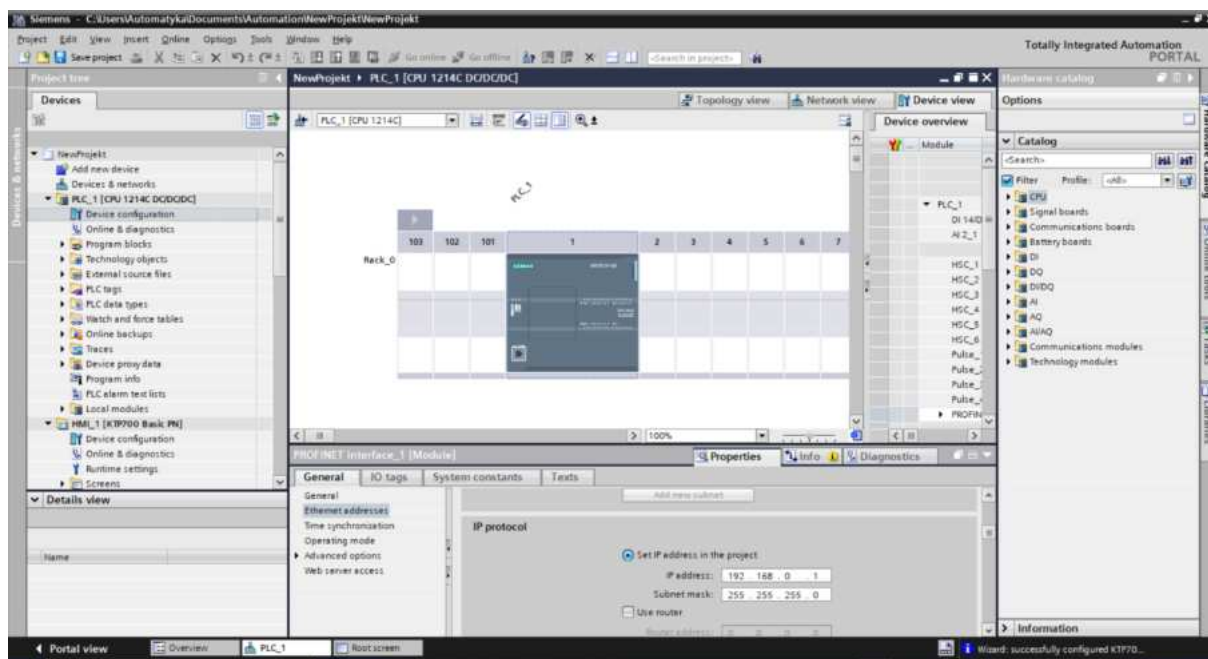
Rys. 6.7 Start programu TIA Portal

7. Sterownik wymaga skonfigurowania adresu internetowego. Aby skonfigurować port PROFINET należy wybrać z okna **Project tree** pole **Device configuration**, Rys. 6.8.



Rys. 6.8 Konfiguracja portu komunikacyjnego PROFINET

Następnie w oknie dialogowym **Device view** zaznaczamy zielony prostokąt PROFINET. W znajdującym się poniżej oknie **Properties** wybieramy pole konfiguracji **Ethernet address** i konfigurujemy adres sieciowy w oknie poniżej, Rys. 6.9.



Rys. 6.9 Wprowadzenie nowego adresu sieciowego w TIA Portal

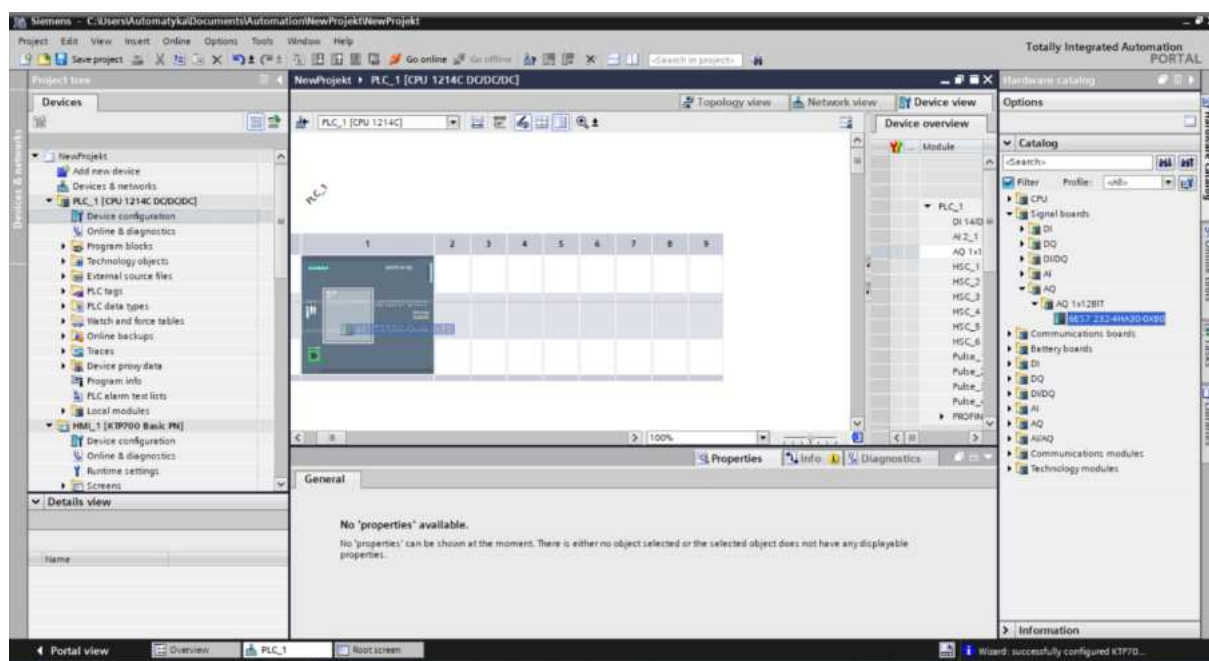
TIA Portal wyświetla okno dialogowe konfiguracji adresu Ethernet, które pozwala powiązać program zawierający projekt z adresem IP tego CPU, które otrzyma ten projekt. **Adres IP jest ustalany w chwili wczytania ustawień do CPU.**

CPU nie ma wstępnie ustawionego adresu IP. Użytkownik powinien w trakcie konfiguracji CPU ręcznie wpisać adres IP urządzenia w programie. Pomaga to uniknąć konfliktu adresu IP, gdyż wszystkie urządzenia w sieci PROFINET muszą mieć swój indywidualny adres IP. Jeżeli CPU jest połączona z routerem sieciowym, to należy również wpisać adres IP routera.

Pierwszą częścią adresu IP jest **Network ID** (który identyfikuje sieć w jakiej znajduje się dane urządzenie). Drugą częścią adresu IP jest **Host ID** (unikalny dla każdego urządzenia w danej sieci). Adres IP 192.168.x.y jest standardowo rozpoznawany jako sieć prywatna, która nie jest dostępna w Internecie. Maska podsieci (Subnet Mask) zaktualizuje się automatycznie po kliknięciu jeden raz w polu deklaracji adresu maski podsieci.

8. Sterownik laboratoryjny wyposażony jest w jeden port wyjściowy, który należy dodać do sterownika, po wciśnięciu w oknie inspektora w menu sterownika PLC – konfiguracja urządzenia, następnie przechodzimy do trzeciego okna katalogowego, rozwijamy punkt **Signal Boards** wyjście AQ pojedyncze 12Bit i myszą przeciągamy na sterownik, Rys. 6.10.

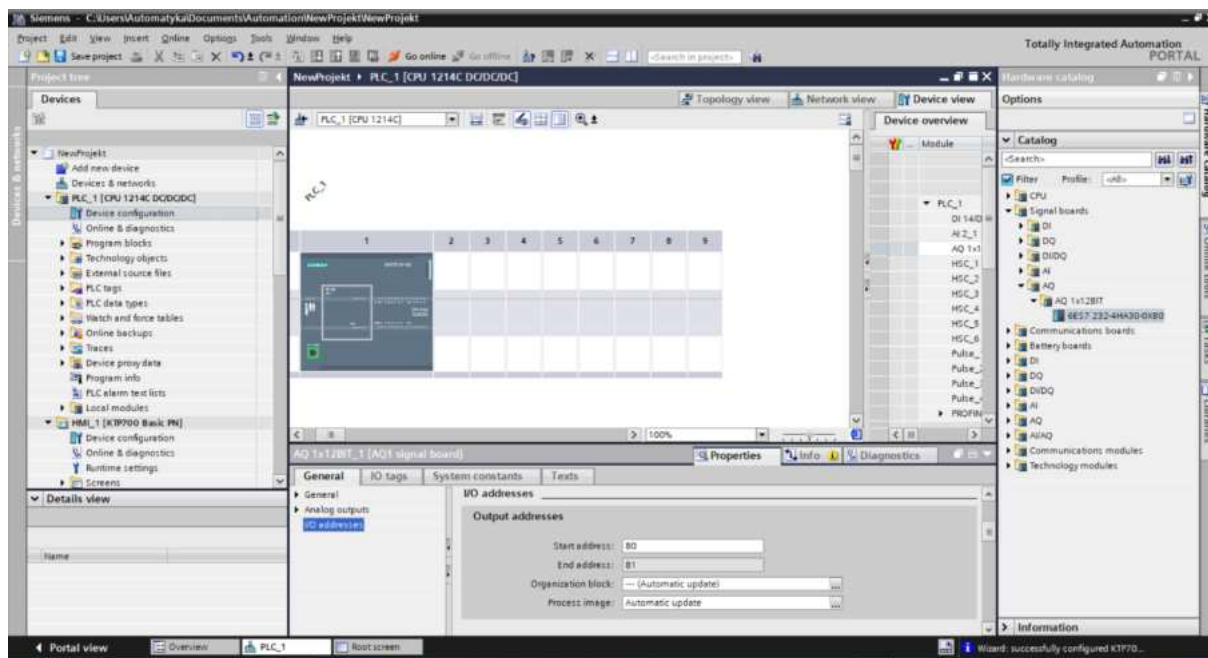
W ten sposób dodaje się moduł sterownika z katalogu firmy i kolejne moduły, jeżeli są w zestawie.



Rys. 6.10 Składanie zestawu sterownika z modułów w TIA Portal

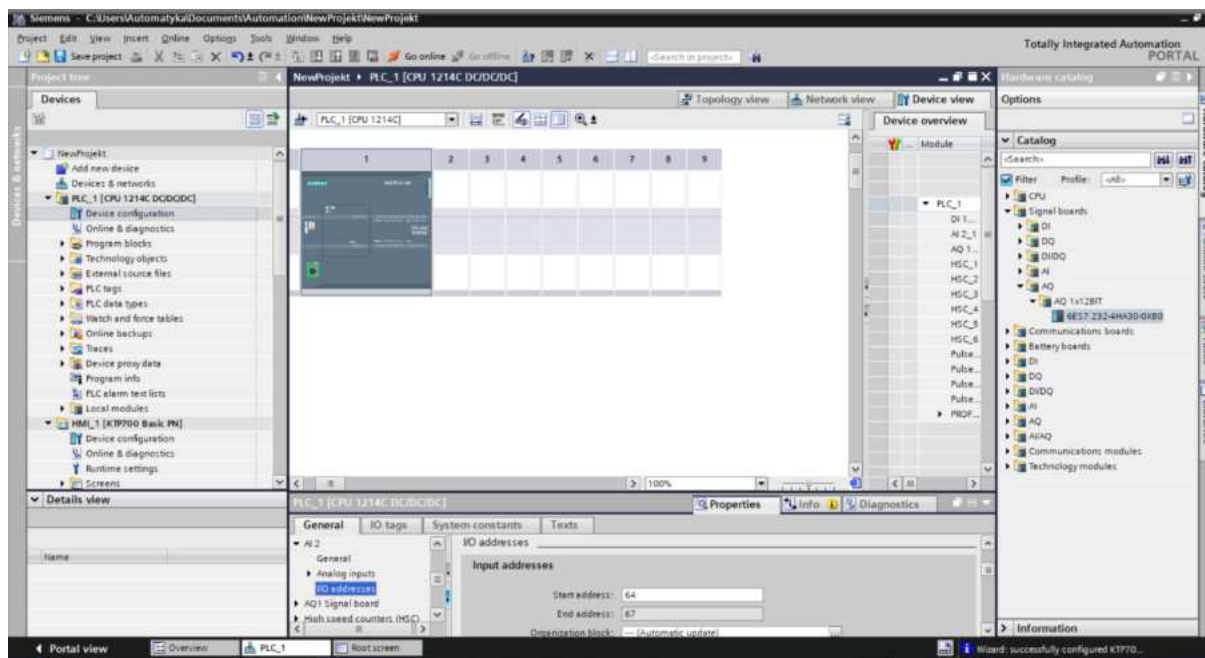
9. Analogowy port wyjściowy %AQ konfigurujemy poprzez podwójne kliknięcie jego myszą w oknie poniżej, Rys. 6.11.

Wyjście może być prądowe lub napięciowe i ma przykładowy adres %QW80 i zajmuje dwa bajty o numerach 80 i 81.



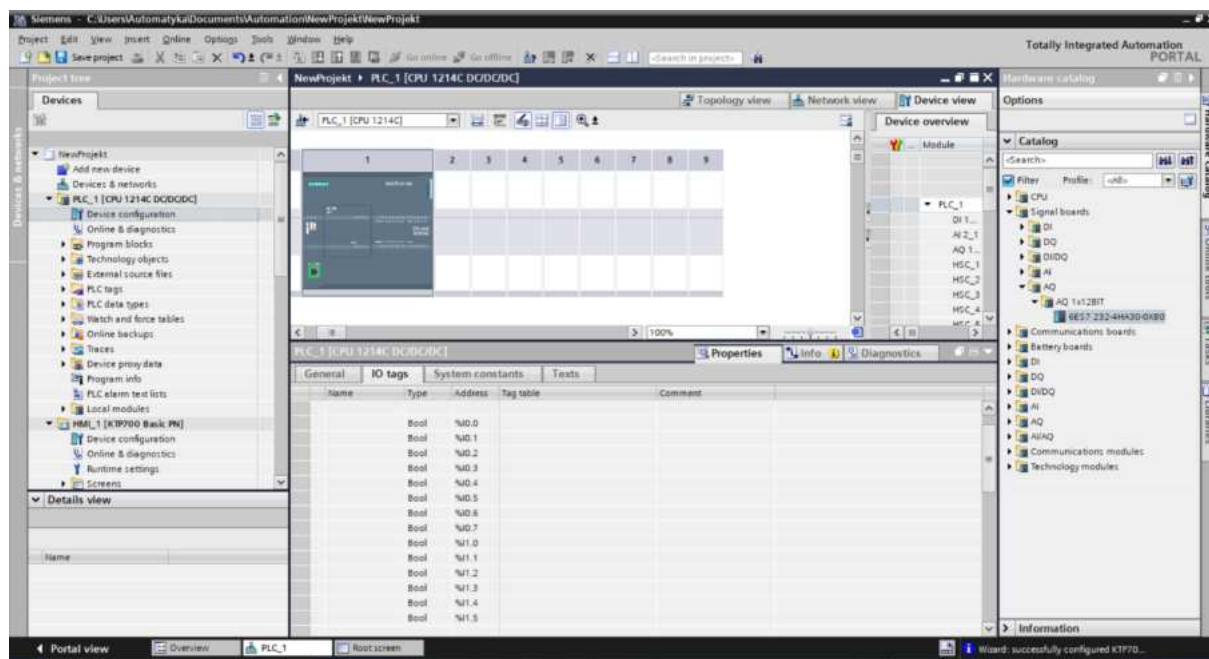
Rys. 6.11 Ustalenie parametrów analogowego portu wyjściowego %QW80

10. Analogowe adresy wejściowe na sterowniku znajdziemy po kliknięciu samego sterownika i jego konfiguracji. Adresy wejść analogowych są następujące %IW64 i %IW66, Rys. 6.12.



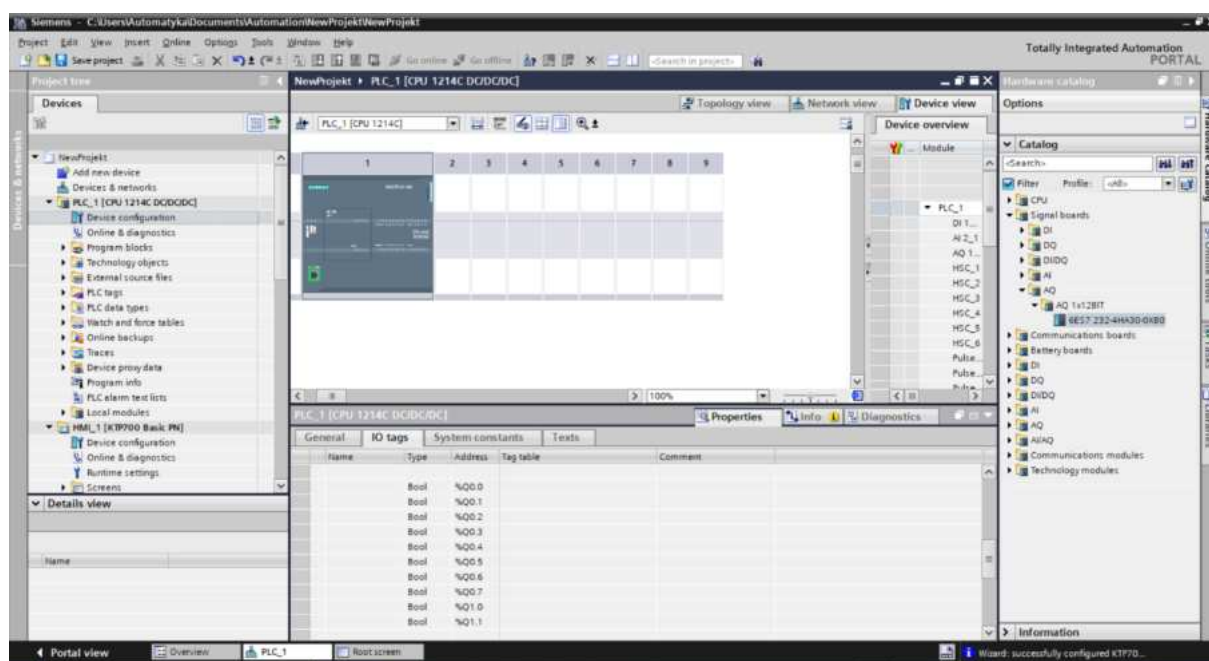
Rys. 6.12 Ustalenie parametrów analogowego portu wejściowego %IW64

Podobnie postępujemy z adresami binarnymi. Wejść binarnych jest 14 i mają adresy od %I0.0 do %I0.7 i od %I1.0 do %I1.5, Rys. 6.13.



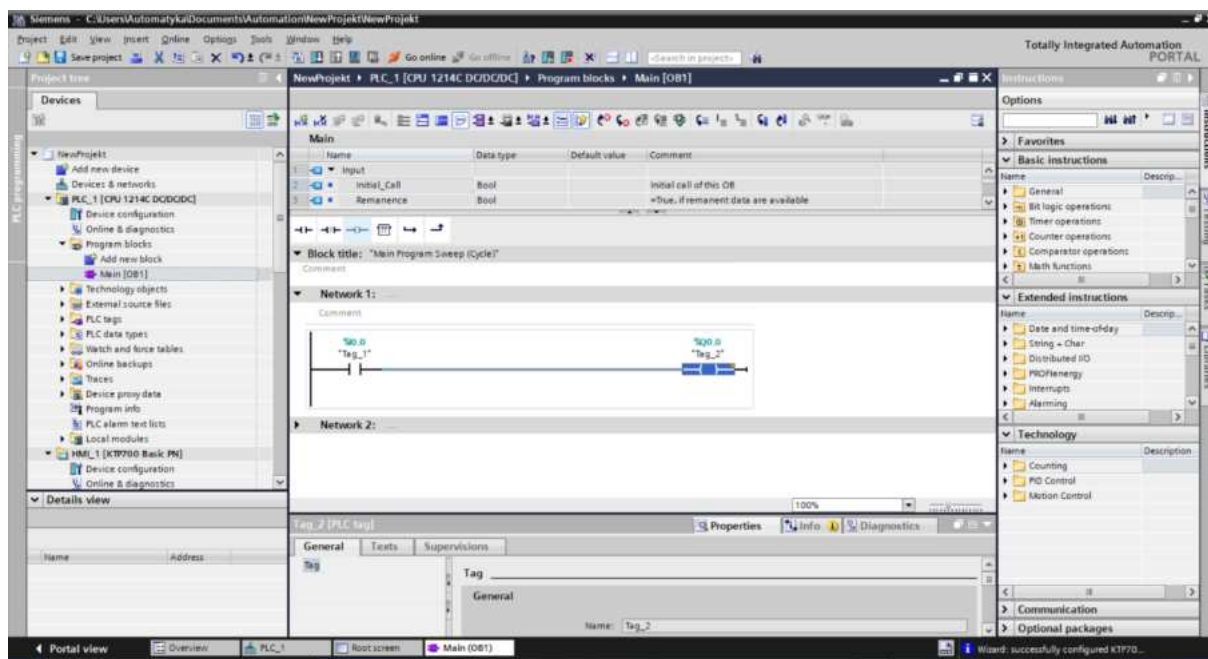
Rys. 6.13 Ustalenie adresów wejść binarnych w TIA Portal

Wyjść binarnych w wybranym modelu jest dziesięć i mają adresy od %Q0.0 do %Q0.7 i od %Q1.0 do %Q1.1, Rys. 6.14.



