

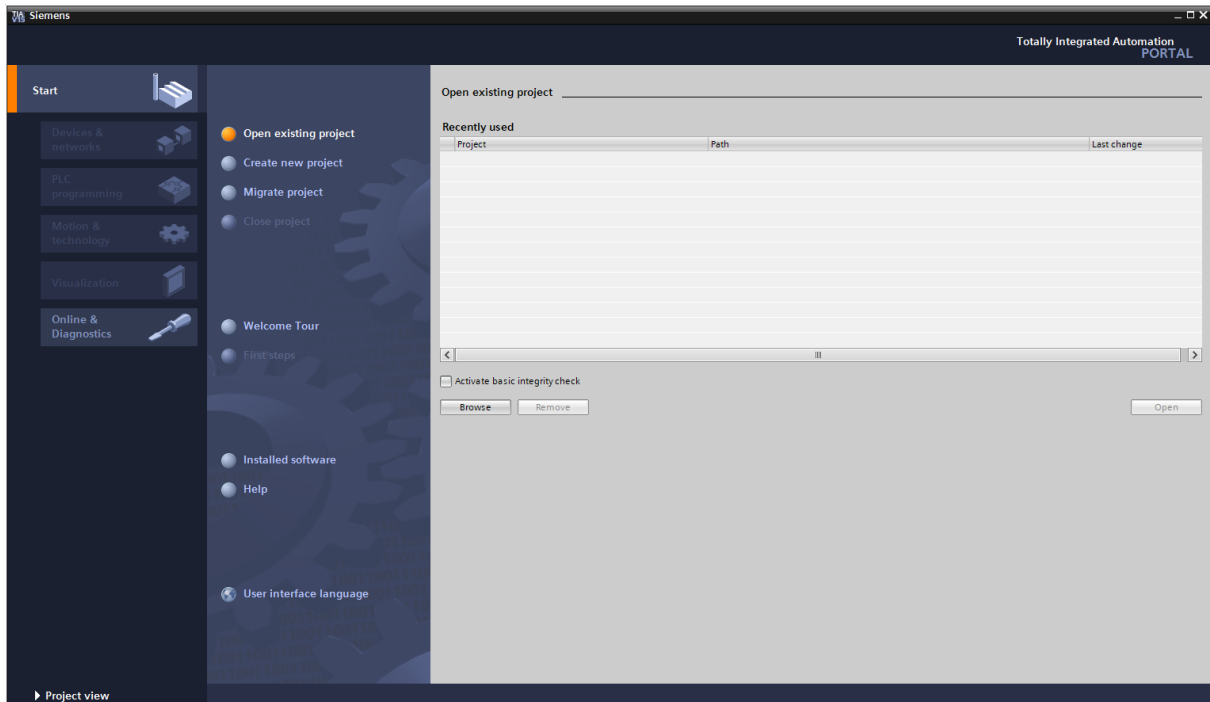
## PROGRAMOWALNE SYSTEMY MECHATRONIKI

Laboratorium nr 2, 3

**Podstawy programowania sterowników PLC – operacje na bitach**

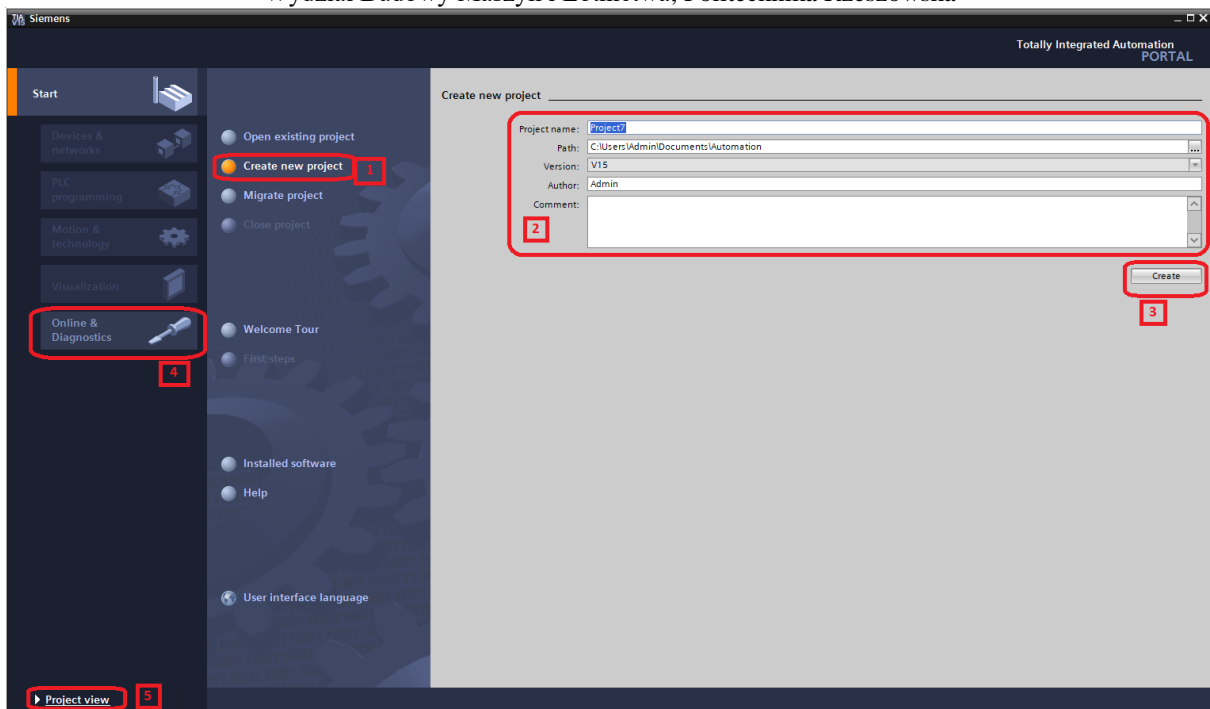
## 1. Tworzenie nowego projektu w TIA Portal V18

Po uruchomieniu programu TIA Portal V18 otwiera się główne okno programu pokazane na rys. 1.



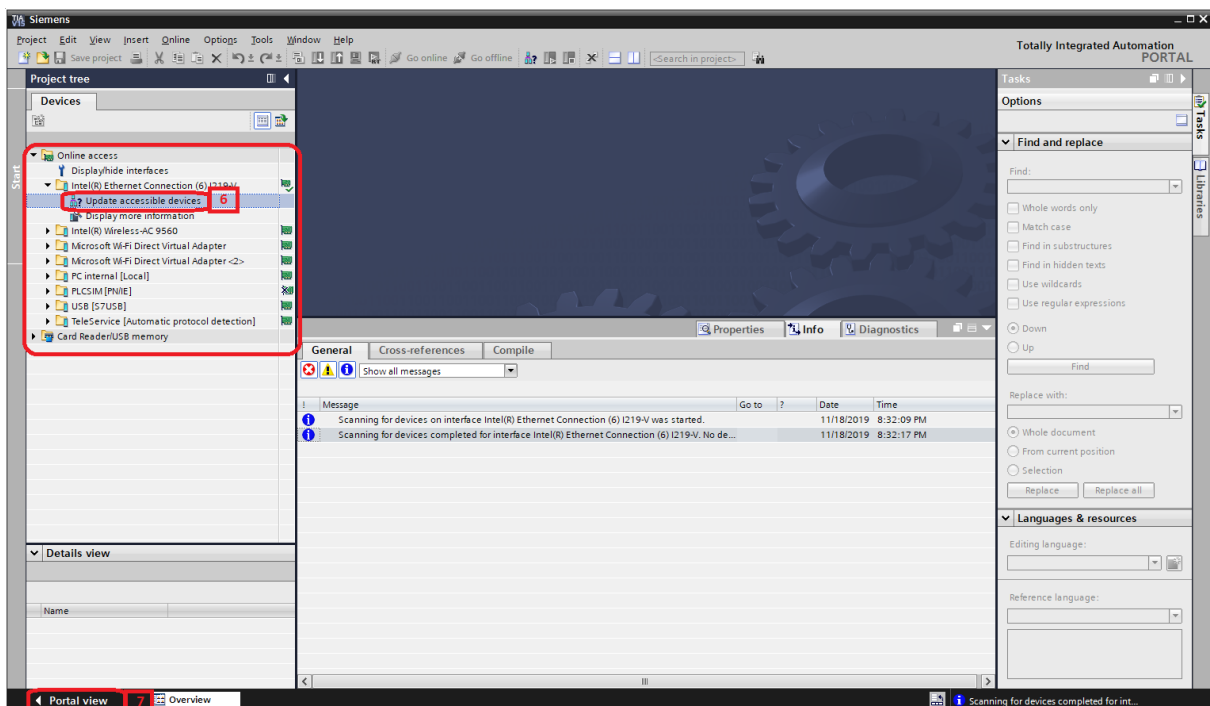
Rys. 1. Główne okno programu TIA Portal V18