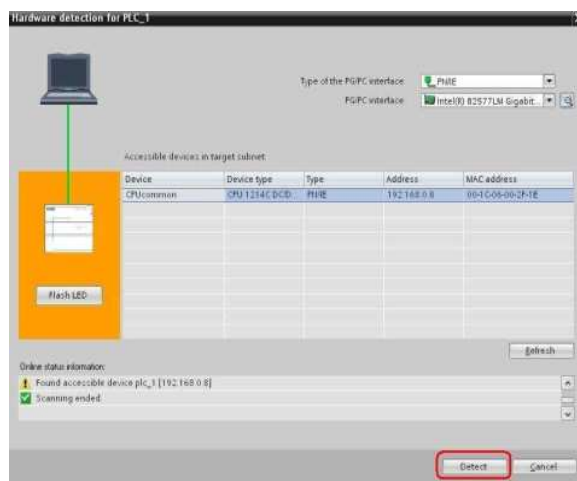
Rys. 6.14 Ustalenie adresów wyjść binarnych w TIA Portal

11. Sterownik został skonfigurowany, należy przejść do jego programowania według analizy obiektu sterowania, wybieramy w oknie inspektora **Program Blok** i podpunkt **Main**, Rys. 6.15.



Rys. 6.15 Zapis programu sterownika w TIA Portal

Przed wgraniem programu należy zidentyfikować sterownik, który planujemy programować. W tym celu klikamy ikonę sieciową Accessible devices i zaznaczamy sterownik z odpowiednim adresem MAC lub IP, Rys. 6.16. Każde urządzenie znajdujące się w sieci PROFINET posiada indywidualny adres MAC. Aby sprawdzić czy wybrano prawidłowy sprzęt, zaznacz urządzenie (właściwy adres MAC) i wciśnij **Flash LED**. Diody LED statusu operacyjnego CPU powinny migać na wybranym sterowniku.

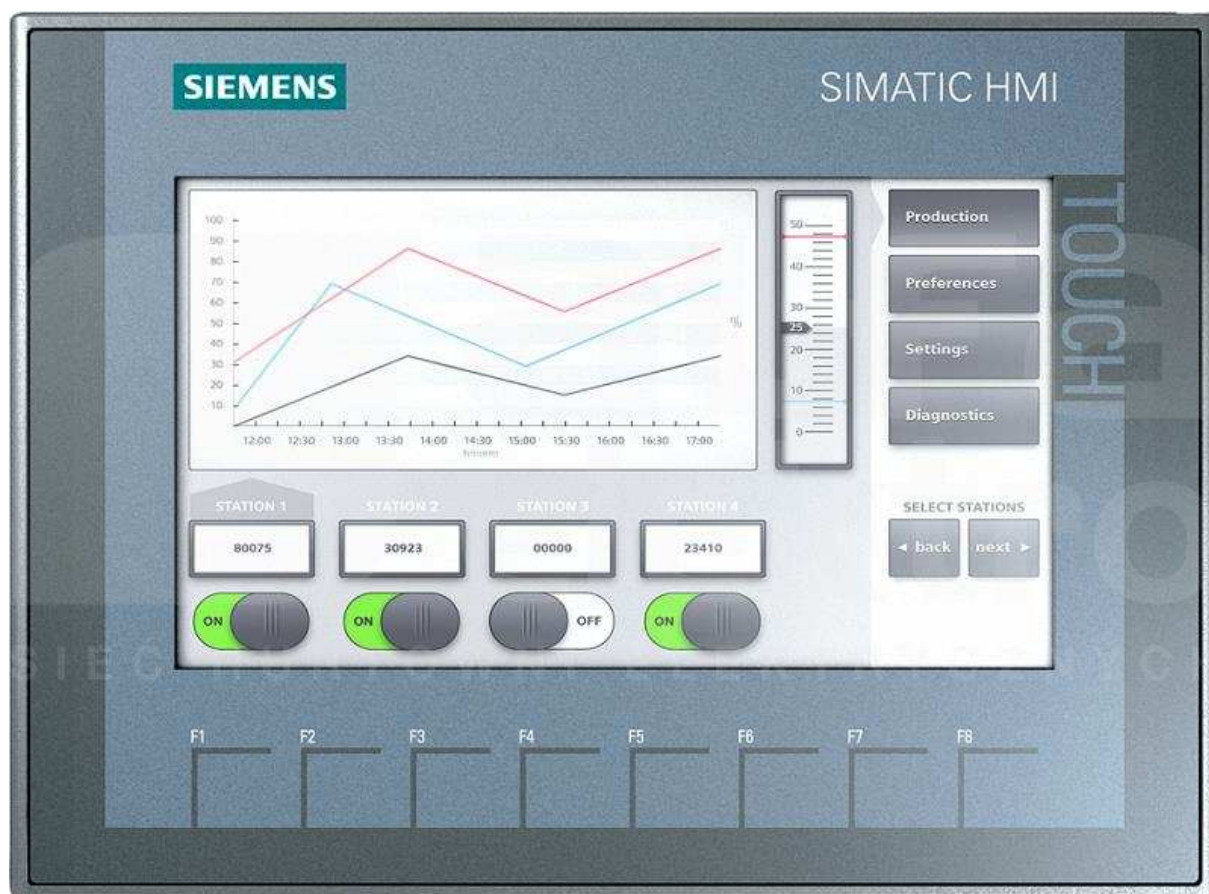


Rys. 6.16 Wybór i kontrola sterownika w sieci

Po zidentyfikowaniu urządzenia możemy przejść do wgrania programu na sterownik przez wciśnięcie ikony **Download to device** i wybierając to samo urządzenie.

6.2 Dodawanie panelu operatorskiego do projektu w TIA PORTAL v15

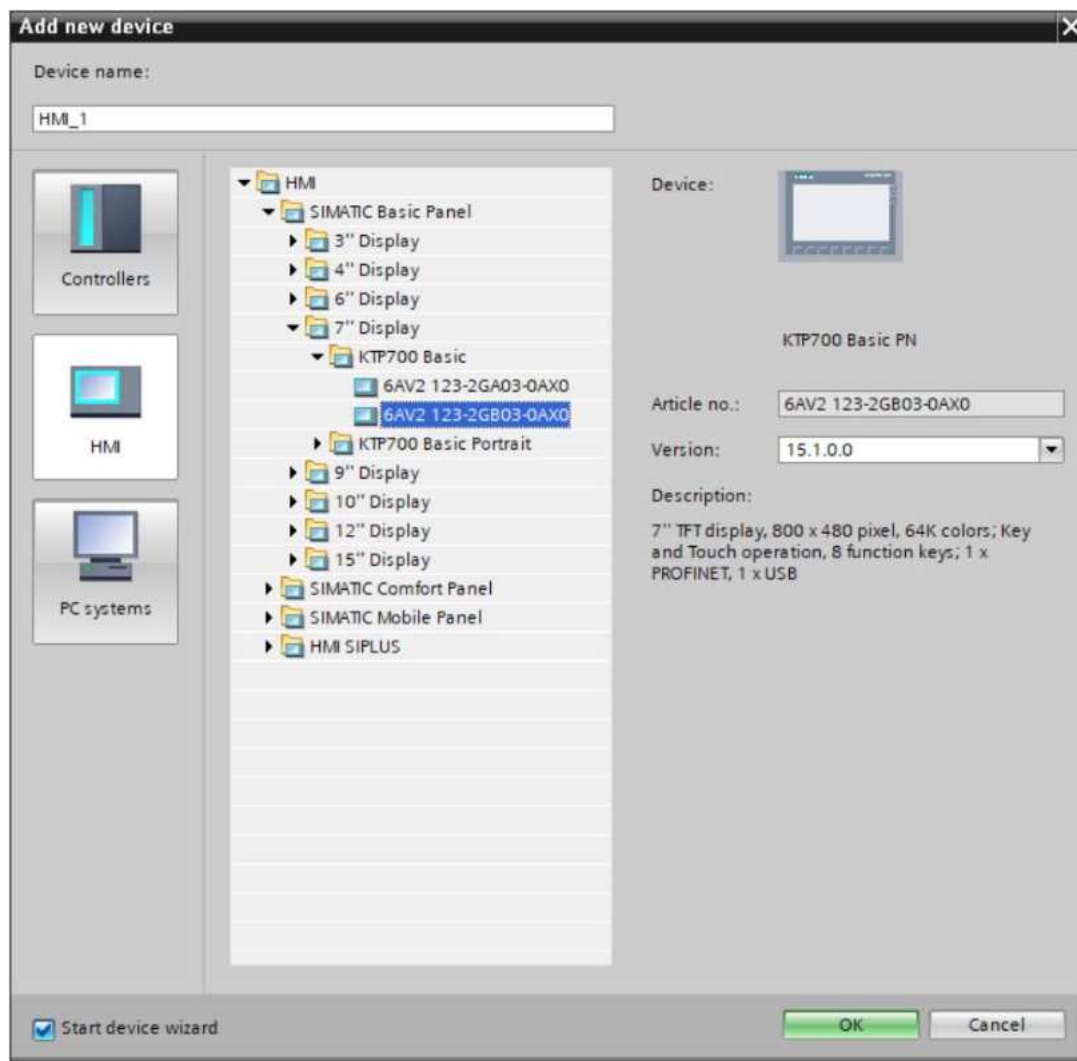
Kolejne urządzenie ważne w pracy ze sterownikiem to panel operatorski HMI, przykładowy panel KPT700 Basic PN przedstawiono na Rys. 6.17.



Rys. 6.17 Panel HMI KPT700 Basic PN

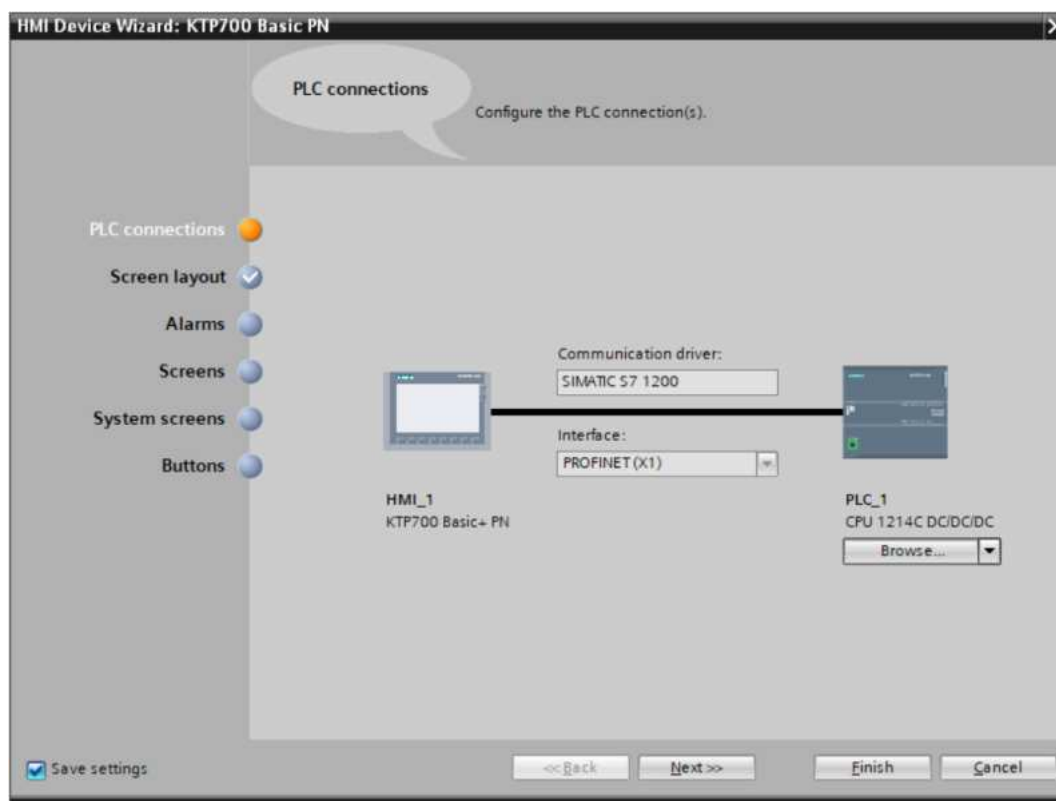
Panel operatorski może być wykorzystany do celów wizualizacji i sterowania nadrzędnego procesu technologicznego. Dotykowy panel operatorski KTP 700 Basic PN z rodziny SIMATIC HMI należy do paneli operatorskich serii „Basic Panels”. Panel ten przeznaczony jest do małych aplikacji przemysłowych, gdy nie ma konieczności korzystania ze skryptów i archiwowania zmiennych. Panel posiada ograniczoną funkcjonalność HMI, ale umożliwia realizację takich funkcji stacji operatorskich, jak tworzenie systemów alarmowych, wykorzystywanie receptur czy wyświetlanie trendów. Komunikację z panelami umożliwiają złącza RS422/485 (wersja „Basic DP”) lub RJ45-Ethernet (wersja „Basic PN”). Ta ostatnia przeznaczona jest do komunikacji ze sterownikami rodziny SIMATIC S7 z portem komunikacyjnym Profinet. Serię „Basic Panels” opracowywano z myślą o wykorzystaniu ze sterownikami SIMATIC S7-1200, można też stosować do współpracy ze sterownikami S7-300/400. Do wykonania przykładowej aplikacji służy środowisko programowe TIA Portal v15.

W oknie inspektora projektu wybieramy: dodaj nowe urządzenie typu panel operatorski – wyświetlacz podstawowy 7” **KPT700 Basic** o numerze 6AV2 123-2GB03-0AX0 i zatwierdzamy, Rys. 6.18.



Rys. 6.18 Wybór panelu operatorskiego TIA Portal

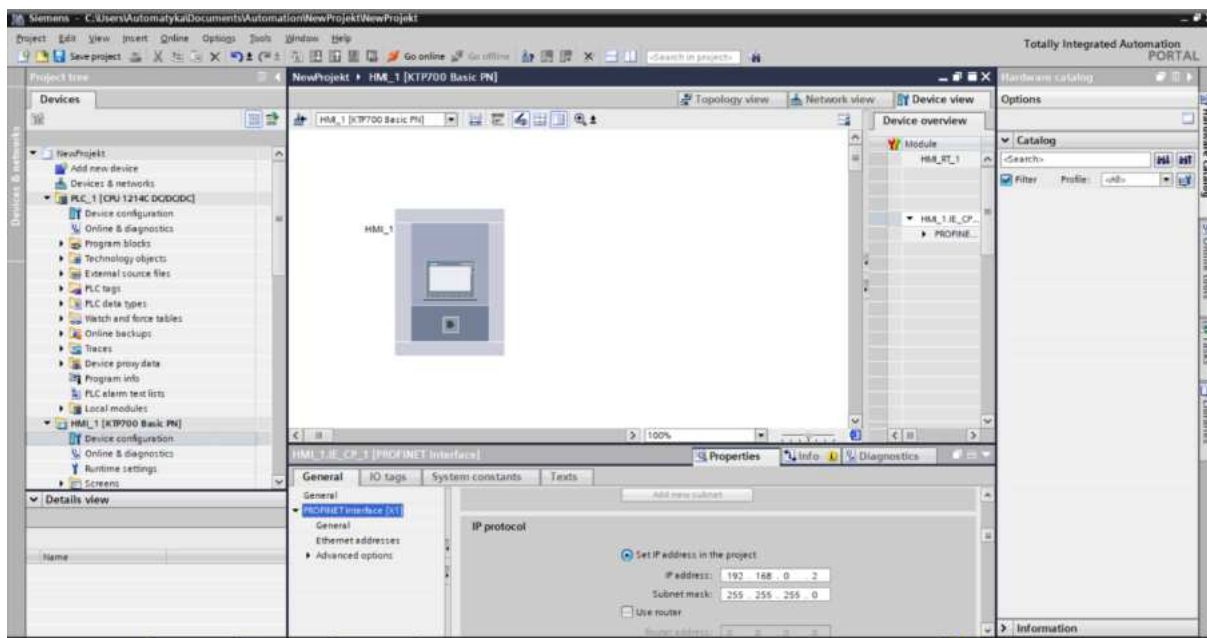
W oknie konfiguracyjnym łączymy go ze sterownikiem, Rys. 6.19.



Rys. 6.19 Połączenie panelu operatorskiego ze sterownikiem w TIA Portal

Pozostałe punkty można opuścić i zakończyć przyciskiem Finish.

Kolejnym krokiem jest nadanie adresu IP panelu operatorskiego. W tym celu klikamy w oknie **Project tree** pole **HMI – Device configuration**. Następnie w oknie dialogowym **Device view** zaznaczamy zielony prostokąt PROFINET. W znajdującym się poniżej oknie **Properties** wybieramy pole konfiguracji **Ethernet address**, Rys. 6.20.

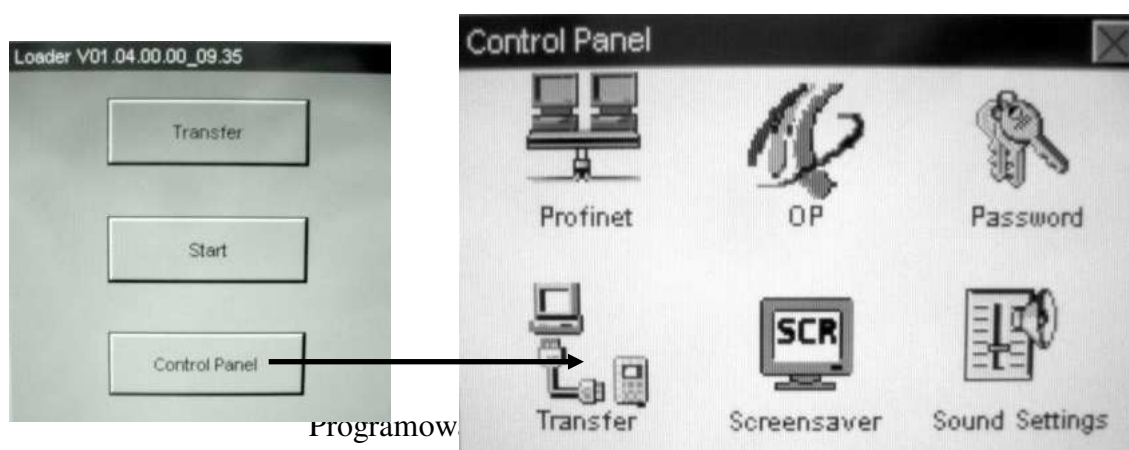


Rys. 6.20 Wprowadzenie adresu IP panelu operatorskiego w TIA Portal

TIA Portal wyświetla okno dialogowe konfiguracji adresu Ethernet, które pozwala powiązać program zawierający projekt z adresem IP wybranego panelu, który otrzyma ten projekt. **Adres IP nie jest ustalany w chwili wczytania programu** – wymaga wpisania bezpośrednio na panelu. Wszystkie urządzenia w sieci PROFINET muszą mieć swój indywidualny adres IP. Adres panelu musi być w tej samej sieci co sterownik i musi się różnić na ostatniej pozycji przykładowo o jeden.

6.3 Przygotowanie panelu operatorskiego do pracy

Po włączeniu zasilania panelu operatorskiego na jego ekranie pojawia się informacja dotycząca zainstalowanego oprogramowania, a po chwili ekran programu ładującego wyświetla trzy przyciski dotykowe **Transfer**, **Start** i **Control Panel**, Rys. 6.21. Dwa szybko po sobie następujące dotknięcia przycisku **Transfer** pozwalają na uruchomienie z komputera transmisji projektu przygotowanego w programie *WinCC flexible 2008 Compact*. (Normalnie programowanie panelu realizuje się z programu TIA Portal i wybranie opcji **Remote Control w Control Panel → Transfer → Transfer Settings**). Użycie przycisku **Start** powoduje przejście do ekranu startowego projektu. Zanim jednak będziemy się posługiwać tymi przyciskami powinniśmy skonfigurować panel wywołując przyciskiem **Control Panel** ekran konfiguracyjny. Konieczne jest, aby zapewnić komunikację panelu ze sterownikiem w sieci Profinet należy, używając okna **Profinet settings**, ustawić w zakładkach parametry takie jak np. adres IP, szybkość transmisji czy nazwę stacji. Można też odczytać adres MAC urządzenia, Rys. 6.22.