Aby utworzyć nowy projekt, należy wybrać *Create new project* (1) rys. 2, następnie wpisać niezbędne dane identyfikujące projekt, jak nazwa, autor czy komentarz, można zmienić ścieżkę dostępu do projektu (2). Następnie należy kliknąć *Create* (3), aby przejść do kolejnego etapu tworzenia projektu. Z tego okna programu można również przejść do sekcji *Online and Diagnostics* (4) i połączyć się z dostępnymi stacjami bez tworzenia projektu, czy przełączyć się z okna portalu (*Portal view*) na okno projektu (*Project view*) (5).



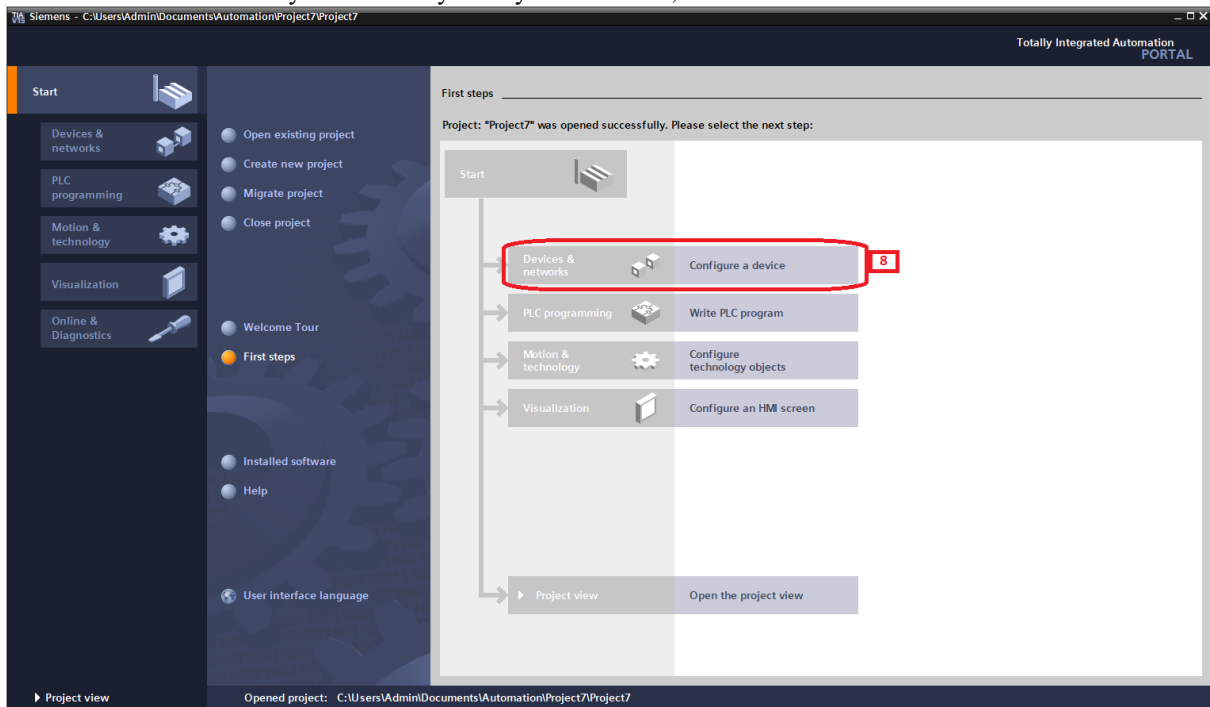
Rys. 2. Etapy tworzenia nowego projektu 1

Po przejściu do widoku *Project view* mamy możliwość wyszukania dostępnych urządzeń poprzez wybrane interfejsy komunikacji (6), lub powrót do okna portalu (7).



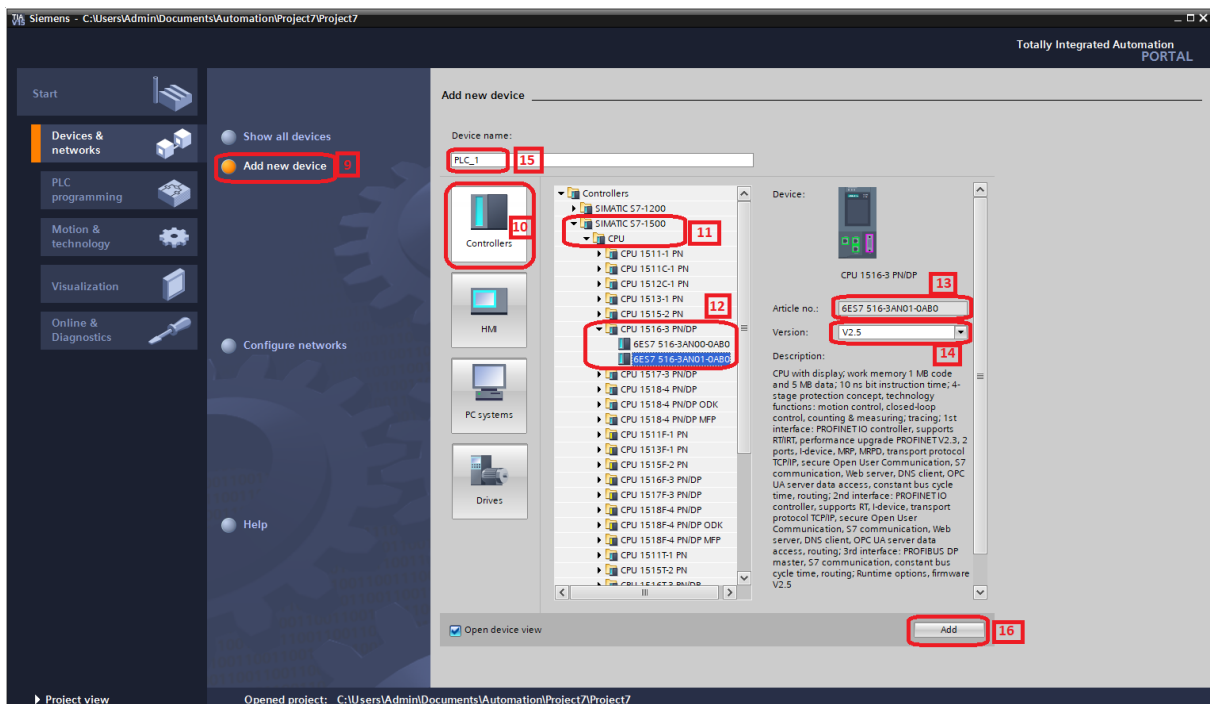
Rys. 3. Okno *Project view* bez utworzonego projektu

Po utworzeniu nowego projektu można przejść do konfiguracji stacji (8) rys. 4, lub tworzenia programu.



Rys. 4. Możliwe etapy tworzenia projektu

Po wyborze etapu tworzenia konfiguracji sprzętowej mamy możliwość dodania nowego urządzenia do projektu (*Add new device*) (9) rys. 5. Może to być sterownik (10), panel HMI, system oparty o podzespoły PC, czy przemiennik częstotliwości. Wybieramy PLC, następnie dokonujemy wyboru rodziny sterowników (11), konkretnej jednostki CPU z danej rodziny (12) po numerze zamówienia (13) uwzględniając wersję oprogramowania firmowego zainstalowanego w fizycznym sterowniku na stanowisku (14). Możemy zmienić nazwę stacji (15), kończymy ten etap konfiguracji klikając *Add* (16).