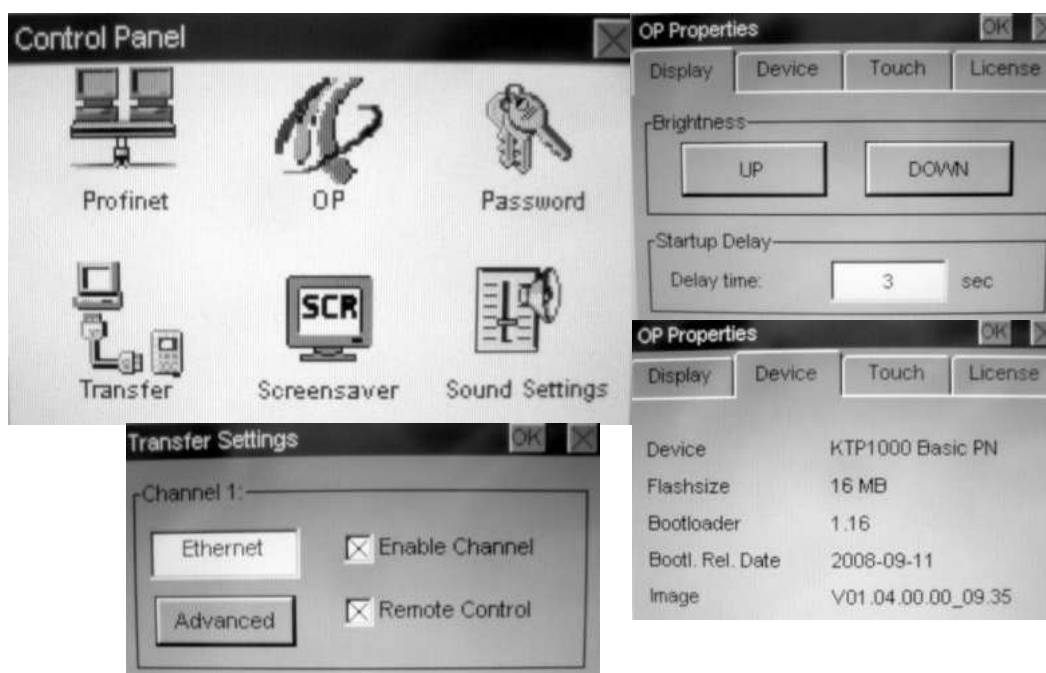
Programow

Rys. 6.21 Menu panelu operatorskiego



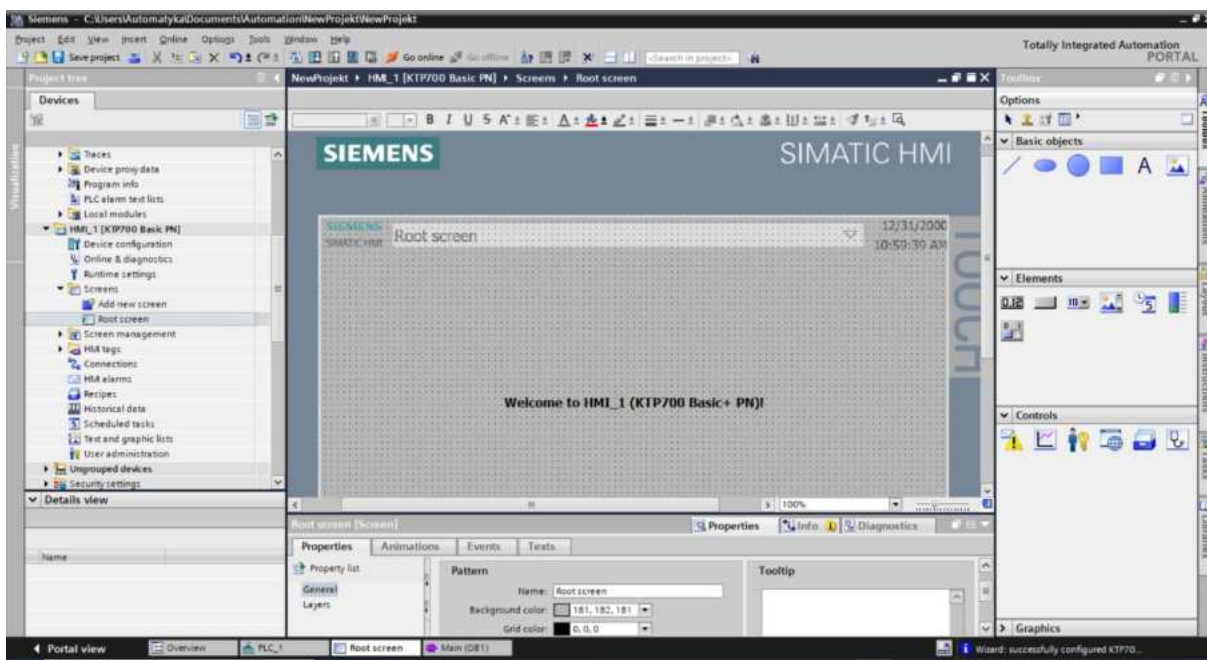
Rys. 6.22 Parametry panelu operatorskiego

Dwukrotne kliknięcie w ikonę **OP** na ekranie panelu powoduje pojawienie się ekranu **OP Properties**, którego kolejne zakładki dają możliwość ustawienia jasności ekranu, czasu opóźnienia pojawiania się ekranu startowego, wyświetlenia informacji o panelu i warunkach licencyjnych, Rys. 6.23 Kolejne parametry panelu operatorskiego. Zaznaczenie na ekranie **Transfer Settings** opcji **Remote Control** umożliwia uruchomienie transferu projektu z komputera, bez zezwolenia z panelu.



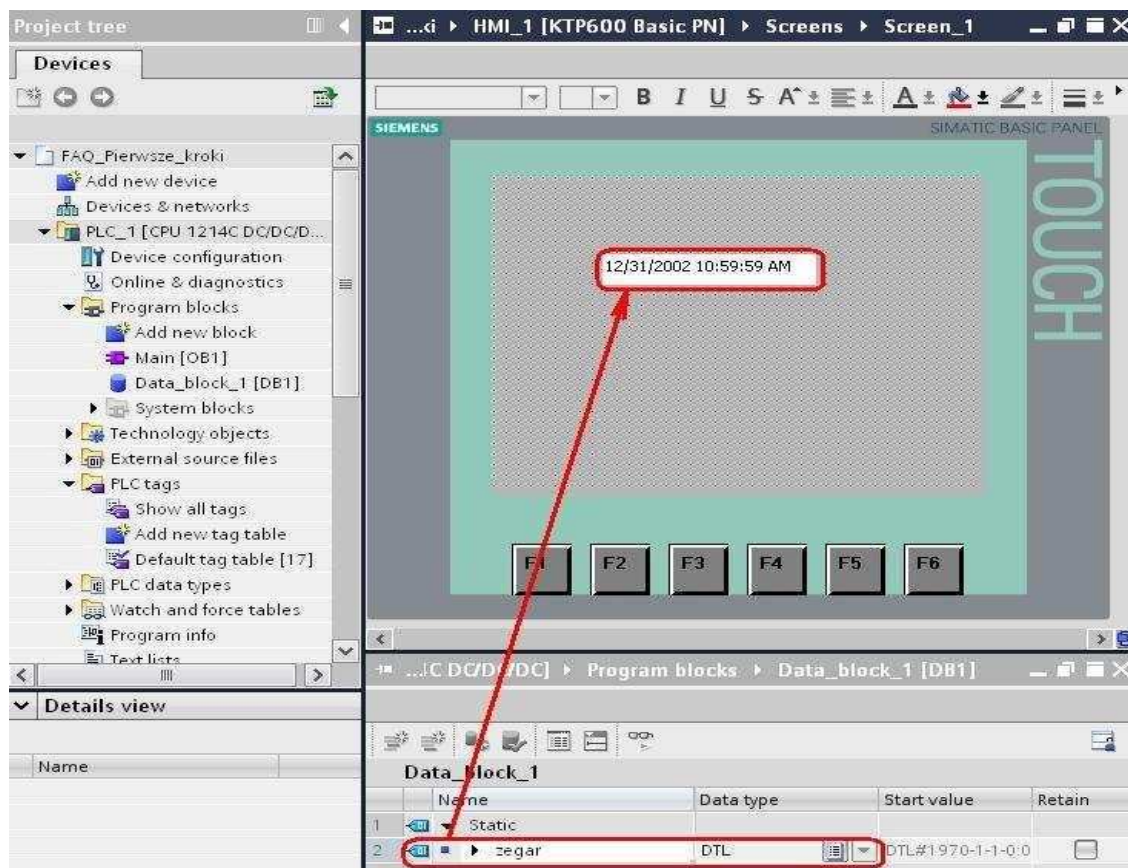
Rys. 6.23 Kolejne parametry panelu operatorskiego

Panel operatorski został skonfigurowany, należy przejść do programowania panelu według analizy układu sterowania, następnie wybieramy w oknie inspektora **Screens** i **Root Screen** (ekran główny) i realizujemy projekt, Rys. 6.24 Programowanie panelu operatorskiego w TIA Portal



Rys. 6.24 Programowanie panelu operatorskiego w TIA Portal

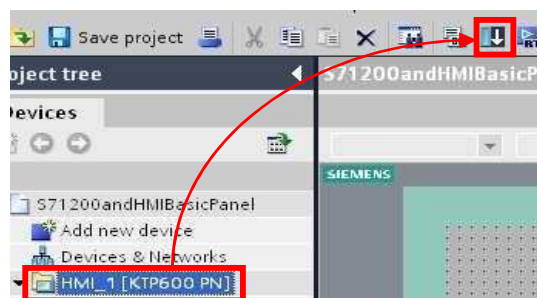
Przykładowo do celu testowania konfiguracji układu sterowania można wyświetlić na panelu HMI Basic bieżący czas lokalny sterownika PLC przechowywany w zmiennej „zegar”. Należy



Rys. 6.25 Projektowanie zegara na panelu operatorskim

bezpośrednio przeciągnąć zmienną z bloku danych sterownika na pole edytora **Root screen**. W ten sposób zostaje automatycznie utworzone pole I/O field, które wyświetla czas lokalny, Rys. 6.25.

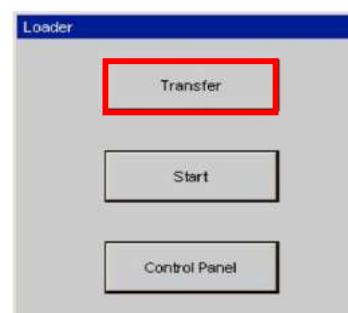
Wczytanie programu do panelu HMI Basic panel odbywa się poprzez zaznaczenie panelu HMI w oknie **Project tree** i kliknięcie ikony **Download to device**



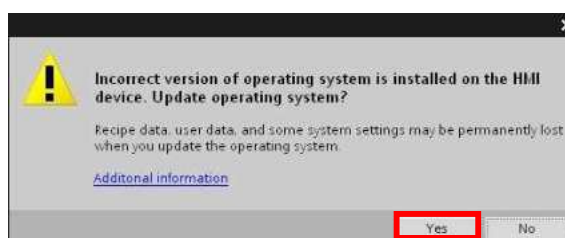
W nowo otwartym oknie zaznaczamy pole **Overwrite all** i wybieramy **Load**. Program zostaje automatycznie wgrany do panelu HMI.



W przypadku nie zaznaczonej opcji **Remote Control** w ustawieniach **Transfer Settings** (patrz punkt 3.2), konieczne jest wybranie pozycji **Transfer** z okna **Loader**, podczas wczytywania programu do panelu HMI.

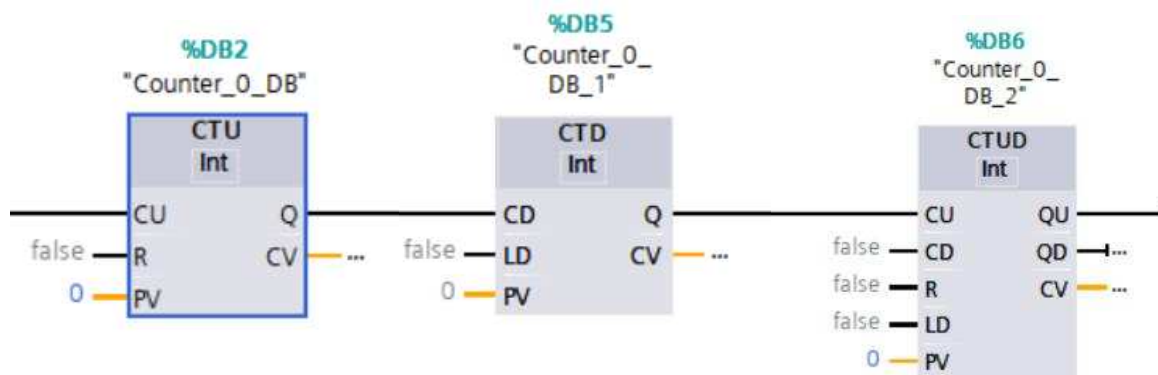


Jeżeli firmware HMI Basic Panels nie jest zgodny z oprogramowaniem WinCC Basic, system powiadomi nas o takim zdarzeniu podczas próby wczytania programu do HMI. Aby kontynuować proces należy zatwierdzić **YES**, firmware zostanie zaktualizowany.



6.4 Programowanie sterownika Siemens S7-1200 w TIA PORTAL v15.

1. Liczniki stosowane w TIA Portal, Rys. 6.26



Rys. 6.26 Dostępne liczniki w TIA Portal

CTU: licznik zliczający w górę

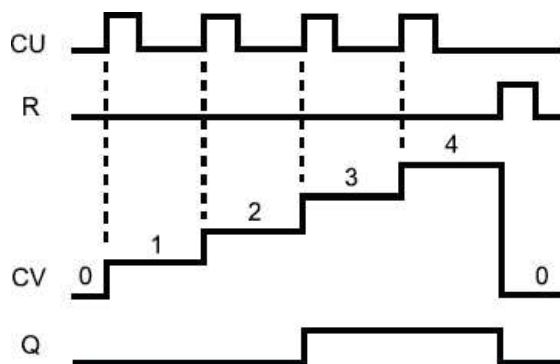
Jeżeli wartość parametru CU zmienia się z 0 na 1, to CTU zlicza w górę o 1. Jeżeli wartość parametru CV (*current count value* – bieżąca wartość zliczeń) jest większa lub równa wartości parametru PV (*preset count value* – ustalona wartość zliczeń), to parametr wyjściowy licznika Q = 1.

Jeżeli wartość parametru kasującego R zmienia się z 0 na 1, to bieżąca wartość zliczeń zostaje skasowana do 0.

Tabela 6.1 Parametry liczników

Parametr	Opis
CU, CD	Wejście zliczające liczników
R	Kasowanie stanu licznika do zera
LD	Ustawienie stanu licznika na ustaloną wartość PV
PV	Ustalona wartość zliczeń
CV	Bieżący stan licznika, wartość zliczeń
Q, QU	Wyjście potwierdzające stan licznika – wynosi 1, gdy CV ≥ PV
QD	Wyjście potwierdzające stan licznika – wynosi 1, gdy CV ≤ PV

Na poniższym rysunku przedstawiono przebieg czasowy w przypadku licznika CTU zliczającego liczby całkowite bez znaku (dla PV = 3), Rys. 6.27.



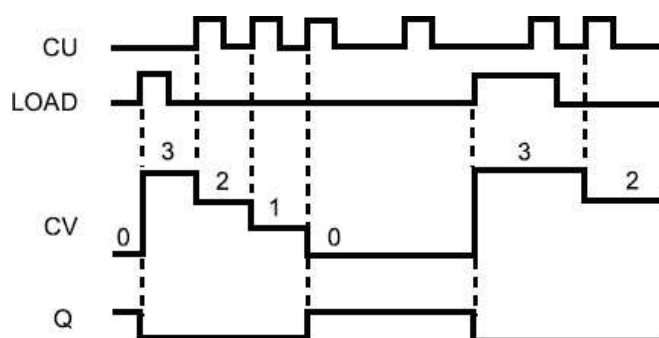
Rys. 6.27 Przebieg czasowy sygnałów wejściowych i wyjściowych na liczniku CTU

CTD: licznik zliczający w dół

Jeżeli wartość parametru CD zmienia się z 0 na 1, to CTU zlicza w dół o 1. Jeżeli wartość parametru CV (*current count value* – bieżąca wartość zliczeń) jest większa lub równa 0, to parametr wyjściowy licznika Q = 1.

Jeżeli wartość parametru LOAD zmienia się z 0 na 1, to wartość parametru PV (*preset count value* – ustalona wartość zliczeń) jest wpisywana do licznika jako nowa wartość CV (*current count value* – bieżąca wartość zliczeń).

Na poniższym rysunku przedstawiono przebieg czasowy w przypadku licznika CTD zliczającego liczby całkowite bez znaku (dla PV = 3), Rys. 6.28.



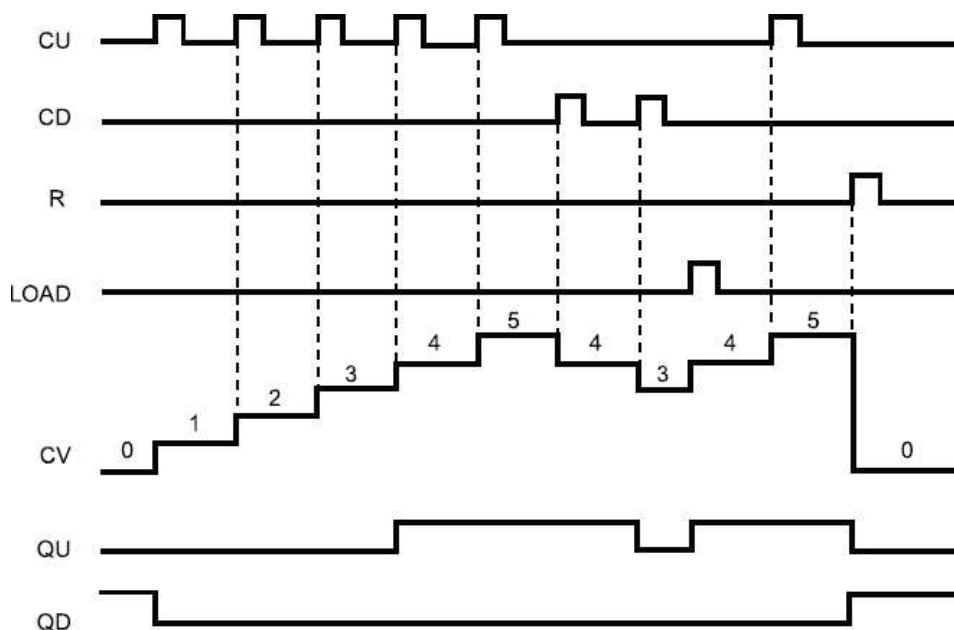
Rys. 6.28 Przebieg czasowy sygnałów wejściowych i wyjściowych na liczniku CTD

CTUD: licznik zliczający w górę i w dół

Licznik CTUD zlicza o 1 w górę lub w dół przy każdej zmianie z 0 na 1 na wejściach CU (*count up* – zliczanie w górę) lub CD (*count down* – zliczanie w dół). Jeżeli wartość parametru CV (*current count value* – bieżąca wartość zliczeń) jest równa lub większa od wartości parametru PV (*preset value* – ustalona wartość), to parametr wyjściowy licznika QU = 1. Jeżeli wartość parametru CV jest mniejsza lub równa 0, to parametr wyjściowy licznika QD = 1.

Jeżeli wartość parametru LOAD zmienia się z 0 na 1, to wartość parametru PV (*preset value* – ustalona wartość) jest wpisywana do licznika jako nowa wartość CV (*current count value* – bieżąca wartość zliczeń). Jeżeli wartość parametru kasującego R zmienia się z 0 na 1, to bieżąca wartość zliczeń zostaje skasowana do 0.

Na poniższym rysunku przedstawiono przebieg czasowy w przypadku licznika CTUD zliczającego liczby całkowite bez znaku (dla PV = 4), Rys. 6.29.



Rys. 6.29 Przebieg czasowy sygnałów wejściowych i wyjściowych na liczniku CTUD

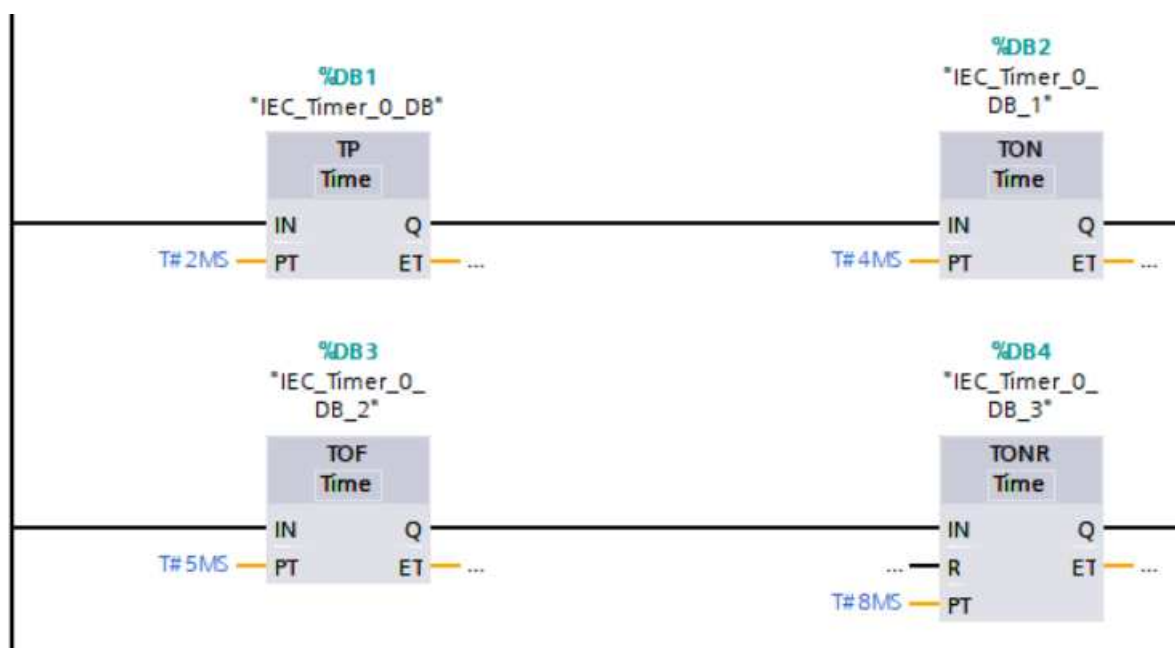
2. Timery stosowane w TIA Portal, Rys. 6.30.

TP (Pulse Timer) generuje impuls o ustalonym czasie trwania (ustawia wyjście Q w stan „1” na ustalony czas). Odmierzanie czasu następuje po zmianie stanu sygnału na wejściu IN z „0” na „1”. Jakakolwiek zmiana stanu sygnału na wejściu IN w trakcie odmierzania czasu nie ma żadnego wpływu na pracę timera, Rys. 6.31.

TON (ON-delay Timer) ustawia wyjście Q w stan „1” (ON - włączony) po upływie zadanego czasu opóźnienia. Odmierzanie czasu następuje po zmianie stanu sygnału na wejściu IN z „0” na „1”. Zmiana stanu sygnału na wejściu IN z „1” na „0” w trakcie odmierzania czasu anuluje odmierzanie czasu i powoduje natychmiastowy powrót wyjścia Q do stanu „0”, Rys. 6.32.