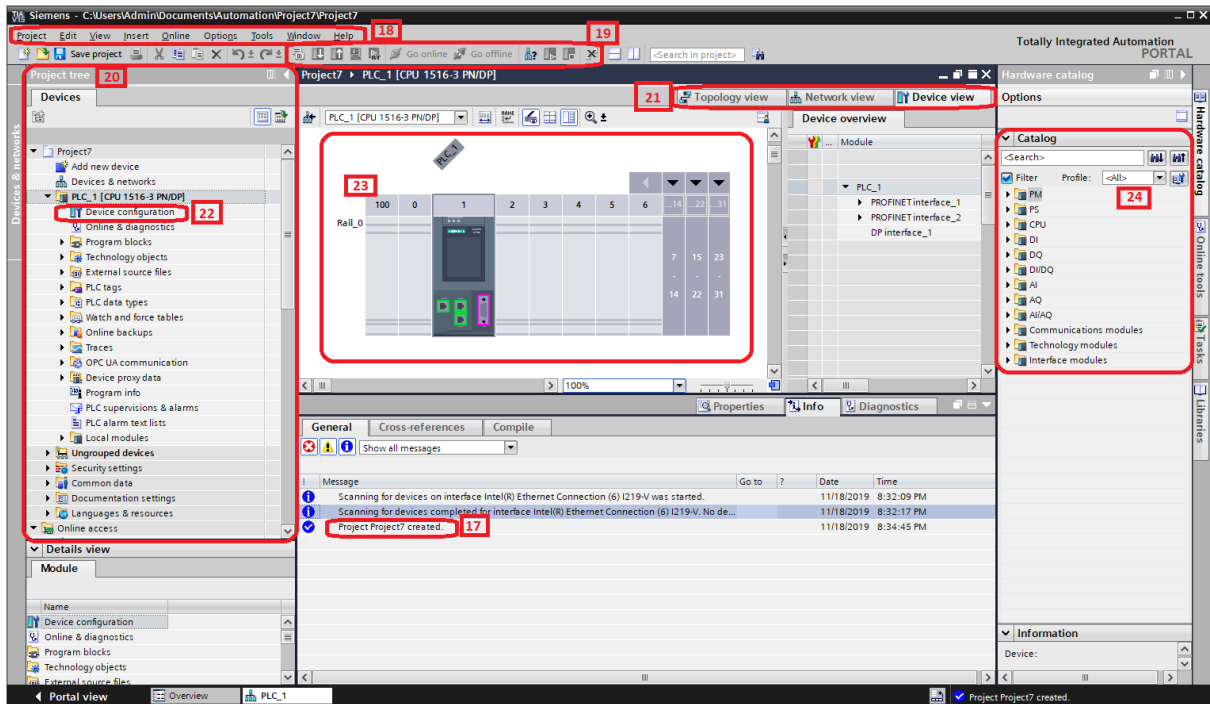


Rys. 5. Dodawanie urządzenia do projektu

## Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

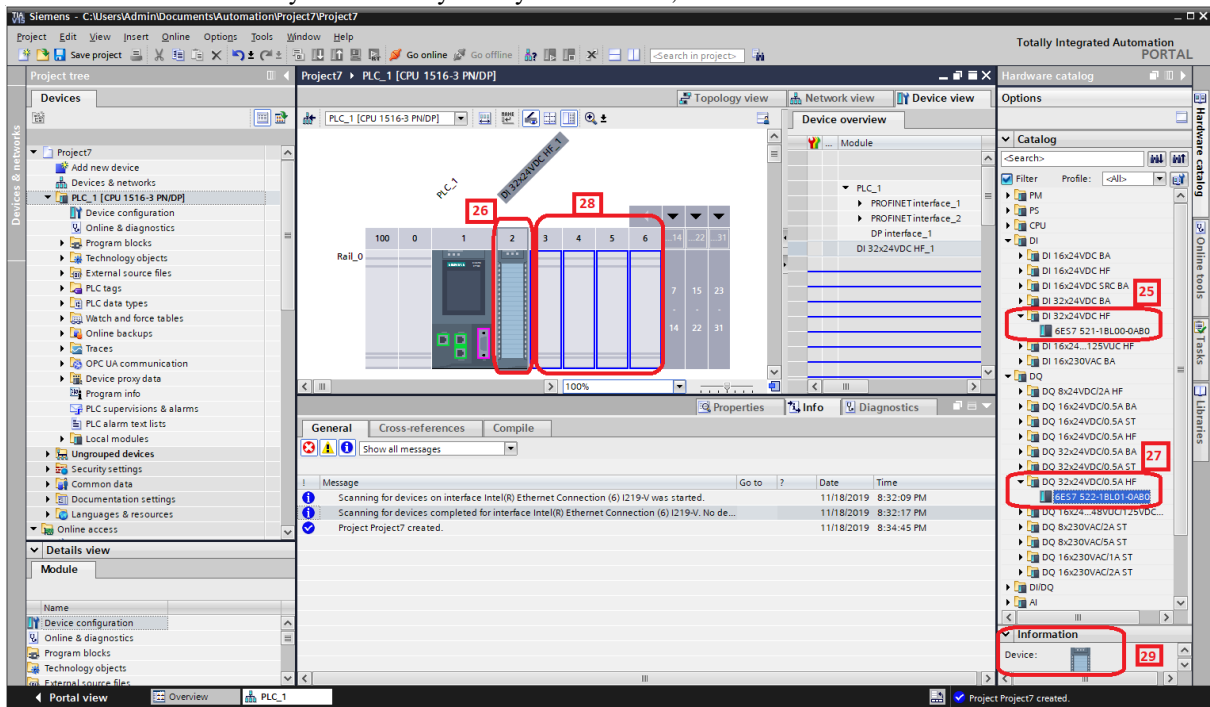
Następuje przełączenie do trybu edycji projektu ze stacją zbudowaną w oparciu o wybraną jednostkę CPU w oknie *Project view* (rys. 6). W dolnej części ekranu, zakładce Info otrzymujemy informację, że projekt został utworzony poprawnie (17). W oknie tym mamy standardowe mniej lub bardziej rozbudowane menu rozwijane (18), najczęściej używane opcje związane z transferem informacji do/z PLC, umożliwiające przejście w tryb pracy on-line, czy zatrzymanie lub wznowienie pracy PLC (19). Po lewej stronie znajduje się okno zawierające drzewo projektu (20). Po wyborze gałęzi *Device configuration* (22) w centralnej części okna/ części roboczej pojawią się trzy zakładki umożliwiające konfigurację stacji, interfejsów oraz połączeń sieciowych (21). Po otwarciu zakładki *Device view* mamy możliwość konfiguracji stacji, używając odpowiednich modułów rozszerzeń z katalogu znajdującego się po prawej stronie okna (24).