TOF (OFF-delay Timer) kasuje wyjście Q do stanu „0” (OFF - wyłączony) po upływie zadanego czasu opóźnienia. Odmierzanie czasu następuje po zmianie stanu sygnału na wejściu IN z „1” na „0”. Zmiana stanu sygnału na wejściu IN z „1” na „0” w trakcie odmierzania czasu anuluje odmierzanie czasu, a wyjście Q pozostaje w stanie „1”, Rys. 6.33.

TONR (ON-delay Retentive Timer) ustawia wyjście Q w stan „1” (ON - włączony) po upływie zadanego czasu opóźnienia. Odmierzanie czasu następuje po zmianie stanu sygnału na wejściu IN z „0” na „1”. Zmiana stanu sygnału na wejściu IN z „1” na „0” w trakcie odmierzania czasu wstrzymuje odmierzanie czasu, ale nie anuluje rozpoczętego procesu. Kolejna zmiana stanu sygnału na wejściu IN z „0” na „1” powoduje, że odmierzanie czasu jest kontynuowane. Powrót wyjścia Q do stanu „0” następuje dopiero w chwili podania sygnału „1” na wejście R, Rys. 6.34.



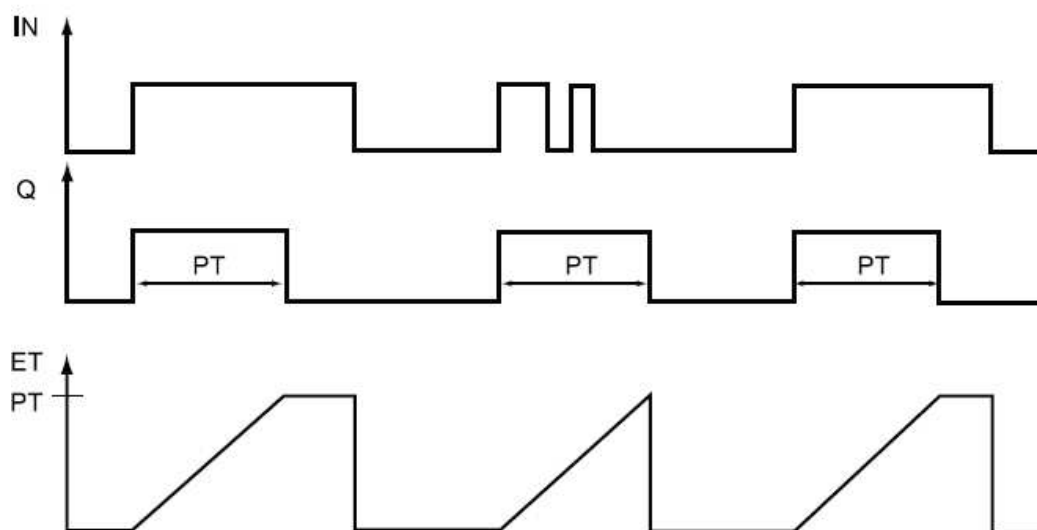
Rys. 6.30 Timery dostępne TIA Portal STEP 7

Tabela 6.2 Parametry timerów

Parametr	Opis
IN	Wejście zezwalające timera
R	Wejście resetujące stan wyjściowy timera TONR
PT	Odmierzana wartość czasu
Q	Wyjście timera
ET	Bieżąca wartość upływającego czasu

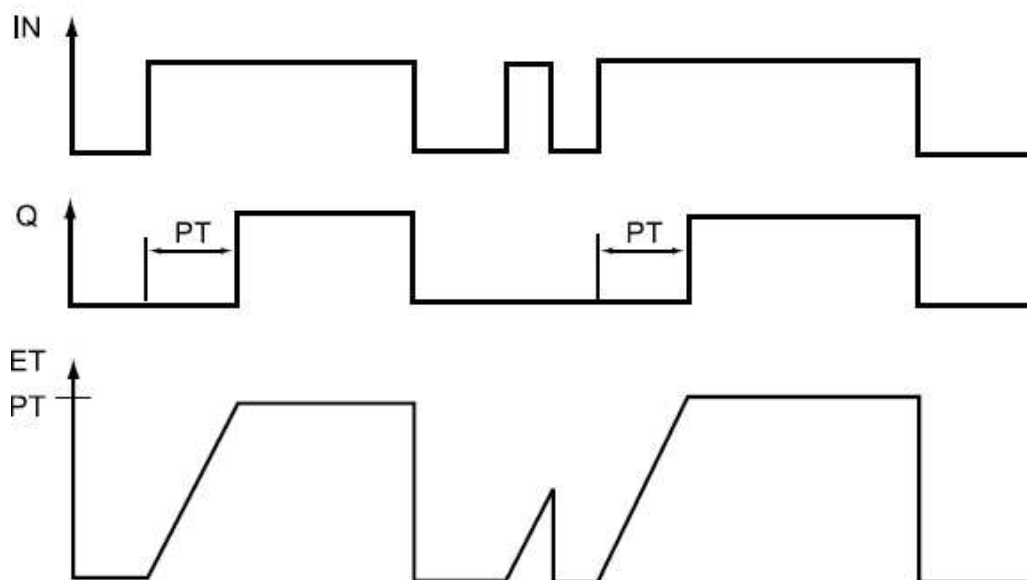
Przebiegi czasowe ukazujące zasadę działania poszczególnych timerów:

TP: przebieg czasowy Pulse



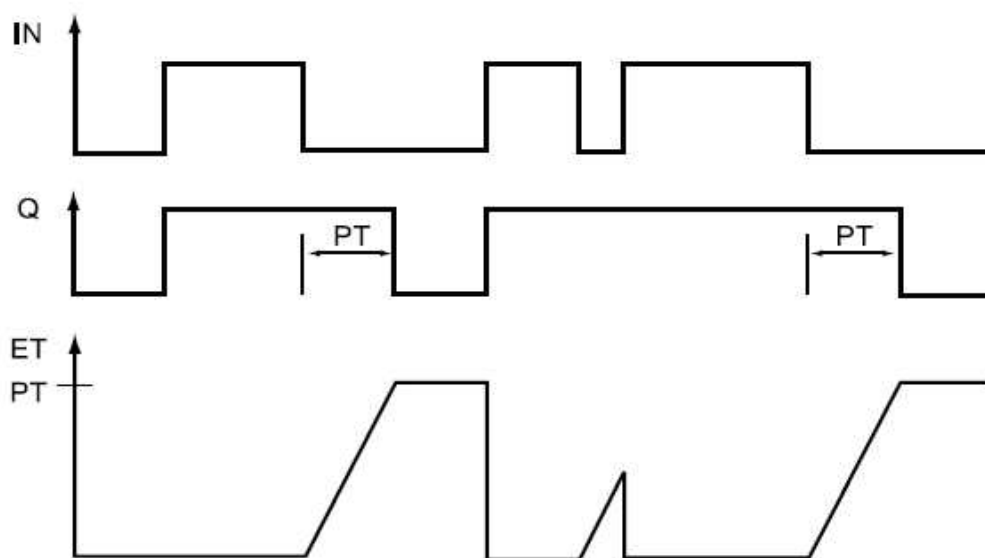
Rys. 6.31 Przebieg czasowy timera TP

TON: Przebieg czasowy ON-delay



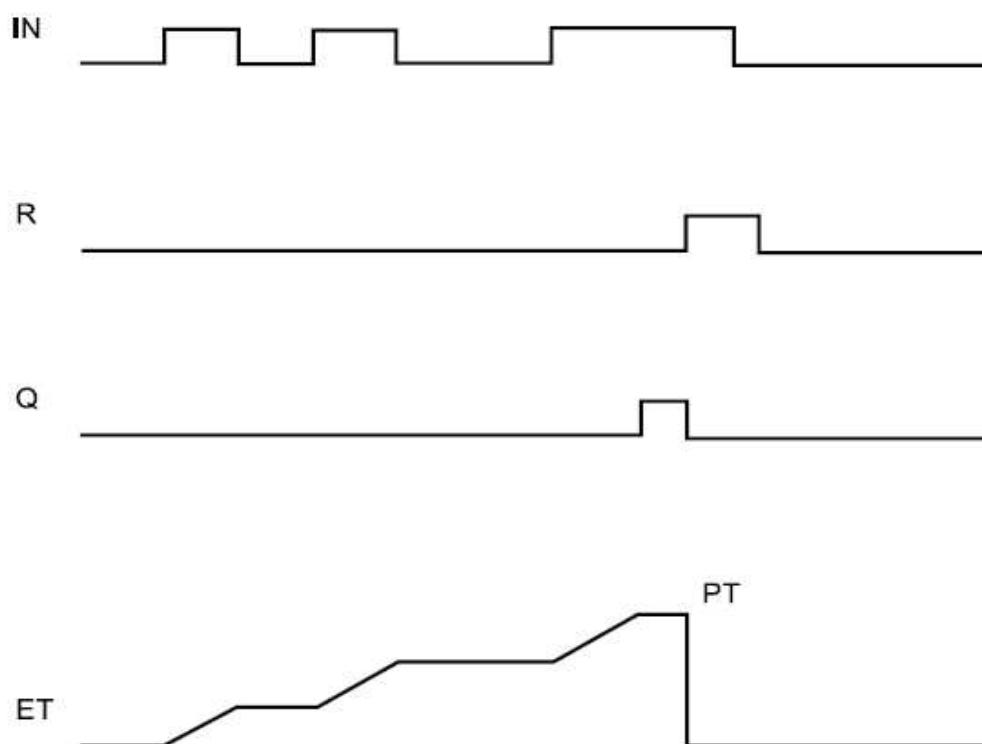
Rys. 6.33 Przebieg czasowy timera TON

TOF: Przebieg czasowy OFF-delay



Rys. 6.32 Przebieg czasowy timera TOF

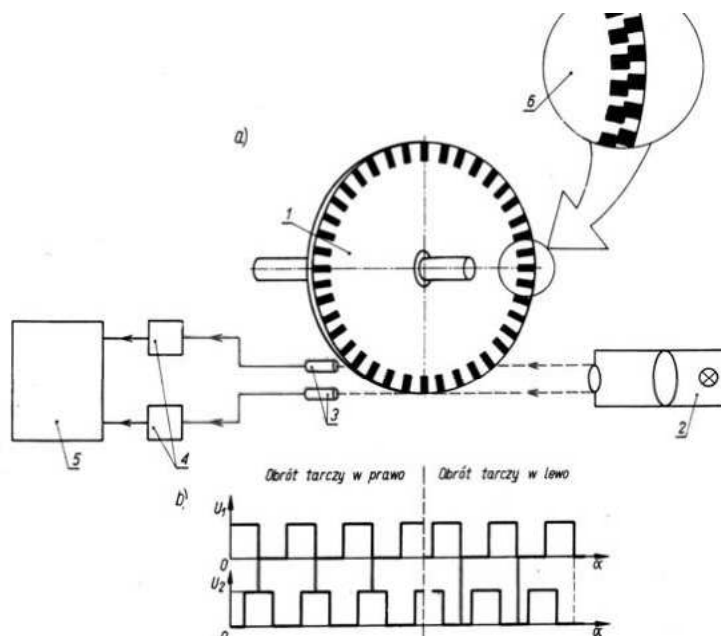
TONR: Przebieg czasowy ON-delay Retentive



Rys. 6.34 Przebieg czasowy timera TONR

6.5 Porty HSC na sterownikach

Każdy licznik HSC może pracować jako 'single phase' jedno kanałowy lub wielokanałowy, Rys. 6.35. W szerszym znaczeniu stosujemy enkodery.



Rys. 6.35 Pomiar prędkości obrotowej wału z kierunkiem obrotu

Czujnik impulsowy generuje impulsy do pomiaru prędkości wału. Enkodery to czujniki kontroli ruchu. Umożliwiają określanie precyzyjnej pozycji elementów maszyny, kąta obrotu, prędkości, przyspieszenia, kierunku ruchu, jak również pomiar odległości. Ale to nie wszystko. Co raz częściej enkoder to nie tylko przetwornik obrotowy, ale też element układu automatyki, który przesyła szereg informacji serwisowych i diagnostycznych określających żywotność naszej maszyny. Wykorzystanie funkcjonalności enkodera pozwala na określenie wielu parametrów ruchu.

Bez względu na to, na jakie typy, czy rodzaje podzielimy czujniki kontroli ruchu i do czego będziemy je wykorzystywać, to zawsze enkodery będą nam służyć przede wszystkim do konwersji ruchu mechanicznego na sygnał elektryczny. Niezależnie czy to będzie ruch liniowy czy obrotowy. Słowo ruch już samo w sobie niesie dynamikę i przemieszczenie. Przesunięcie wiąże się z pomiarem drogi, kierunku ruchu czy kąta obrotu. Dzięki temu, że enkodery przetwarzają przesunięcie i pozycję kątową na sygnał elektryczny, możliwe jest uzyskanie informacji o liczbie obrotów wykonanych przez dany element, a także o przebytej drodze w ruchu postępowym. Tylko kontrolując ruch można go zoptymalizować.

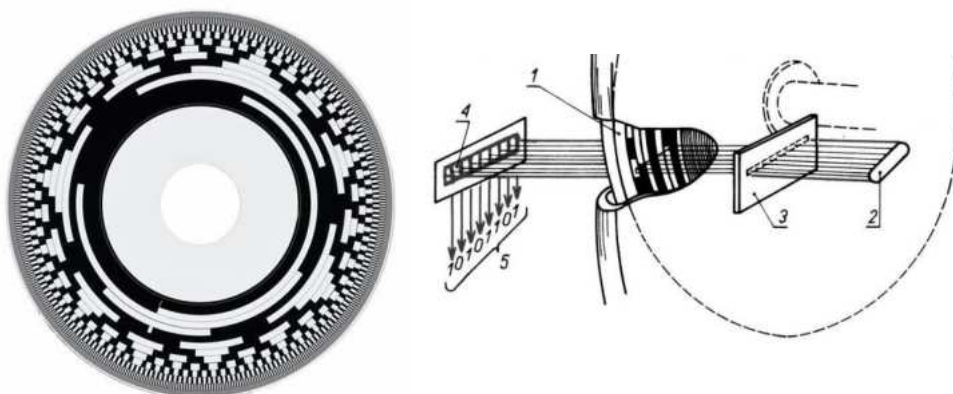
Od kiedy James Watt wynalazł maszynę parową, a ludzkość wkroczyła na ścieżkę pierwszej rewolucji przemysłowej, kontrola i optymalizacja ruchu stała się jednym z priorytetów ówczesnych inżynierów. Jednak dopiero rozwój elektroniki i systemów teleinformatycznych pozwolił na ewolucję czujników kontroli ruchu.

Enkoder inkrementalny to rozwiązanie w prosty sposób pozwoliło kontrolować ruch obrotowy pierwszych układów napędowych. Sam enkoder inkrementalny to generator impulsów elektrycznych prostokątnych lub sinusoidalnych, które są ściśle powiązane z ruchem obrotowym napędu, do którego zamontowany jest enkoder.

Budowa i zasada działania enkodera inkrementalnego

Enkoder inkrementalny składa się z kilku podstawowych elementów. Od strony mechanicznej to **wałek z zamontowaną tarczą kodową**, **dwa łożyska** oddalone od siebie na maksymalną odległość, **układ skanujący** i **układ elektroniczny**, który formuje sygnał wyjściowy.

Bardzo istotnym elementem w enkoderach optycznych jest tarcza kodowa, Rys. 6.36. Metalowa tarcza kodowa jest idealnym rozwiązaniem w aplikacjach, gdzie występują wstrząsy czy wibracje. Poza tym metalowa tarcza kodowa gwarantuje zdecydowanie większą powtarzalność pomiaru. Inkreментy naniesione na tarczę kodową to nic innego jak **wycięte laserowo przestrzenie w metalowej tarczy**, przez które przechodzi światło nadajnika. Ilość naniesionych inkrementów definiuje nam parametr **rozdzielczości** enkodera.



Rys. 6.36 Tarcza kodowa 12-bitowa i pomiar położenia wału

Enkodery inkrementalne nie są w stanie zapamiętać swojej pozycji po utracie zasilania. Oznacza to, że po ponownym uruchomieniu systemu enkoder musi zostać skalibrowany od nowa. Ponadto, enkodery inkrementalne nie są w stanie zapewnić dokładności pomiarów na takim samym poziomie jak enkodery absolutne. Są one stosowane do kontroli prędkości silników elektrycznych, monitorowania ruchu maszyn oraz do sterowania robotami przemysłowymi. W przypadku zastosowań wymagających wysokiej precyzji pomiarów, lepszym wyborem będą enkodery absolutne.

Enkodery absolutne

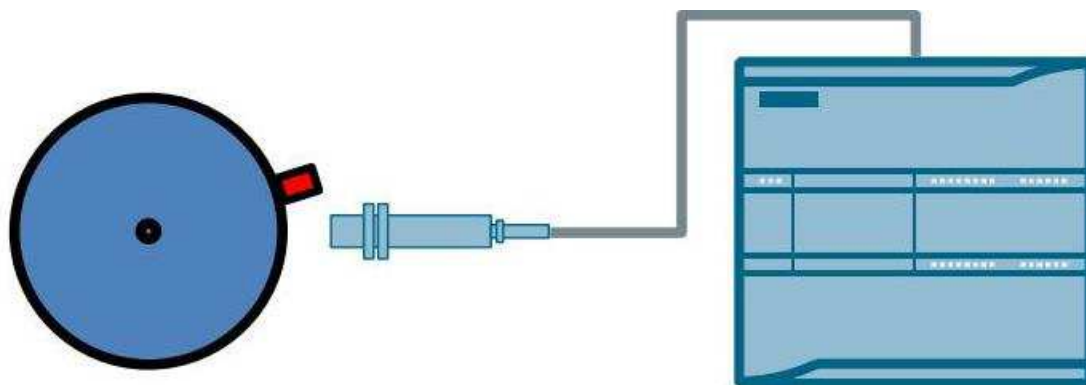
Enkodery absolutne to urządzenia, które pozwalają na dokładne określenie położenia wałka. Dzięki temu są one idealnym rozwiązaniem w przypadku zastosowań wymagających precyzyjnego pomiaru położenia maszyn, takich jak np. obrabiarki CNC czy roboty przemysłowe. W odróżnieniu od enkoderów inkrementalnych, enkodery absolutne nie tracą informacji o położeniu w momencie wyłączenia zasilania.

Jedną z największych zalet enkoderów absolutnych jest ich wysoka dokładność pomiaru. Dzięki temu mogą być one stosowane w aplikacjach wymagających bardzo precyzyjnego ustawienia. Ponadto, enkodery absolutne charakteryzują się dużą odpornością na zakłócenia elektromagnetyczne oraz drgania mechaniczne. Ze względu na swoją budowę i sposób działania, enkodery absolutne są bardziej skomplikowane i wymagają większej ilości czasu i pracy przy instalacji oraz konfiguracji. Jednakże, jeśli potrzebujemy bardzo precyzyjnego pomiaru położenia obrotowego lub pracujemy w warunkach wymagających dużej odporności na zakłócenia, enkodery absolutne są najlepszym wyborem.

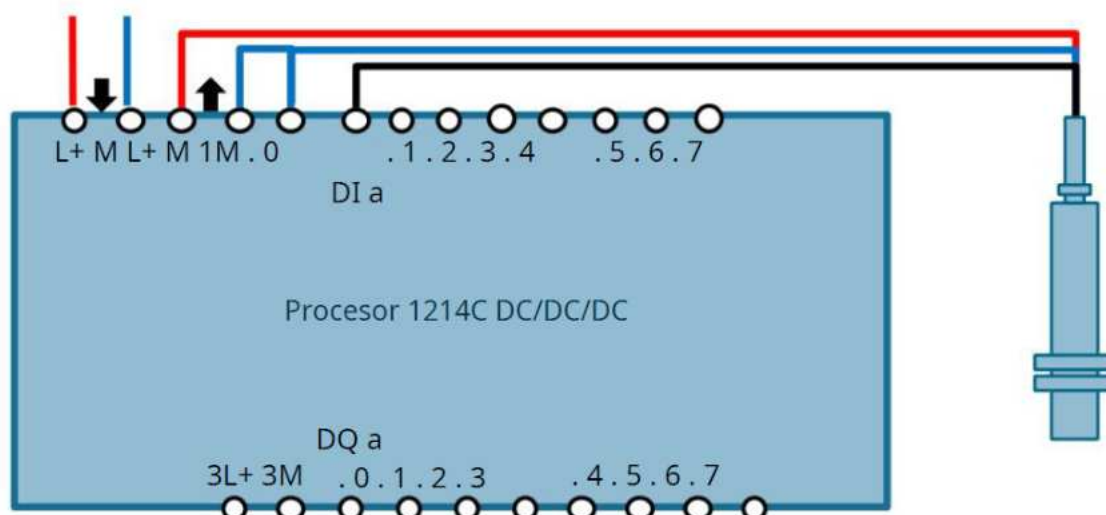
6.6 Programowanie portu HSC na sterowniku S7-1200 - tworzenie projektu w programie TIA PORTAL v.15

Pomiar prędkości wału

W technicznym zastosowaniu istnieje wiele szybkich zdarzeń, których nie można wykryć w cyklu programu głównego na sterowniku. Szybkie liczniki HSC w S7-1200 pomagają przetwarzać te zdarzenia. Pomiar prędkości wału w przypadku tylko jednego impulsu lub kilku impulsów na obrót przedstawia Rys. 6.37.

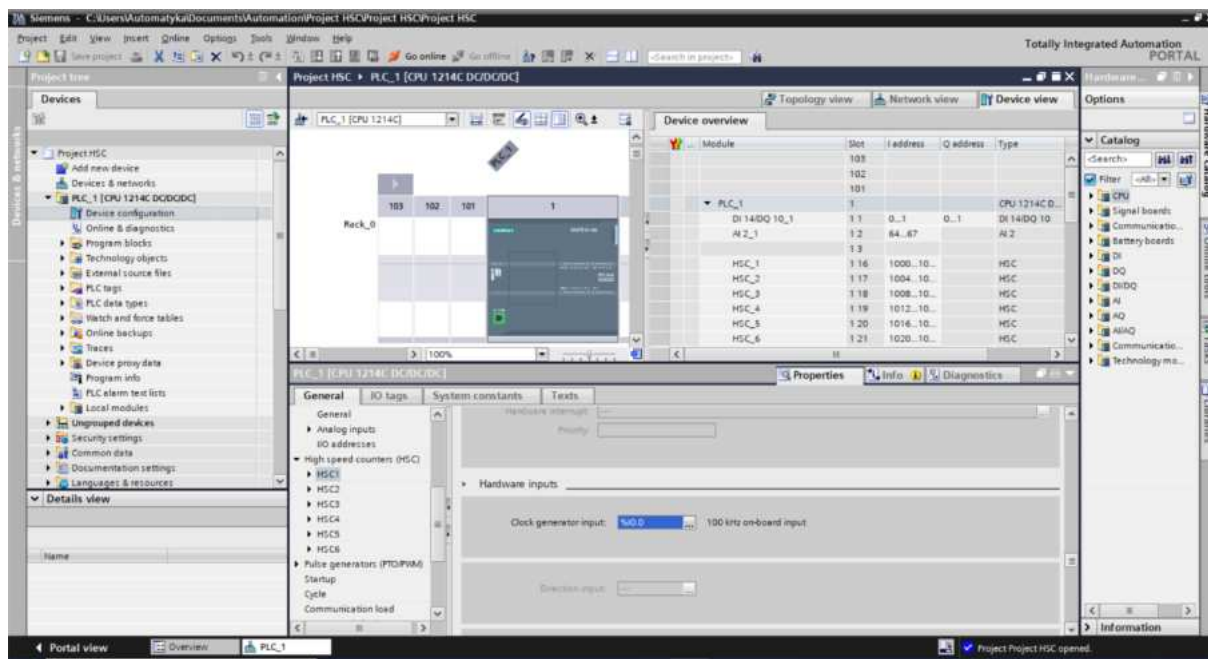


Rys. 6.37 Pomiar prędkości wału w przypadku jednego impulsu na obrót



Rys. 6.38 Konfiguracja sprzętu do pomiaru prędkości – podłączenie czujnika impulsów

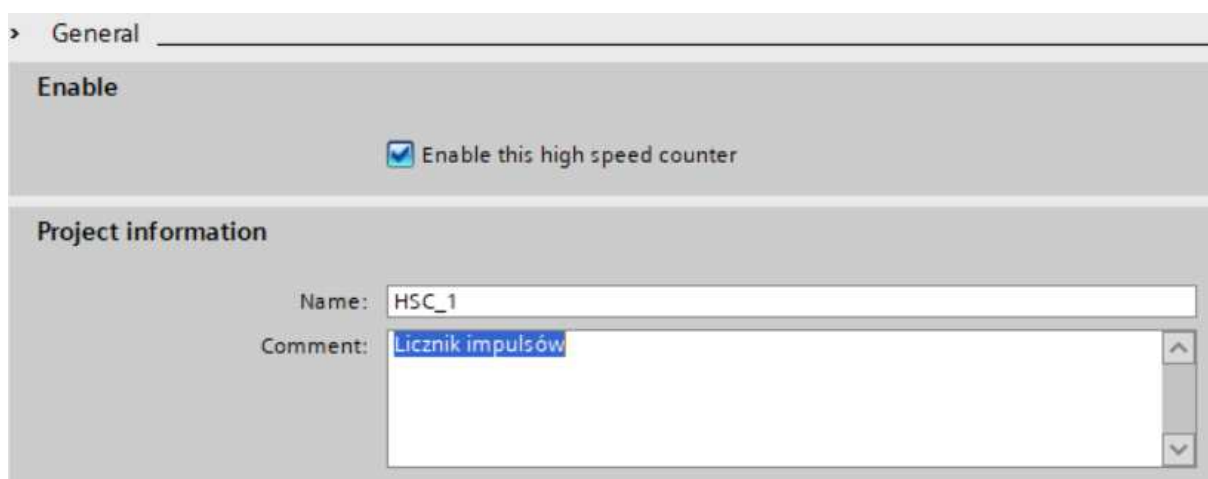
Podłączenie sprzętowe pokazane jest na Rys. 6.38. Liczniki HSC w sterowniku S7-1200 są zdefiniowane na wejściach cyfrowych %I0.x i dostępnych jest sześć liczników, Rys. 6.39.



Rys. 6.39 Wybór licznika HSC i jego portu

Konfiguracja szybkiego licznika w TIA Portal:

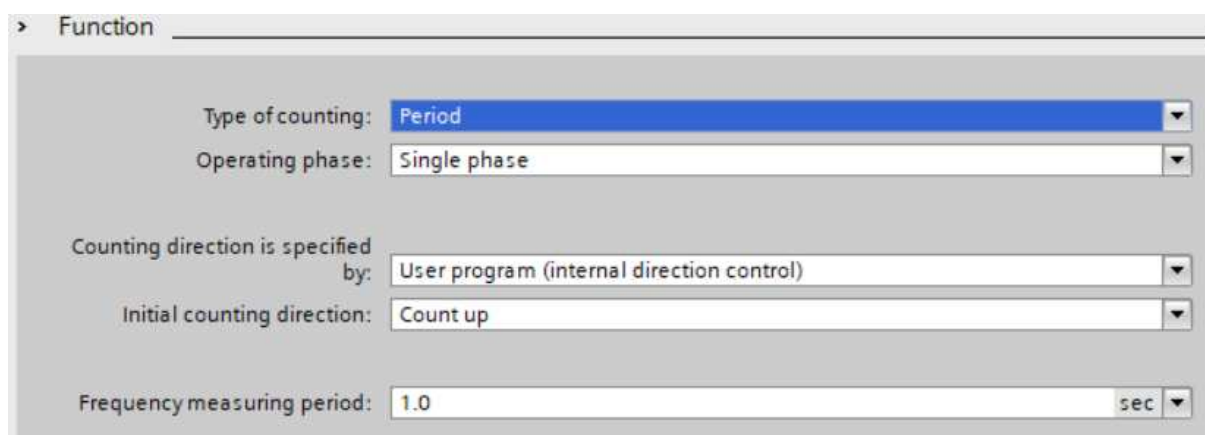
1. W widoku urządzenia lub sieci wybierz sterownik S7-1200.
2. W oknie inspektora przejdź do „Właściwości > Ogólne > Szybkie liczniki (HSC)” i kliknij szybki licznik „HSC1”, Rys. 6.40.
3. Włącz szybki licznik w grupie parametrów „General”, zaznaczając właściwe pole wyboru. W sekcji „Informacje o projekcie” możesz wpisać nazwę i komentarz do licznika.



Rys. 6.40 Włączanie licznika impulsów HSC

4. W grupie parametrów „Function”, Rys. 6.41, zdefiniuj działanie licznika jako następujące:
 - Rodzaj liczenia 'Count, Period, Frequently'
 - Rodzaj pracy 'Single phase'
 - Kierunek zliczania jako kierunek ustalany wewnętrznie
 - Początkowy kierunek liczenia 'Count up'

- Okres pomiaru częstotliwości 1,0sek



The screenshot shows the 'Function' configuration window for the HSC counter. It includes the following settings:

- Type of counting: Period
- Operating phase: Single phase
- Counting direction is specified by: User program (internal direction control)
- Initial counting direction: Count up
- Frequency measuring period: 1.0 sec

Rys. 6.41 Funkcje licznika impulsów HSC

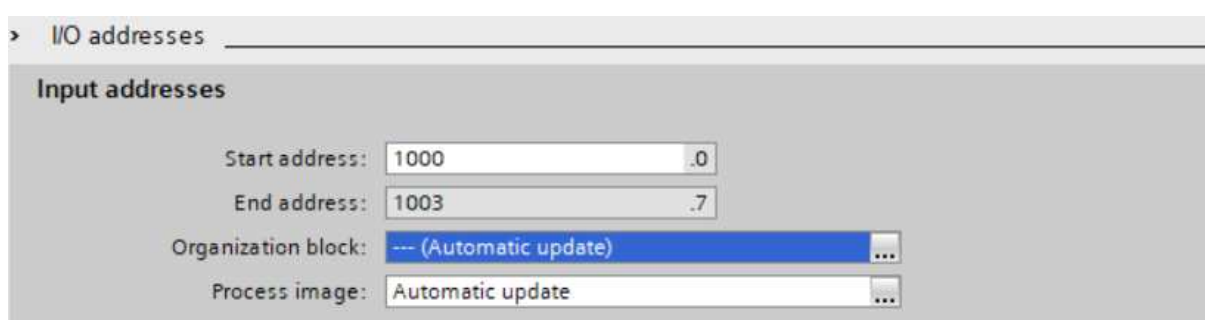
5. W grupie parametrów „Hardware inputs”, Rys. 6.42, przejdź do „Wejście generatora zegara” i wprowadź wejście sprzętowe „%I0.0”.



The screenshot shows the 'Hardware inputs' configuration window. The 'Clock generator input' is set to '%I0.0' and '100 kHz on-board input'.

Rys. 6.42 Wejście sprzętowe licznika impulsów HSC

6. W obszarze „Input addresses”, Rys. 6.43, ustaw parametry adresów wejściowych.

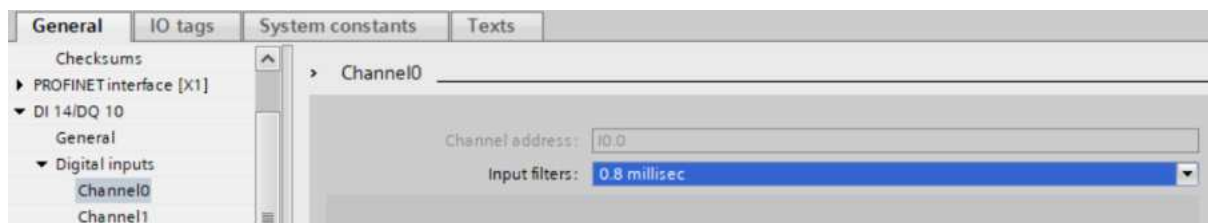


The screenshot shows the 'I/O addresses' configuration window. The 'Input addresses' section is expanded, showing the following settings:

- Start address: 1000
- End address: 1003
- Organization block: --- (Automatic update)
- Process image: Automatic update

Rys. 6.43 Konfiguracja standardowa adresu wejścia cyfrowego

7. Aby zapewnić bezpieczne wykrywanie impulsów generatora zegara, Rys. 6.44, czas filtrowania wejścia cyfrowego musi być ustawiony na mniej niż czas trwania sygnału wejściowego. Ustawiamy czas filtrowania w następujący sposób:
 - W widoku urządzenia lub sieci wybierz procesor S7-1200.
 - W oknie inspektora przejdź do 'Właściwości > Ogólne>DI 14/DQ 10 > Cyfrowe wejścia' i kliknij 'Channel 0'.
 - Ustaw 'Input filters', np. '0,8 miliseć'.



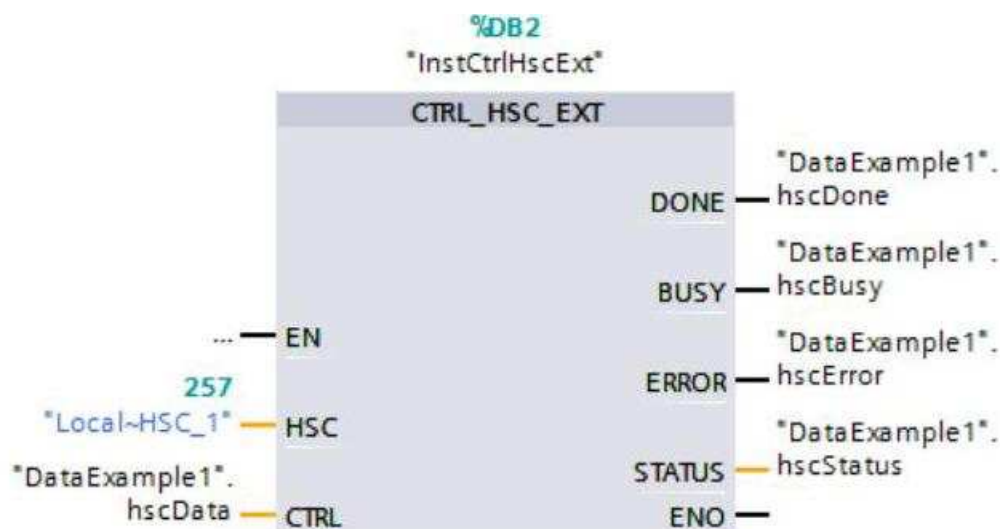
Rys. 6.44 Filtry wejściowe dla generatora zegara

Integracja licznika z projektem użytkownika w programie

Praca licznika HSC jest niezależna od programu, licznik HSC pracuje w tle programu i wymaga ustawienia sprzętowego pokazanego wcześniej. Dostęp do wskazań licznika HSC w programie zapewnia instrukcja 'CTRL_HSC_EXT' i 'CTRL_HSC'

Używając instrukcji „CTRL_HSC_EXT” („Kontroluj szybkie liczniki (rozszerzony)”), możemy konfigurować i sterować szybkimi licznikami obsługiwanymi przez CPU za pośrednictwem oprogramowania. Instrukcja „Sterowanie szybkim licznikiem (rozszerzony)” jest wykonywana tylko wtedy, gdy stan sygnału na wejściu EN wynosi „1”. Dopóki operacja jest wykonywana, bit na wyjściu BUSY jest ustawiony. Po całkowitym wykonaniu operacji bit na wyjściu BUSY jest resetowany. Wyjście ENO enable jest ustawiane tylko wtedy, gdy wejście EN enable ma stan „1” i podczas wykonywania operacji nie występują błędy. Po wstawieniu instrukcji „Sterowanie szybkim licznikiem (rozszerzonym)” tworzony jest blok danych instancji, w którym zapisywane są dane operacyjne.

Instrukcja „CTRL_HSC_EXT” obsługuje pomiar czasu trwania okresu. Oferuje dostęp programowy do liczby impulsów wejściowych w określonym przedziale pomiarowym. Instrukcja jest wywoływana w programie cyklicznym w następujący sposób, Rys. 6.45.



Rys. 6.45 Wywołanie instrukcji „CTRL_HSC_EXT”

Na wejściu „HSC” podaje się identyfikator sprzętowy (HW-ID) szybkiego licznika „HSC1”. Parametr „CTRL” wymaga wskaźnika typu danych systemowych „HSC_Period”.

Zmienna typu danych systemowych „HSC_Period” jest zdefiniowana w bloku danych „DataExample1”, Rys. 6.46.

DataExample1				
		Name	Data type	Start value
1		Static		
2		hscData	HSC_Period	
3		ElapsedTime	UDInt	0
4		EdgeCount	UDInt	0
5		EnHSC	Bool	TRUE
6		EnPeriod	Bool	false
7		NewPeriod	Int	1000

Rys. 6.46 Struktura typu danych systemowych „HSC_Period”

„ElapsedTime” określa czas w nanosekundach pomiędzy ostatnimi zdarzeniami zliczania kolejnych interwałów pomiarowych.

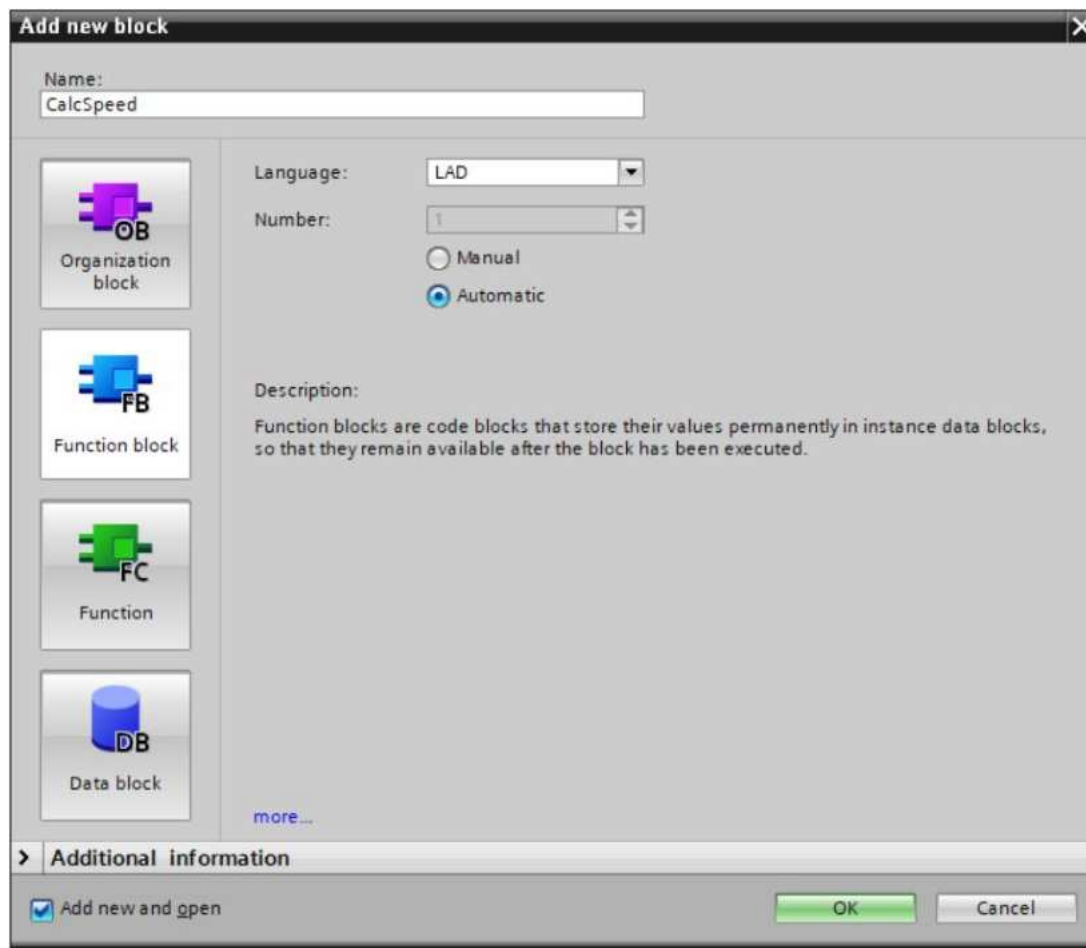
„EdgeCount” wyświetla liczbę zdarzeń zliczania odebranych w okresie pomiarowym.

Przy wartości początkowej „TRUE” dla „EnHSC” pomiar jest na stałe włączony.

Dla parametru „NewPeriod” określa się interwał pomiaru okresu w milisekundach. Możesz wybrać pomiędzy 10, 100 i 1000. W przykładzie aplikacji 1000 jest określone jako interwał pomiaru 1.0sek. Okres można zaktualizować, ustawiając „TRUE” dla „EnPeriod”.

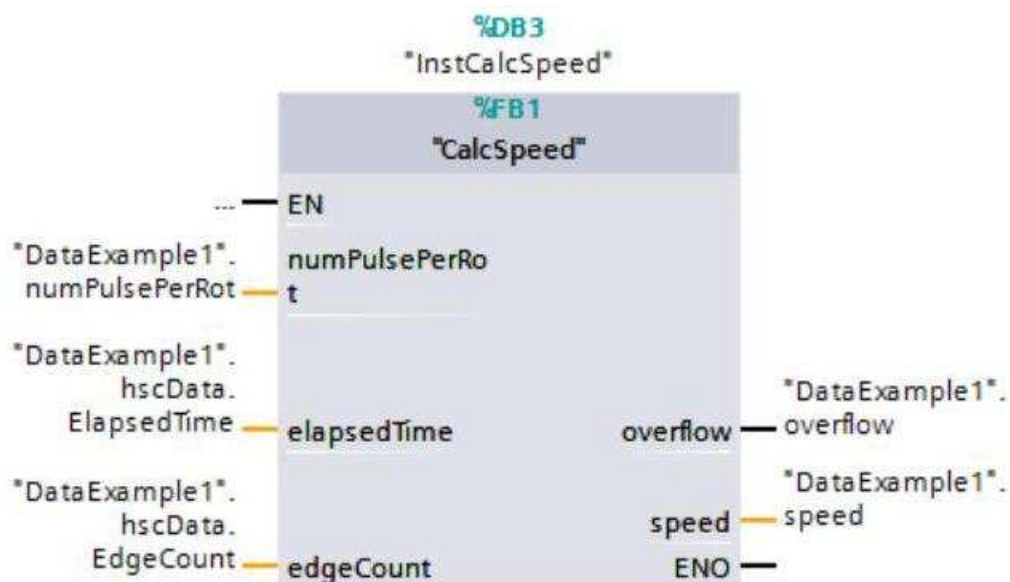
Obliczanie prędkości

Do wyznaczenia prędkości wskazane jest dodać nowy ‘Function Blok’ o nazwie ‘CalcSpeed’, Rys. 6.47.



Rys. 6.47 Dodawanie bloku funkcyjnego

Blok funkcyjny (FB) „CalcSpeed” służy do obliczania prędkości na podstawie wartości określonych w liczniku HSC „ElapsedTime” i „EdgeCount”, a następnie wylicza prędkość dla parametru „Speed”, Rys. 6.48.