Rys. 6. Etapy konfiguracji stacji 1

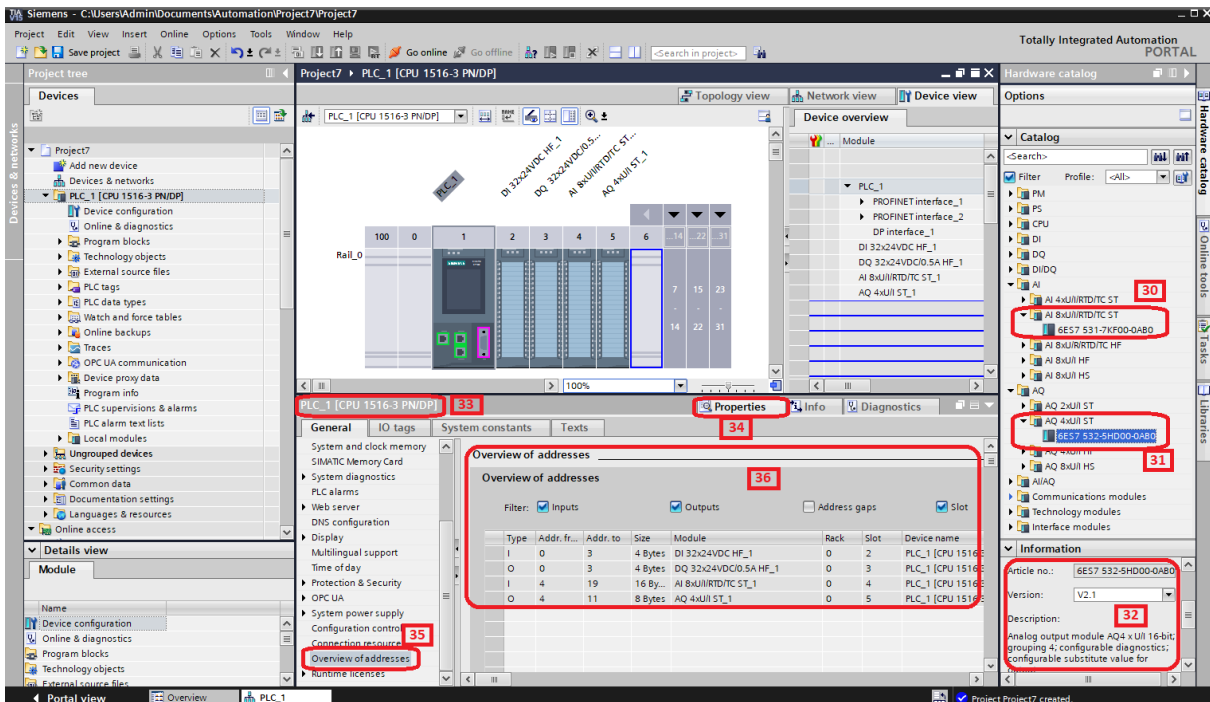
Wybieramy z katalogu moduł, który znajduje się w stacji fizycznej, biorąc pod uwagę numer zamówienia oraz wersję oprogramowania firmowego, a następnie dwukrotnie klikamy, lub przeciągamy na odpowiedni slot stacji (25),(26),(27). Kliknięcie danego modułu powoduje podświetlenie na niebiesko slotów, w których dany moduł może zostać umieszczony (28). Ponadto w prawym dolnym rogu okna wyświetlana jest sekcja *Information*, podająca skrócone informacje na temat wybranego modułu.

**Katedra Mechaniki Stosowanej i Robotyki**  
Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska



Rys. 7. Etapy konfiguracji stacji 2

Dodajemy kolejne moduły stacji (30),(31),(32). Na tym etapie konfiguracji możemy uzyskać bardzo szerokie informacje na temat poszczególnych modułów. Szczególnie istotne są kwestie związane z adresowaniem wejść/wyjść poszczególnych modułów. Informację na temat adresowania modułów możemy uzyskać klikając dwukrotnie na jednostkę CPU, wyświetlając zakładkę *Properties* (34) jednostki CPU (33), wybierając z przewijanego menu *Overview of addresses* (35). Adresy poszczególnych modułów zostaną przedstawione w formie tabeli (36).