Rys. 6.48 Wywołanie „CalcSpeed”

Tabela 6.3 Parametry FB i funkcji „CalcSpeed”

Name	P type	Data type	Comment
NumPulse PerRot	IN	Int	Liczba impulsów na obrót
ElapsedTime	IN	UDInt	Czas w ns między narastającymi krawędziami od „edgeCount”
EdgeCount	IN	UDInt	Liczba narastających zboczy w czasie, jaki upłynął od „elapsedTime”
Overflow	OUT	Bool	Przepełnienie okresu
Speed	OUT	Real	Obliczona prędkość w 1/min

Funkcja ‘CalcSpeed’ oblicza okres w [s] za pomocą następującego wzoru:

$$Period = \frac{ElapsedTime}{EdgeCount \cdot 10^9} \cdot numPulsePerRot$$

Okres zostanie obliczony tylko wtedy, gdy wartość „edgeCount” jest większa od zera, a wartość „elapsedTime” mieści się w zakresie od 0 do 4.294.967.280. Jeśli „elapsedTime” osiągnął wartość 4.294.967.295 (0xFFFF FFFF), następuje przepełnienie okresu.

Przekroczenie jest wyświetlane na wyjściu „przepełnienie” a okres ustawiany na zero.

Parametr „prędkość” wyliczany jest ze wzoru:

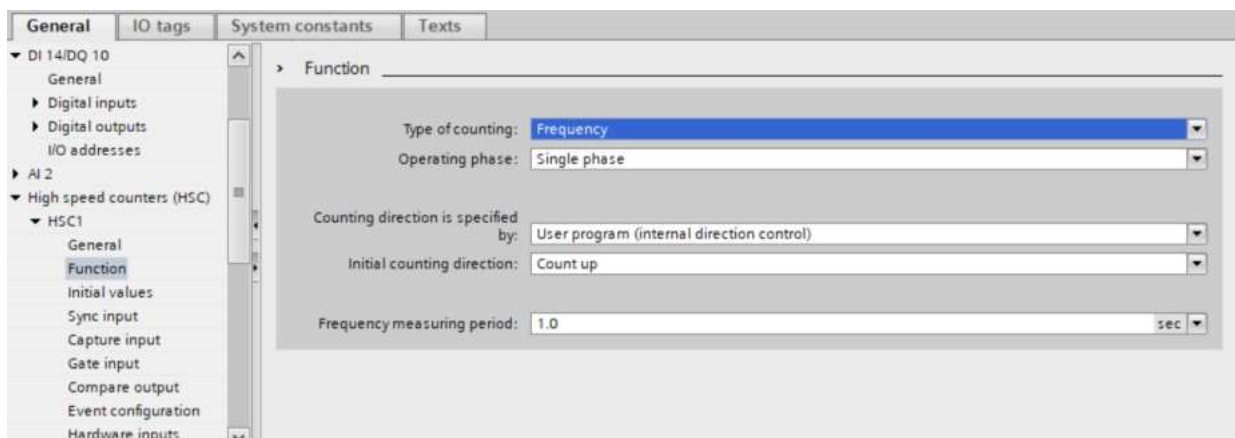
$$Speed = \frac{1}{Period} \cdot 60$$

„Speed” zostanie obliczona tylko wtedy, gdy okres jest większy od zera.

Funkcja ‘Calcspeed’ z opcją ‘Frequency’

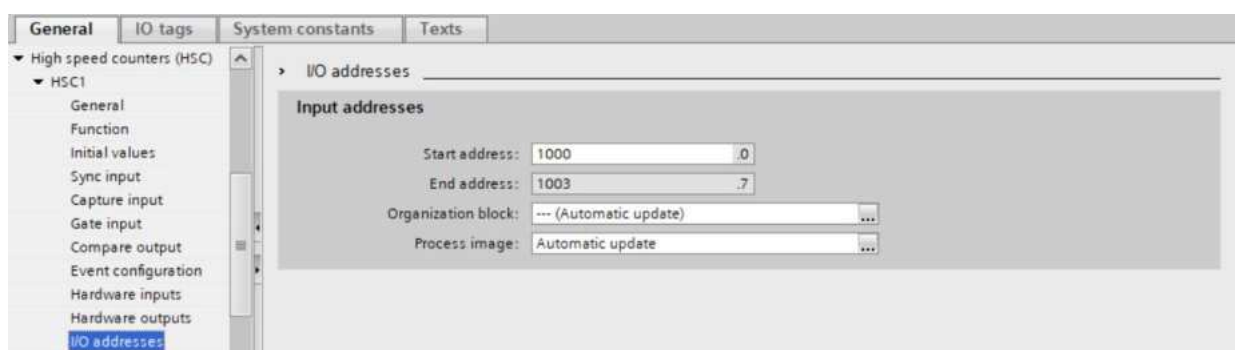
Konfiguracja sprzętu i układ pomiarowy pozostaje bez zmian. W grupie parametrów „Function”, Rys. 6.49, wybieramy działanie licznika jako następujące:

- Rodzaj liczenia wybierz ‘Frequency’
- Faza pracy wybierz ‘Single phase’
- Kierunek zliczania jest ustalany jako wewnętrzny
- Początkowy kierunek liczenia wybierz ‘Count up’
- Okres pomiaru częstotliwości ustaw na 1,0sek



Rys. 6.49 Wybór funkcji licznika impulsów HSC

W grupie parametrów „I/O addresses”, Rys. 6.50, zdefiniowany jest adres wejścia licznika HSC1



Rys. 6.50 Funkcja HSC

Podane wejście jest częstotliwością sygnału czujnika impulsów w jednostce [1/sek]. W programie podaną wartość należy przepisać do pamięci, zamienić na typ rzeczywisty (operacja czterobitowa) i wyświetlić na panelu HMI. Jeżeli na jeden obrót przypada kilka impulsów wynik dzielimy przez ilość impulsów na jeden obrót. Pomnożenie wyniku przez 60 zmienia jednostkę na [obr/min], Rys. 6.51.



Rys. 6.51 Program LAD – wyliczenie prędkości z licznika impulsów HSC

6.7 Regulacja PID na sterowniku Siemens SIMATIC S7-1200

Środowisko STEP 7 umożliwia wykorzystanie instrukcji sterujących procesorem na sterowniku S7-1200, które pozwalają na prostą konfigurację i zastosowanie regulatora PID. W sterowniku dostępne są dwie instrukcje realizujące regulatory PID:

- instrukcja PID_Compact – wykorzystywana w celu sterowania procesami technicznymi o ciągłych wartościach na wejściu i wyjściu,
- instrukcja PID_3Step wykorzystywana do sterowania urządzeniami napędzanymi za pośrednictwem silników, takimi jak zawory, które potrzebują sygnałów dyskretnych w celu otwierania i zamykania.

Obydwie wyżej wymienione instrukcje mogą obliczać parametry składników P, I oraz D za pośrednictwem wstępnego strojenia - pretuning. W celu dostrojenia "optymalnych" parametrów należy wykorzystać funkcję dostrajania - fine tuning. Użytkownik nie musi w sposób ręczny wykonywać dostrajania parametrów.

W programowaniu sterowników S7-1200 wersja regulatora PID-compact realizuje następujące równanie:

$$Y(s) = K_p \left[(b \cdot W - X) + \frac{1}{T_i s} (W - X) + \frac{T_d s}{a \cdot T_d s + 1} (c \cdot W - X) \right]$$

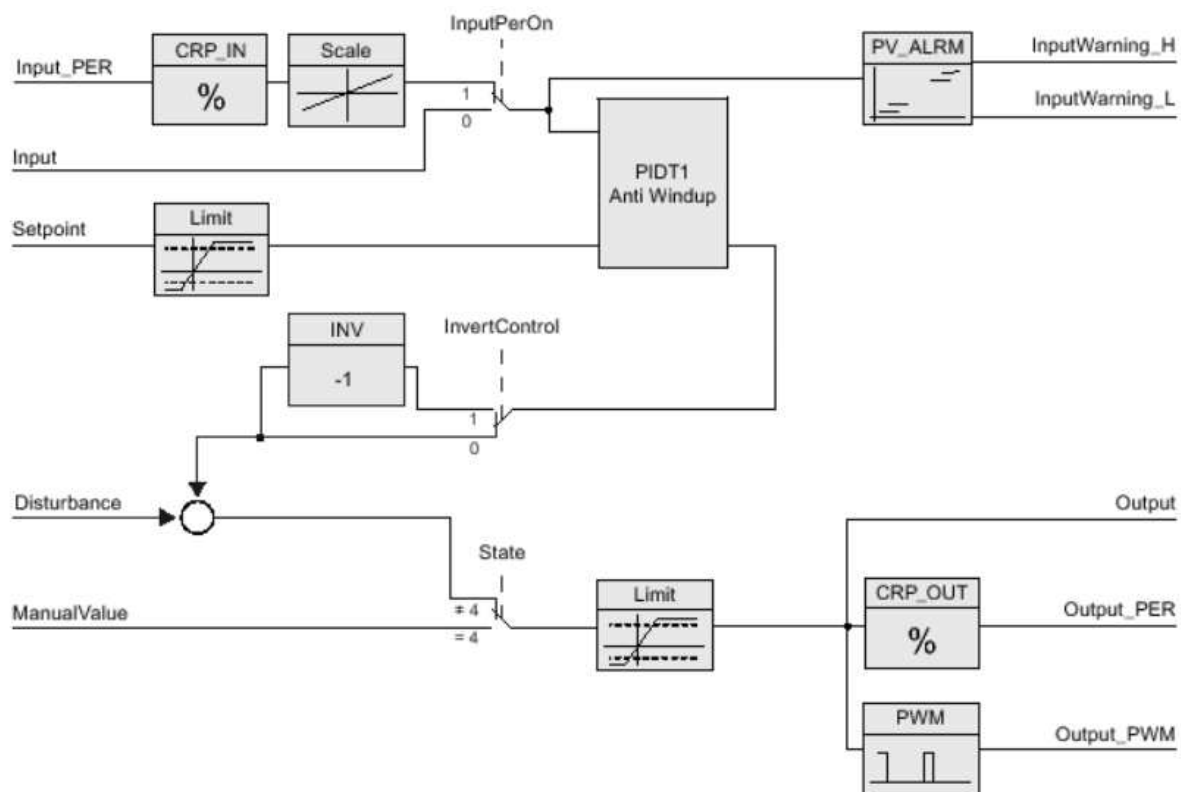
gdzie:

- K_p – wzmacnienie regulatora,
- T_i – akcja całkująca regulatora,
- T_d – akcja różniczkująca regulatora,
- X – sygnał wejścia,
- Y – sygnał wyjścia,
- W – wartość zadana,
- a – współczynnik inercji członu różniczkującego,
- b – współczynnik wagowy członu proporcjonalnego,
- c – współczynnik wagowy członu różniczkującego

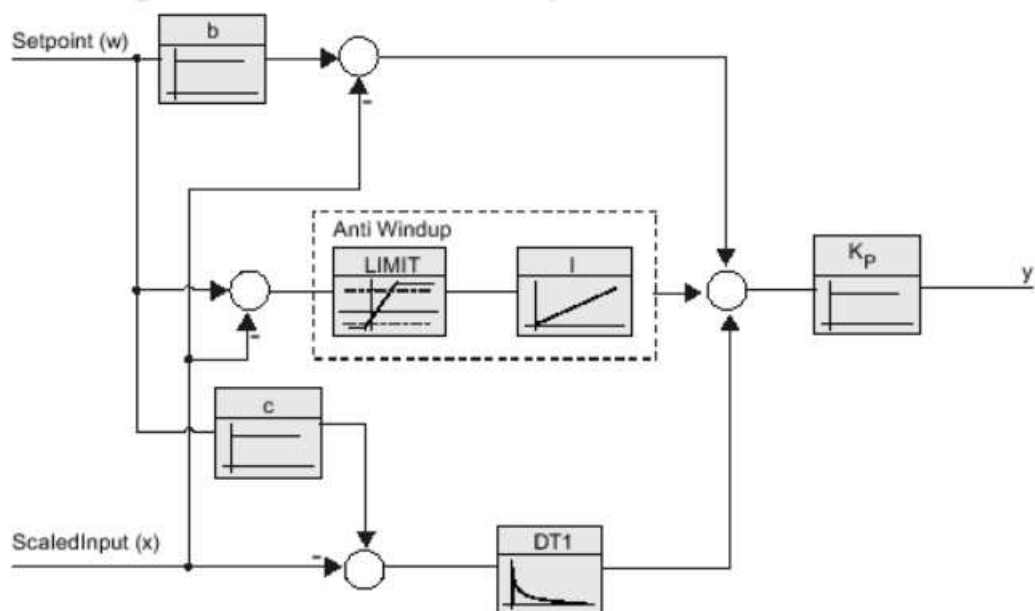
Schemat blokowy regulatora PID_Compact pokazano na Rys. 6.52, a blok anti-windup na Rys. 6.53. Do standardowej transmitancji dodano zmienne wagowe a , b i c , które modyfikują szczegółowo równanie regulatora.

Współczynnik ' a ' ustawia filtrowanie członu różniczkującego, jeżeli $a = 0$ to wprowadzamy czyste różniczkowanie bez filtrowania sygnału uchybu. W procesach rzeczywistych ogranicza się skok odpowiedzi regulatora w jednym kroku iteracji z uwagi wykonalność odpowiedzi na elemencie wykonawczym. Proces regulacji powinien się cechować płynnością przejść w kolejnych krokach. Wartość ' a ' powinna zawierać się od 0 do 1.

Różnica sygnałów $W - X$ oznacza uchyb procesu, jeżeli $b = 1$ i $c = 1$ oznacza to standardowy algorytm regulacji. Jeżeli $b = 0$ i $c = 0$ oznacza to, że algorytm członu proporcjonalnego i różniczkującego obliczany jest tylko od wartości sygnału mierzonego. Są to stosowane modyfikacje w wybranych procesach regulacji.



Rys. 6.52 Schemat blokowy regulatora PID-compact na sterowniku S7-1200



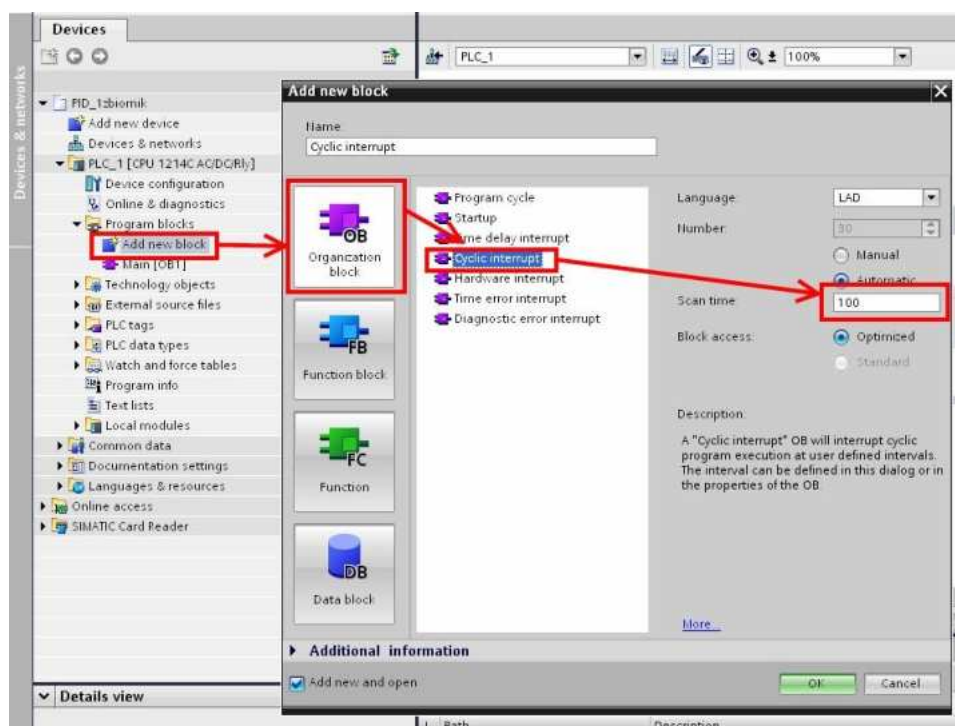
Rys. 6.53 Schemat bloku anti-windup regulatora PID-compact

6.8 Programowanie i parametryzacja regulatora PID na sterowniku Siemens SIMATIC S7-1200

Instrukcje PID należy wykonywać w stałych przedziałach czasowych czasu próbkowania. Zaleca się do tego celu wykorzystać blok programu "Cyclic interrupt" dostępny w blokach organizacyjnych (Organization block). Nie należy wstawiać instrukcji regulatora PID do głównego bloku programu Main [OB1] jak i bloku głównego cyklu programu OB.

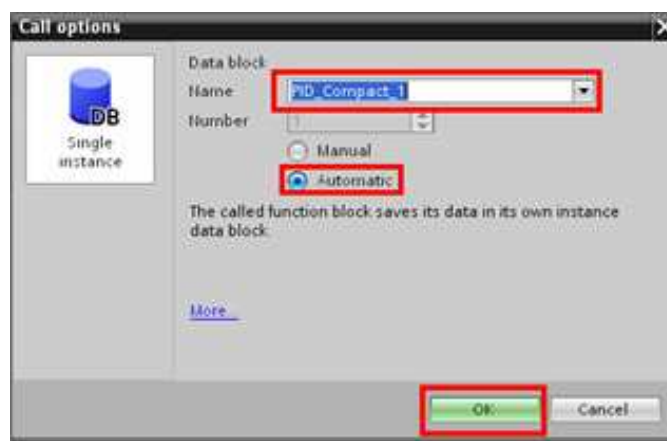
Poniżej przedstawiono przykładowe kroki prowadzące do konfiguracji sterownika oraz regulatora PID:

1. Utworzenie nowego projektu i konfiguracja sterownika
2. Dodanie bloku "**Cyclic interrupt**", Rys. 6.54, oraz wybór czasu skanowania "**Scan time**". Parametr ten podajemy w [ms]. Parametr "**Scan time**" można również zmienić po utworzeniu bloku we właściwościach bloku w opcji **Cyclic interrupt** → **Cyclic time(ms)**.



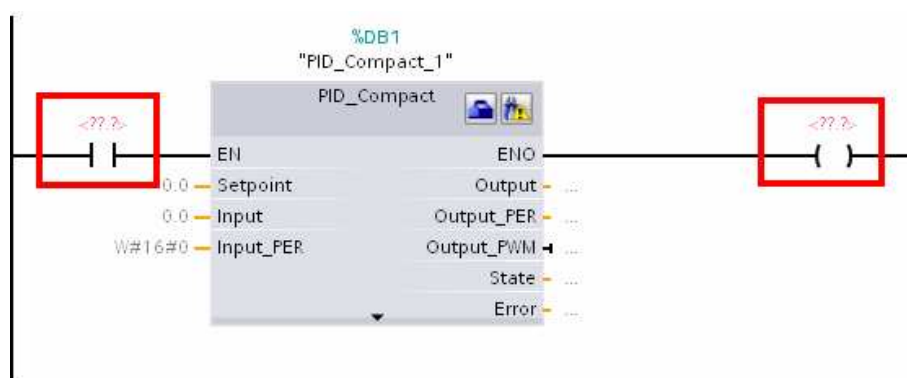
Rys. 6.54 Blok 'Cyclic interrupt'

3. Wstawiamy do bloku "**Cyclic interrupt**" instrukcje "**PID_Compact**". Instrukcję tą znajdziemy w zakładce **Instructions** → **Technology** → **PID Control** → **Compact PID** → **PID_Compact**.
4. Po wstawieniu instrukcji **PID_Compact** aplikacja zgłosi monit o utworzenie nowego **obiektu technologicznego**. Obiekt ten zawiera wszystkie ustawienia dla pętli regulacji. Wybieramy stosowną nazwę obiektu lub zostawiamy domyślną oraz zaznaczamy **automatyczne** przydzielenie **numeru obiektu**. Należy potwierdzić utworzenie obiektu technologicznego przyciskiem **OK**, Rys. 6.55. Obiekty technologiczne występujące w projekcie znaleźć można w grupie **Technology objects** w drzewie projektu (Project tree).



Rys. 6.55 Parametry obiektu technologicznego regulatora PID

5. Tworzymy styk odpowiedzialny za włączanie/wyłączanie regulatora PID oraz cewkę informującą o stanie regulatora, Rys. 6.56.



Rys. 6.56 Styk włączający regulator PID oraz cewka stanu

6. Definiujemy zmienne wejściowe i wyjściowe powiązane z regulatorem PID. Zmienne definiujemy w **tablicy zmiennych (Tag table)**. Można je zdefiniować w **domyślnej tabeli zmiennych (Default Tag table)**. Przykładowe nazwy zmiennych, przykładowe adresy i odpowiednie typy zmiennych podano w Tabela 1. Należy przestrzegać aby typy zmiennych były zgodne z typami wykorzystywanymi przez wejścia / wyjścia regulatora PID. Szczegóły dotyczące dozwolonych typów zmiennych można znaleźć w dokumentacji technicznej sterownika Siemens S7-1200 lub w pomocy dotyczącej instrukcji PID.

Tabela 6.4 Przykładowe nazwy, przykładowe adresy i odpowiednie typy zmiennych regulatora PID

Lp.	Nazwa	Typ	Adres	Opis
1	Włącz_PID	Bool	%I0.1	Styk włączający regulator
2	PID_włączony	Bool	%Q0.0	Cewka informująca o stanie regulatora
3	Wartość_zadana	Real	%MD0	Wartość zadana (SP)
4	Wejście_analogowe	Word	%IW66	Rejestr kanału wejścia analogowego
5	Wyjście_analogowe	Word	%QW80	Rejestr kanału wyjścia analogowego

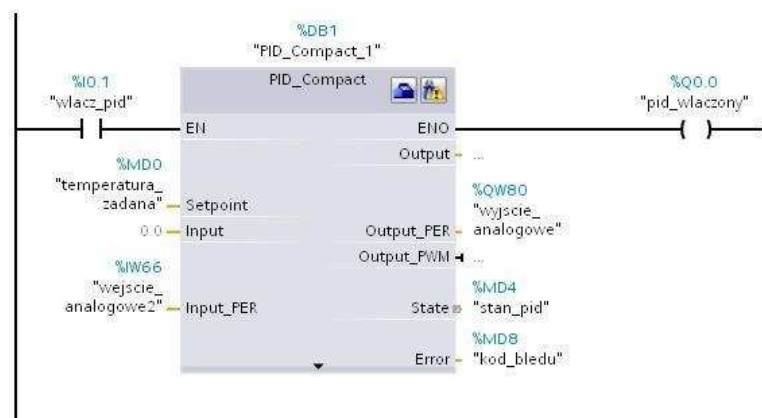
6	Stan_PID	Int	%MW4	Bieżący tryb pracy regulatora PID Możliwe stany: 0 – Inactive, 1 – Pretuning, 2 – Finetuning, 3 – Automatic, 4 – Manual
7	Kod_błędu	DWord	%MD8	Powiadomienie o błędzie. Kod błędu.

Przy adresowaniu zmiennych należy zwrócić szczególną uwagę na liczbę rejestrów zajmowanych przez typ zmiennej. Nie można dopuścić aby adresy zmiennych nakładały się na siebie. W Tabeli 2 zamieszczono podstawowe typy zmiennych oraz liczbę bitów i rejestrów zajmowanych przez odpowiedni typ zmiennej.

Tabela 6.5 Podstawowe typy zmiennych i ilość zajmowanej pamięci

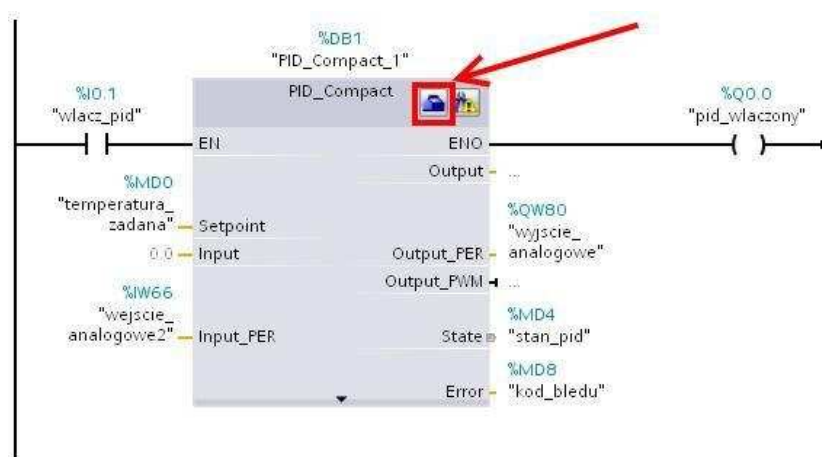
Lp.	Typ zmiennej	Liczba bitów	Ilość zajmowanych rejestrów
1	Real	32	4
2	DWord	32	4
3	Word	16	2
4	Int	16	2
5	Bool	1	Jeden bit rejestru

7. Przypisujemy zmienne w odpowiednie miejsca regulatora, Rys. 6.57.

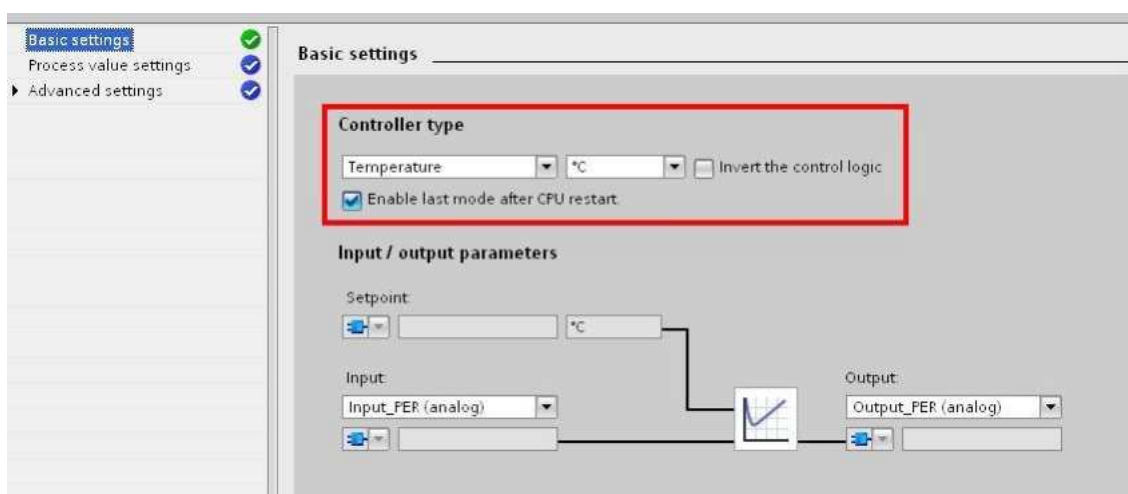


Rys. 6.57 Regulator PID z przypisanymi zmiennymi

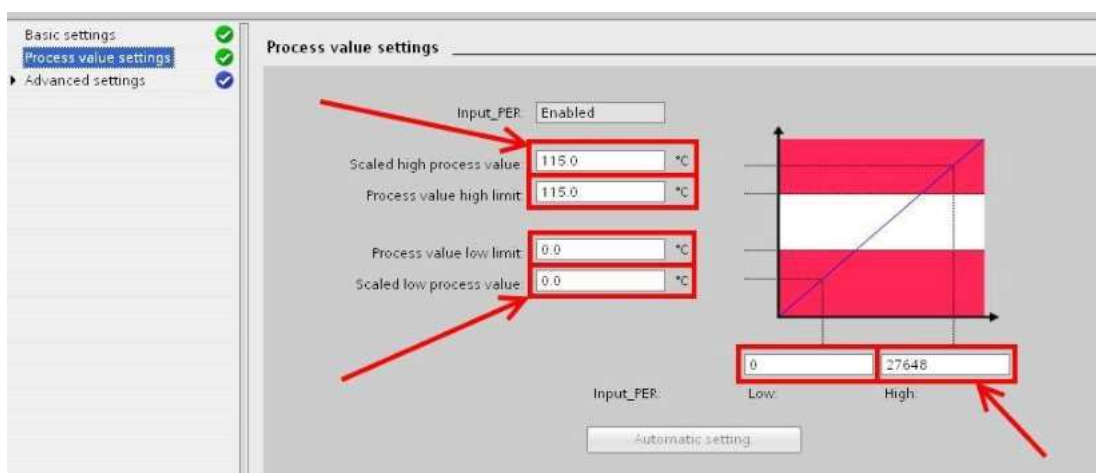
8. Tworzymy nową tablicę **Watch table** i wypełniamy ją zdefiniowanymi wcześniej zmiennymi.
9. Konfigurujemy regulator PID wybierając ikonę , Rys. 6.58. Konfiguracja regulatora została przedstawiona na: Rys. 6.59, Rys. 6.60, Rys. 6.61, Rys. 6.62. Elementy wskazane strzałką należy ustawić zgodnie z przeznaczeniem regulatora.



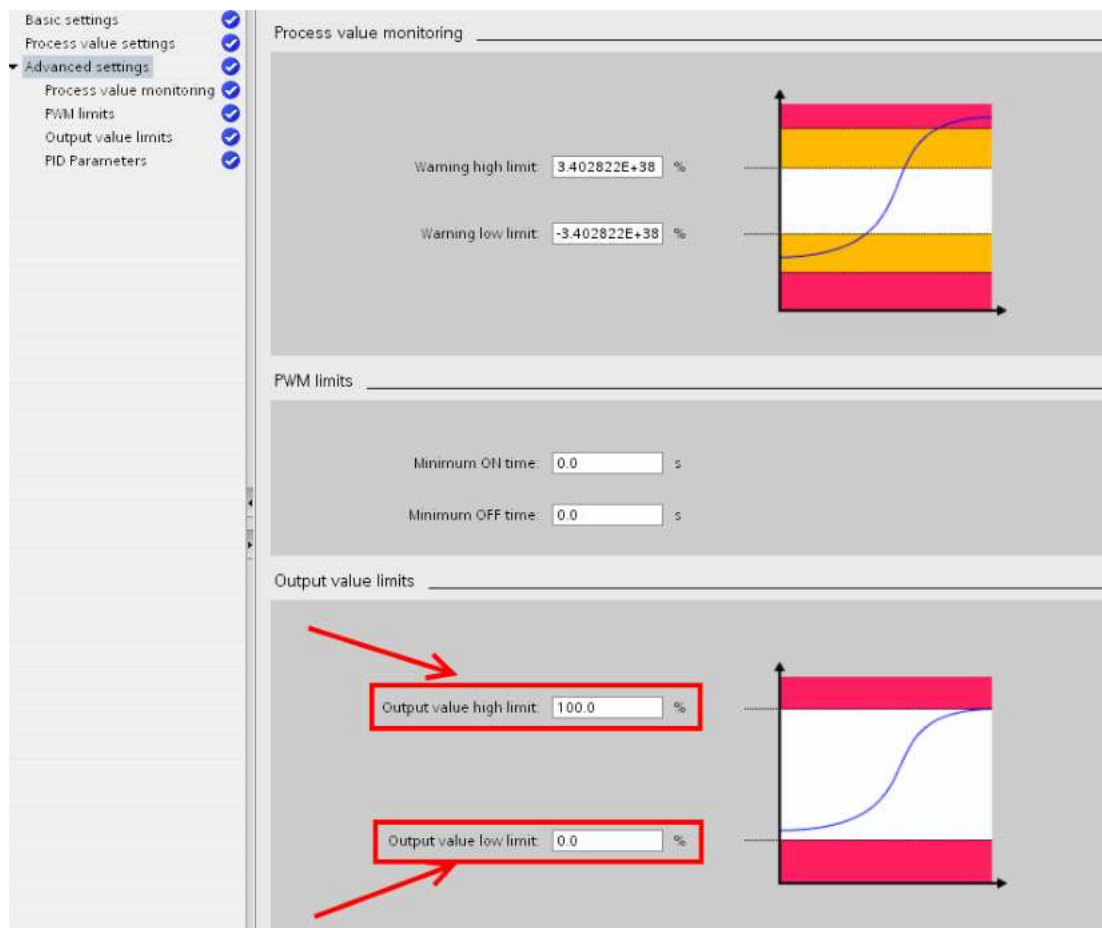
Rys. 6.58 Dostęp do konfiguracji regulatora PID



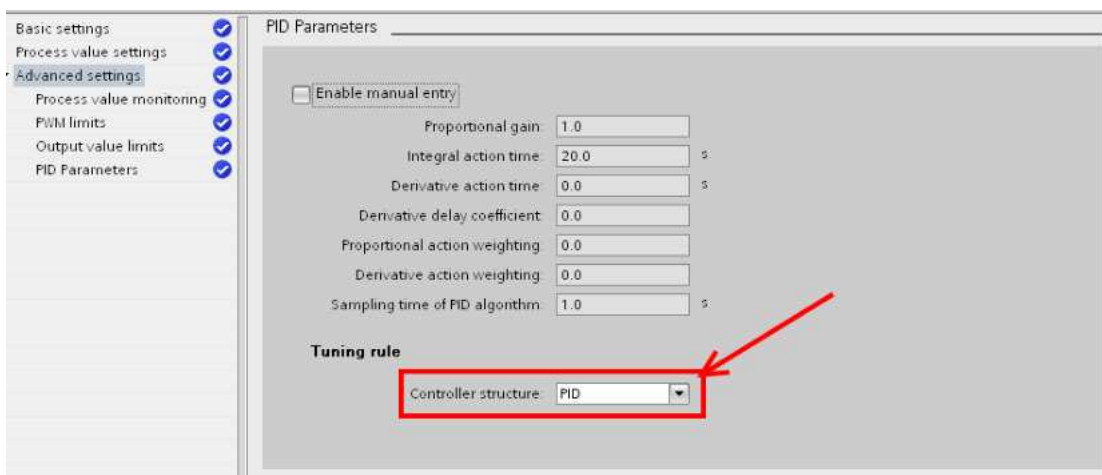
Rys. 6.59 Konfiguracja regulatora PID – podstawowe ustawienia



Rys. 6.60 Konfiguracja regulatora PID – ustawienia skalowania PV



Rys. 6.61 Konfiguracja regulatora PID – ustawienia zaawansowane cz.1

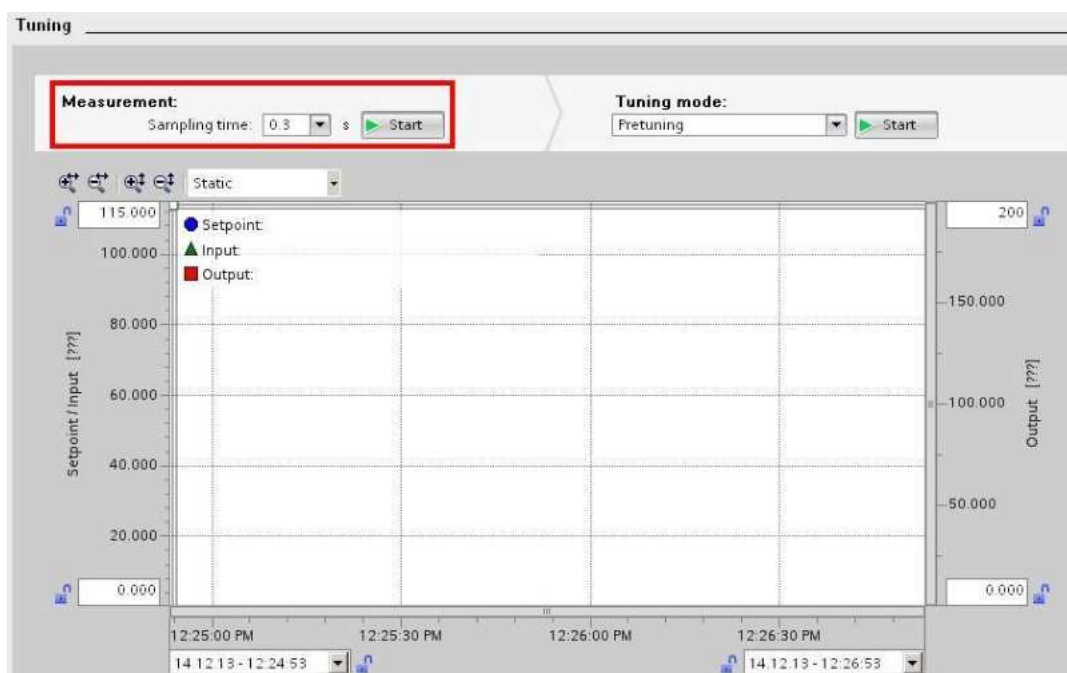


Rys. 6.62 Konfiguracja regulatora PID – ustawienia zaawansowane cz.2

10. Wgrywamy program i konfiguracje na sterownik. Ustawiamy sterownik w tryb **RUN**.
11. Włączamy podgląd programu oraz podgląd zmiennych w tablicy **Watch table**.
12. Włączamy regulator PID (styk odpowiedzialny za włączenie regulatora).
13. W podglądzie programu wchodzimy w **tryb rozruchu regulatora PID**  (PID

commissioning).

14. W oknie rozruchu regulatora PID wybieramy czas próbkowania pomiarów oraz załączamy pomiary przyciskiem **Start**  , Rys. 6.63. Czas próbkowania pomiarów jest jedynie wykorzystywany przez trend obrazujący zmienne SP, PV oraz CV. Nie jest on wykorzystywany w instrukcji regulatora PID.



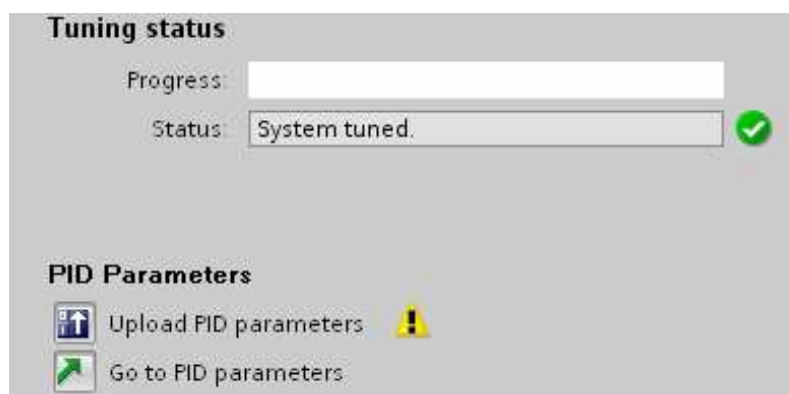
Rys. 6.63 Okno rozruchu regulatora PID

15. Ustanawiamy wartość zadana (SP). Wartość zadana nie powinna znajdować się „blisko” aktualnie odczytywanej wartości procesowej (PV) oraz powinna być większa od aktualnej wartości procesowej PV. Odpowiedni dobór wartości zadanej SP pozwoli na pomyślne przeprowadzenie wstępnego automatycznego Pretuningu parametrów regulatora PID.
16. Wybieramy **tryb tuningu: Pretuning** i naciskamy przycisk **Start**  , Rys. 6.64.



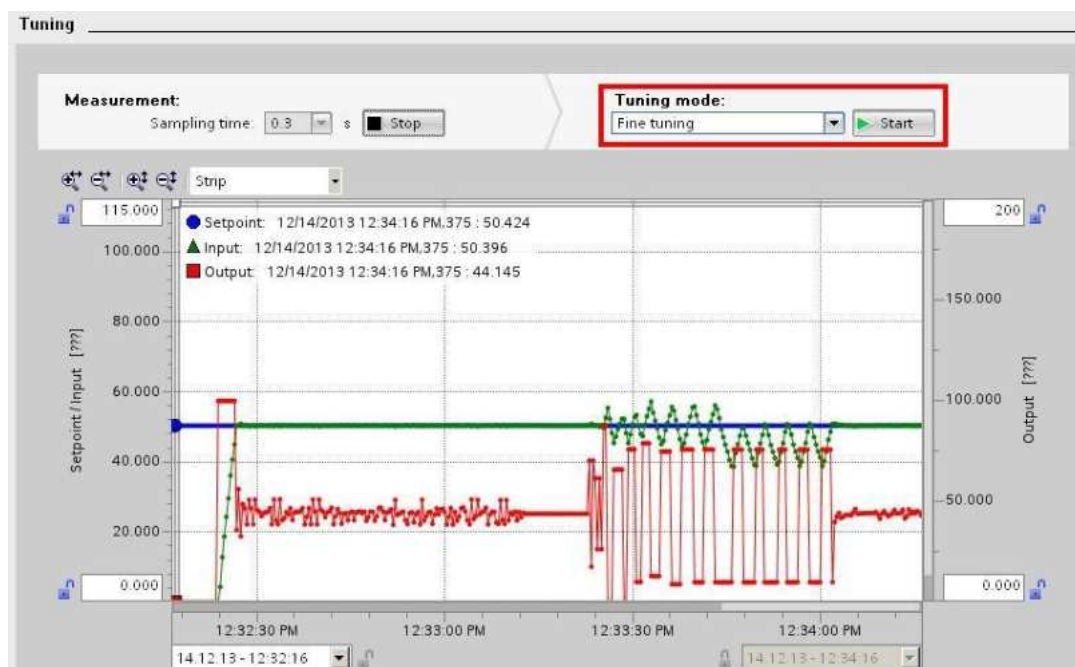
Rys. 6.64 Pretuning regulatora PID

17. Jeżeli **Pretuning** parametrów regulatora przebiegł pomyślnie zostaniemy o tym poinformowani komunikatem „**System tuned**” , Rys. 6.65. Po zakończonym pretuningu przycisk  zamieni się ponownie na przycisk .



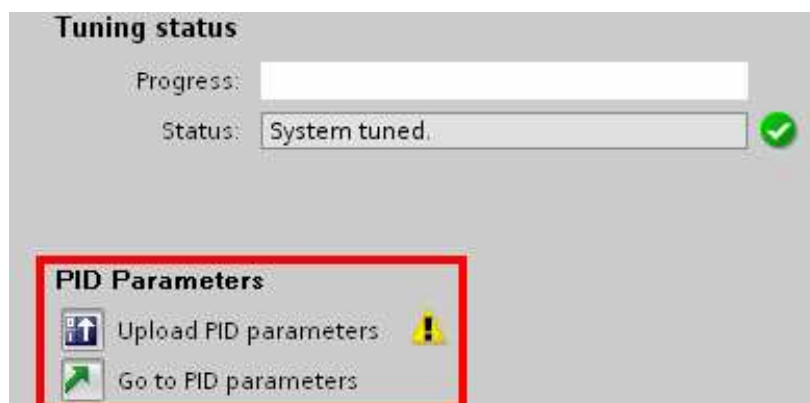
Rys. 6.65 Informacja o pomyślnym zakończeniu procesu Pretuningu

18. Po wykonaniu Pretuningu należy wykonać „Fine tuning” w celu dokładniejszego nastrojenia parametrów regulatora PID, Rys. 6.66. Fine tuning wykonuje się również gdy nie jesteśmy w stanie wykonać Pretuning. Sytuacja taka zachodzi gdy nie można dostatecznie „oddalić” się wartością zadaną od wartości procesowej.



Rys. 6.66 Fine tuning regulatora PID

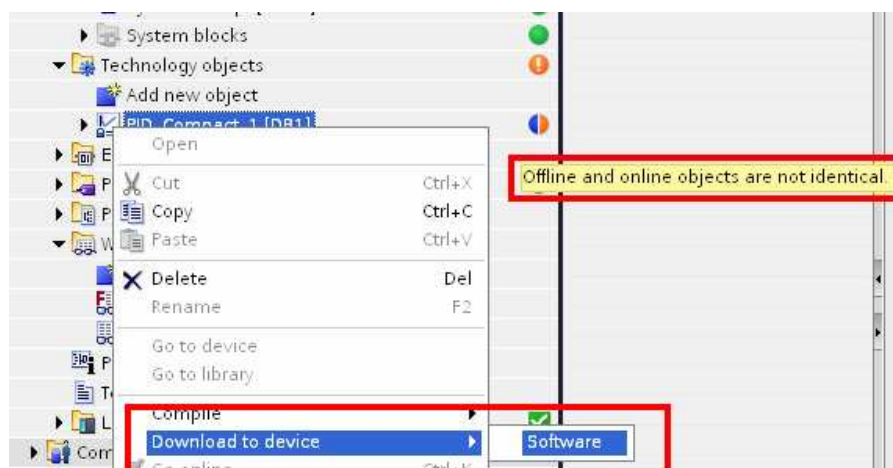
19. Analogicznie jak w punkcie 18 aplikacja poinformuje nas o pomyślnie przeprowadzonym procesie Fine tuningu lub o niepowodzeniu tego procesu.
20. Nastawy regulatora PID dobrane w procesie Pretuningu lub/i Fine tuningu należy wyeksportować do projektu za pomocą polecenia „Upload PID parameters” , Rys. 6.67. Za pomocą polecenia „Go to PID parameters” jest możliwy podgląd dobranych parametrów regulatora PID.



Rys. 6.67 Eksport parametrów regulatora PID do projektu

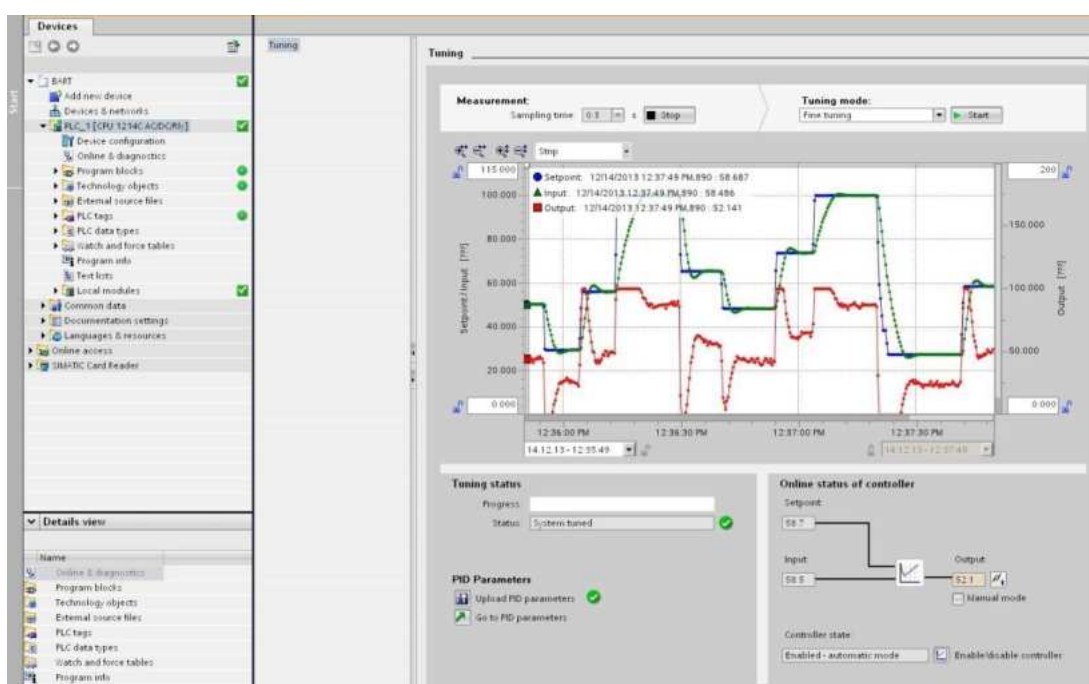
Rys. 18 Eksport parametrów regulatora PID do projektu

21. Po przesłaniu parametrów do projektu należy przesłać zmodyfikowany projekt z powrotem do sterownika aby struktura projektu zgadzała się ze strukturą programu na sterowniku, Rys. 6.68.



Rys. 6.68 Zaktualizowanie programu sterownika

22. Efekt końcowy strojenia regulator PID w sterowniku Siemens S7-1200 przedstawia Rys. 6.69.



Rys. 6.69 Prawidłowo skonfigurowany regulator PID

Spis ilustracji

Rys. 2.1 Pozycja sterownika w procesie sterowania	4
Rys. 2.2 Sterownik kompaktowy LOGO firmy Siemens.....	5
Rys. 2.3 Sterownik VersaMax firmy GE-Fanuc	5
Rys. 2.4 Sterownik firmy Horner serii XI4e	6
Rys. 2.5 Sterownik modułowy firmy GE-Fanuc serii 90-30	6
Rys. 2.6 Sterownik Mitsubishi typu FX3U o zasilaniu AC 100-240 V; zintegrowane wejście 16 DC 24 V i zintegrowane wyjście tranzystorowe 16 DC oraz moduły zewnętrzne	7
Rys. 2.7 Sterownik S7-1200 firmy Siemens zintegrowany oraz z modułami zewnętrznymi	7
Rys. 2.8 Budowa sterowników PLC.....	8
Rys. 2.9 Podłączenie obwodów zewnętrznych sterownika – wyjścia tranzystorowe	9
Rys. 2.10 Podłączenie obwodów zewnętrznych sterownika – wyjścia przekaźnikowe.....	10
Rys. 2.11 Presostat, schemat połączenia, styki presostatu.....	10
Rys. 2.12 Podłączenie obwodu wyjściowego do sterownika z wyjściem przekaźnikowym	12
Rys. 2.13 Przetwornik ciśnienia CCA-300 firmy SIMEX; zakres pomiarowy od -1...0 bar do 0...16 bar, sygnał wyjściowy: 4...20 mA, czujnik piezorezystancyjny krzemowy, dokładność 0,25%, sygnał wyjściowy 4...20 mA (2-przewodowe) lub 0...10V (3-przewodowe)	12
Rys. 2.14 Podłączenie sygnału analogowego do sterownika S7-1200 (modułu analogowo-cyfrowego sterownika)	13
Rys. 2.15 Przetwornik ciśnienia PC-28 Smart z protokołem HART firmy SIMEX;	13
Rys. 2.16 Podłączenie czujnika analogowego z zewnętrznym zasilaniem i sterownikiem	14
Rys. 2.17 Przykład układu pomiarowego analogowego z przetwarzaniem cyfrowym b.....	15
Rys. 2.18 Pomiar ciśnienia – układ rzeczywisty	16
Rys. 2.19 Sterownik S7-1200 i panel operatorski	16
Rys. 2.20 Skalowanie wartości w sterowniku	17
Rys. 2.21 Układ wykonawczy dla wyjścia analogowego sterownika – zawór regulacyjny z ustawnikiem pozycyjnym firmy Siemens	18
Rys. 2.22 Generator sygnału standardowego do diagnostyki obwodów sterownika	18
Rys. 2.23 Panel operatorski z przykładowym procesem firmy Siemens	20
Rys. 2.24 Panel operatorski firmy Mitsubishi z programem chłodni powiantowej	20
Rys. 2.25 Sterownik firmy Horner z panelem HMI, model X5 i X2	21
Rys. 3.1 Algorytm pracy sterownika PLC	22
Rys. 3.2 Sterowanie silnikiem włącz/wyłącz na sterowniku LOGO, GE-Fanuc, S7-1200	26
Rys. 4.1 Sterownik Siemens LOGO i program LOGO Soft Comfort	28
Rys. 4.2 Przykładowy projekt sieci i adresów IP ze sterownikiem LOGO	29
Rys. 4.3 Przykładowy adres IP komputera	29
Rys. 4.4 Przykładowy adres IP sterownika Logo.....	30
Rys. 4.5 Tworzenie nowego bloku UDF	31
Rys. 4.6 Okno edycji bloku UDF	32
Rys. 4.7 Schemat sumatora dwu bitowego z przepełnieniem złożonego z bramek NAND	33
Rys. 4.8 Okno własności bloku UDF, w którym można nazwać wejścia, wyjścia, blok UDF oraz zmienić wielkość ramki	34
Rys. 4.9 Dodawanie bloku UDF do drzewa projektu z menu kontekstowego.....	35
Rys. 4.10 Dodanie bloku UDF do drzewa projektu	35
Rys. 4.11 Przykładowy schemat utworzony z użyciem bloku UDF – Sumator 4bit	36
Rys. 4.12 Timer On-Delay, symbol i wykres czasowy.....	42
Rys. 4.13 Timer Off-Delay, symbol i wykres czasowy	43

Rys. 4.14 Timer On/Off-Delay, symbol i wykres czasowy	43
Rys. 4.15 Timer Retentive On-Delay, symbol i wykres czasowy	44
Rys. 4.16 Wiping relay (pulse output), symbol i wykres czasowy	44
Rys. 4.17 Edge triggered wiping relay, symbol i wykres czasowy	45
Rys. 4.18 Asynchronous Pulse Generator, symbol i wykres czasowy.....	45
Rys. 4.19 Realizacja programu – podtrzymanie wyjścia	46
Rys. 4.20 Realizacja programu – uruchomienie warunkowe	47
Rys. 4.21 Realizacja programu – ignorowanie polecenia ‘START’	47
Rys. 4.22 Realizacja programu – załączanie i wyłączanie pojedynczym przyciskiem	48
Rys. 4.23 Realizacja programu – załączenie w trybie praca	48
Rys. 4.24 Realizacja programu – blokada przeciwzwarciowa LEWO/PRAWO.....	49
Rys. 4.25 Realizacja programu – dwuręczne załączenie	49
Rys. 4.26 Realizacja programu – opóźnienie załączenia 1	50
Rys. 4.27 Realizacja programu – opóźnienie załączenia 2	50
Rys. 4.28 Realizacja programu – opóźnienie wyłączenia	50
Rys. 4.29 Realizacja programu – miganie lampki sygnalizacyjnej	51
Rys. 4.30 Realizacja programu – miganie i świecenie lampki sygnalizacyjnej	51
Rys. 4.31 Realizacja programu – chwilowa blokada powtórnego załączenia 1.....	52
Rys. 4.32 Realizacja programu – chwilowa blokada powtórnego załączenia 2.....	52
Rys. 4.33 Realizacja programu – wstrzymanie pracy urządzenia	53
Rys. 4.34 Realizacja programu – zliczanie zdarzeń	53
Rys. 4.35 Program FBD, suma dwóch bitów z przeniesieniem, I1 i I2 – pojedyncze bity, I3 – bit z przeniesienia	54
Rys. 4.36 Program LAD, suma dwóch bitów z przeniesieniem, I1 i I2 – pojedyncze bity, I3 – bit z przeniesienia	55
Rys. 4.37 Program na sekwencyjne załączanie świateł drogowych – bloki FBD	56
Rys. 4.38 Program na sekwencyjne załączanie świateł drogowych – program LAD	57
Rys. 4.39 Regulator PI na sterowniku LOGO! – bloki FBD.....	58
Rys. 4.40 Linia trendu wyjścia regulatora PI sterownika LOGO	58
Rys. 4.41 Regulator PI na sterowniku LOGO – program LAD	59
Rys. 5.1 Sterownik modułowy GE-Fanuc 90-30	60
Rys. 5.2 Dostępne zasilacze sterownika GE-Fanuc 90 - 30.....	61
Rys. 5.3 Dodawanie zasilacza i konfiguracja sterownika GE-Fanuc 90 - 30.....	62
Rys. 5.4 Dostępne jednostki CPU sterownika GE-Fanuc 90-30	63
Rys. 5.5 Dodawanie jednostki CPU i konfiguracja sterownika GE-Fanuc 90-30	63
Rys. 5.6 Dostępne moduły wejść cyfrowych sterownika GE-Fanuc 90-30	64
Rys. 5.7 Dodawanie modułów i konfiguracja wejść/wyjść sterownika GE-Fanuc 90-30	65
Rys. 5.8 Opcja RETENTIVE dla zmiennych %Q i %M dla sterownika GE-Fanuc 90 - 30	68
Rys. 5.9 Timer OFDT opóźnienie wyłączenia dla sterownika GE-Fanuc 90 - 30	70
Rys. 5.10 Timer ONDTR opóźnienie załączenia z pamięcią dla sterownika GE-Fanuc 90 - 30.....	70
Rys. 5.11 Timer TMR opóźnienie załączenia dla sterownika GE-Fanuc 90 - 30.....	71
Rys. 5.12 Liczniki UPCTR i DNCTR	71
Rys. 5.13 Przesyłanie danych Move dla sterownika GE-Fanuc 90 - 30.....	73
Rys. 5.14 Zamiana kodu BCD dla sterownika GE-Fanuc 90 - 30	73
Rys. 5.15 Zamiana typu integer dla sterownika GE-Fanuc 90 - 30	74
Rys. 5.16 Zamiana typu real dla sterownika GE-Fanuc 90 - 30	74
Rys. 5.17 Zamiana typu dinteger dla sterownika GE-Fanuc 90 - 30.....	75
Rys. 5.18 Zamiana typu real z zaokrągleniem dla sterownika GE-Fanuc 90 - 30.....	75

Rys. 5.19 Zamiana stopni na radiany i odwrotnie dla sterownika GE-Fanuc 90 - 30.....	76
Rys. 5.20 Podstawowe funkcje logiczne dla sterownika GE-Fanuc 90 - 30	76
Rys. 5.21 Funkcja dodawania liczb całkowitych ADD dla sterownika GE-Fanuc 90 - 30.....	77
Rys. 5.22 Funkcja odejmowania liczb całkowitych SUB dla sterownika GE-Fanuc 90 - 30	77
Rys. 5.23 Funkcja mnożenia liczb całkowitych MUL dla sterownika GE-Fanuc 90 - 30	78
Rys. 5.24 Funkcja dzielenia liczb całkowitych DIV dla sterownika GE-Fanuc 90 - 30	78
Rys. 5.25 Funkcja dzielenia liczb rzeczywistych DIV dla sterownika GE-Fanuc 90 - 30	79
Rys. 5.26 Funkcja dzielenia liczb całkowitych MOD (reszta z dzielenia) dla sterownika GE-Fanuc 90 - 30	79
Rys. 5.27 Funkcja trygonometryczna SIN dla sterownika GE-Fanuc 90 - 30	80
Rys. 5.28 Funkcja trygonometryczna odwrotna ASIN dla sterownika GE-Fanuc 90 - 30.....	80
Rys. 5.29 Funkcja matematyczna logarytm LN dla sterownika GE-Fanuc 90 - 30	81
Rys. 5.30 Funkcja matematyczna wykładnicza EXPT dla sterownika GE-Fanuc 90 - 30	81
Rys. 5.31 Funkcja matematyczna pierwiastek kwadratowy SQRT dla sterownika GE-Fanuc 90 - 30	82
Rys. 5.32 Funkcja skalowanie wartości SCALE dla sterownika GE-Fanuc 90 - 30	82
Rys. 5.33 Funkcja przesuwania w lewo SHIFTL dla sterownika GE-Fanuc 90 - 30	83
Rys. 5.34 Funkcja przesuwania w prawo SHIFTR dla sterownika GE-Fanuc 90 - 30	83
Rys. 5.35 Funkcja rotacji w lewo ROL dla sterownika GE-Fanuc 90 - 30	84
Rys. 5.36 Funkcja rotacji w prawo ROR dla sterownika GE-Fanuc 90 - 30	84
Rys. 5.37 Prosty układ regulacji, $y(t)$ – wielkość wyjściowa (regulowana), $u(t)$ – sygnał sterujący, $w(t)$ – wartość zadana, $e(t)$ – uchyb sterowania, $z(t)$ - zakłócenie.....	85
Rys. 5.38 Schemat blokowy regulatora standardowego PID-ISA dla sterownika GE-Fanuc 90 - 30	85
Rys. 5.39 Schemat blokowy regulatora o funkcjach rozdzielonych PID-IND dla sterownika GE-Fanuc 90 - 30	86
Rys. 5.40 Regulator PID na sterowniku GE-Fanuc 90-30 w wersji standardowej i zmienionej – program LAD.....	86
Rys. 5.41 Blok regulator PID na sterowniku GE-Fanuc 90-30 z opisem sygnałów.....	87
Rys. 5.42 Okno doboru parametrów regulatora PID w sterowniku GE-Fanuc	91
Rys. 5.43 Odczyt parametrów odpowiedzi obiektu – charakterystyka skokowa	93
Rys. 6.1 Zestaw Sterownika Siemens SIMATIC S7-1200.....	96
Rys. 6.2 Start programu TIA Portal	97
Rys. 6.3 Strona startowa programu TIA Portal	97
Rys. 6.4 Strona z menu tworzenia projektu i konfiguracji sterownika	98
Rys. 6.5 Strona menu z wyborem sterownika	98
Rys. 6.6 Strona z wyborem sterownika, dostępny w laboratorium	99
Rys. 6.7 Start programu TIA Portal	99
Rys. 6.8 Konfiguracja portu komunikacyjnego PROFINET	100
Rys. 6.9 Wprowadzenie nowego adresu sieciowego w TIA Portal.....	100
Rys. 6.10 Składanie zestawu sterownika z modułów w TIA Portal	101
Rys. 6.11 Ustalenie parametrów analogowego portu wyjściowego %QW80	102
Rys. 6.12 Ustalenie parametrów analogowego portu wejściowego %IW64.....	102
Rys. 6.13 Ustalenie adresów wejść binarnych w TIA Portal	103
Rys. 6.14 Ustalenie adresów wejść binarnych w TIA Portal	103
Rys. 6.15 Zapis programu sterownika w TIA Portal	104
Rys. 6.16 Wybór i kontrola sterownika w sieci	104
Rys. 6.17 Panel HMI KPT700 Basic PN	105
Rys. 6.18 Wybór panelu operatorskiego TIA Portal	106
Rys. 6.19 Połączenie panelu operatorskiego ze sterownikiem w TIA Portal	107

Rys. 6.20 Wprowadzenie adresu IP panelu operatorskiego w TIA Portal	107
Rys. 6.21 Menu panelu operatorskiego.....	108
Rys. 6.22 Parametry panelu operatorskiego	108
Rys. 6.23 Kolejne parametry panelu operatorskiego.....	109
Rys. 6.24 Programowanie panelu operatorskiego w TIA Portal.....	110
Rys. 6.25 Projektowanie zegara na panelu operatorskim.....	110
Rys. 6.26 Dostępne liczniki w TIA Portal	112
Rys. 6.27 Przebieg czasowy sygnałów wejściowych i wyjściowych na liczniku CTU	112
Rys. 6.28 Przebieg czasowy sygnałów wejściowych i wyjściowych na liczniku CTD	113
Rys. 6.29 Przebieg czasowy sygnałów wejściowych i wyjściowych na liczniku CTUD	114
Rys. 6.30 Timery dostępne TIA Portal STEP 7	115
Rys. 6.31 Przebieg czasowy timera TP	115
Rys. 6.32 Przebieg czasowy timera TOF.....	116
Rys. 6.33 Przebieg czasowy timera TON	116
Rys. 6.34 Przebieg czasowy timera TONR.....	117
Rys. 6.35 Pomiar prędkości obrotowej wału z kierunkiem obrotu	118
Rys. 6.36 Tarcza kodowa 12-bitowa i pomiar położenia wału	119
Rys. 6.37 Pomiar prędkości wału w przypadku jednego impulsu na obrót.....	120
Rys. 6.38 Konfiguracja sprzętu do pomiaru prędkości – podłączenie czujnika impulsów.....	120
Rys. 6.39 Wybór licznika HSC i jego portu	121
Rys. 6.40 Włączanie licznika impulsów HSC	121
Rys. 6.41 Funkcje licznika impulsów HSC	122
Rys. 6.42 Wejście sprzętowe licznika impulsów HSC	122
Rys. 6.43 Konfiguracja standardowa adresu wejścia cyfrowego	122
Rys. 6.44 Filtry wejściowe dla generatora zegara	123
Rys. 6.45 Wywołanie instrukcji „CTRL_HSC_EXT”	123
Rys. 6.46 Struktura typu danych systemowych „HSC_Period”	124
Rys. 6.47 Dodawanie bloku funkcyjnego	125
Rys. 6.48 Wywołanie „CalcSpeed”	125
Rys. 6.49 Wybór funkcji licznika impulsów HSC.....	127
Rys. 6.50 Funkcja HSC.....	127
Rys. 6.51 Program LAD – wyliczenie prędkości z licznika impulsów HSC	127
Rys. 6.52 Schemat blokowy regulatora PID-compact na sterowniku S7-1200.....	129
Rys. 6.53 Schemat bloku anti-windup regulatora PID-compact	129
Rys. 6.54 Blok ‘Cyclic interrupt’	130
Rys. 6.55 Parametry obiektu technologicznego regulatora PID.....	131
Rys. 6.56 Styk włączający regulator PID oraz cewka stanu	131
Rys. 6.57 Regulator PID z przypisanymi zmiennymi.....	131
Rys. 6.58 Dostęp do konfiguracji regulatora PID	131
Rys. 6.59 Konfiguracja regulatora PID – podstawowe ustawienia	131
Rys. 6.60 Konfiguracja regulatora PID – ustawienia skalowania PV.....	131
Rys. 6.61 Konfiguracja regulatora PID – ustawienia zaawansowane cz.1	131
Rys. 6.62 Konfiguracja regulatora PID – ustawienia zaawansowane cz.2	131
Rys. 6.63 Okno rozruchu regulatora PID.....	131
Rys. 6.64 Pretuning regulatora PID.....	131
Rys. 6.65 Informacja o pomyślnym zakończeniu procesu Pretuningu	131
Rys. 6.66 Fine tuning regulatora PID.....	131
Rys. 6.67 Eksport parametrów regulatora PID do projektu	131

Rys. 6.68 Zaktualizowanie programu sterownika	131
Rys. 6.69 Prawidłowo skonfigurowany regulator PID	131