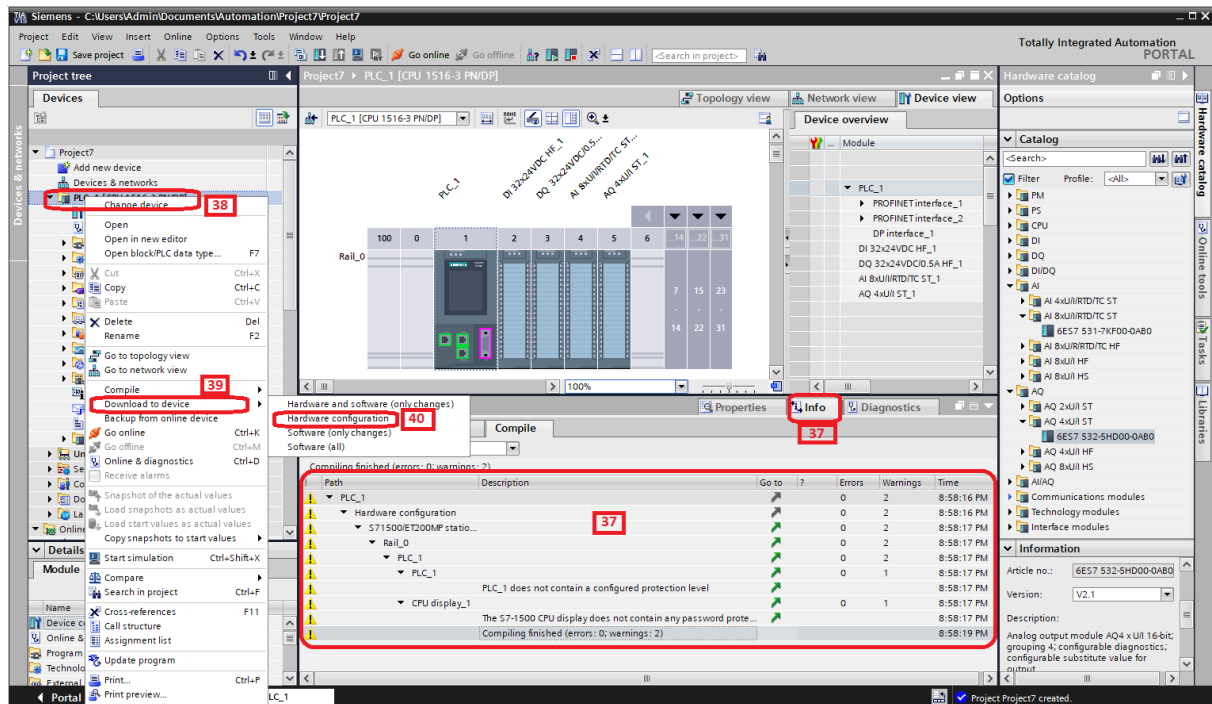


Rys. 8. Etapy konfiguracji stacji 3

## Katedra Mechaniki Stosowanej i Robotyki

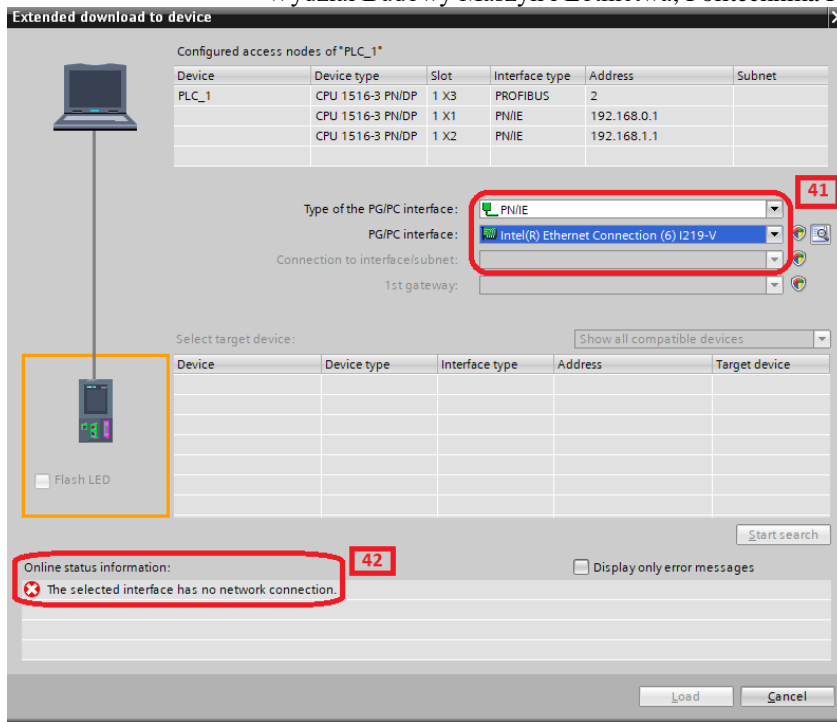
Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

Następnie klikamy prawym przyciskiem myszy na stację w drzewie projektu (38) rys. 9. Z rozwijanego menu wybieramy *Compile* (39) a następnie część projektu, którą chcemy skompilować *Hardware (only changes)*. W zakładce *Info* otrzymamy informację na temat przebiegu procesu kompilacji (37). Kompilacja przebiegła bez błędów, ostrzeżenia wynikają z braku ustawień związanych z funkcjami bezpieczeństwa. Kolejnym etapem jest wgranie skompilowanej konfiguracji sprzętowej do urządzenia. W tym celu klikamy prawym klawiszem myszy na stację w drzewie projektu (38), z rozwijającego menu wybieramy *Download to device* (39), a następnie element, który chcemy wgrać do stacji, w tym przypadku konfigurację sprzętową (*Hardware configuration*) (40).



Rys. 9. Wgranie konfiguracji sprzętowej do stacji

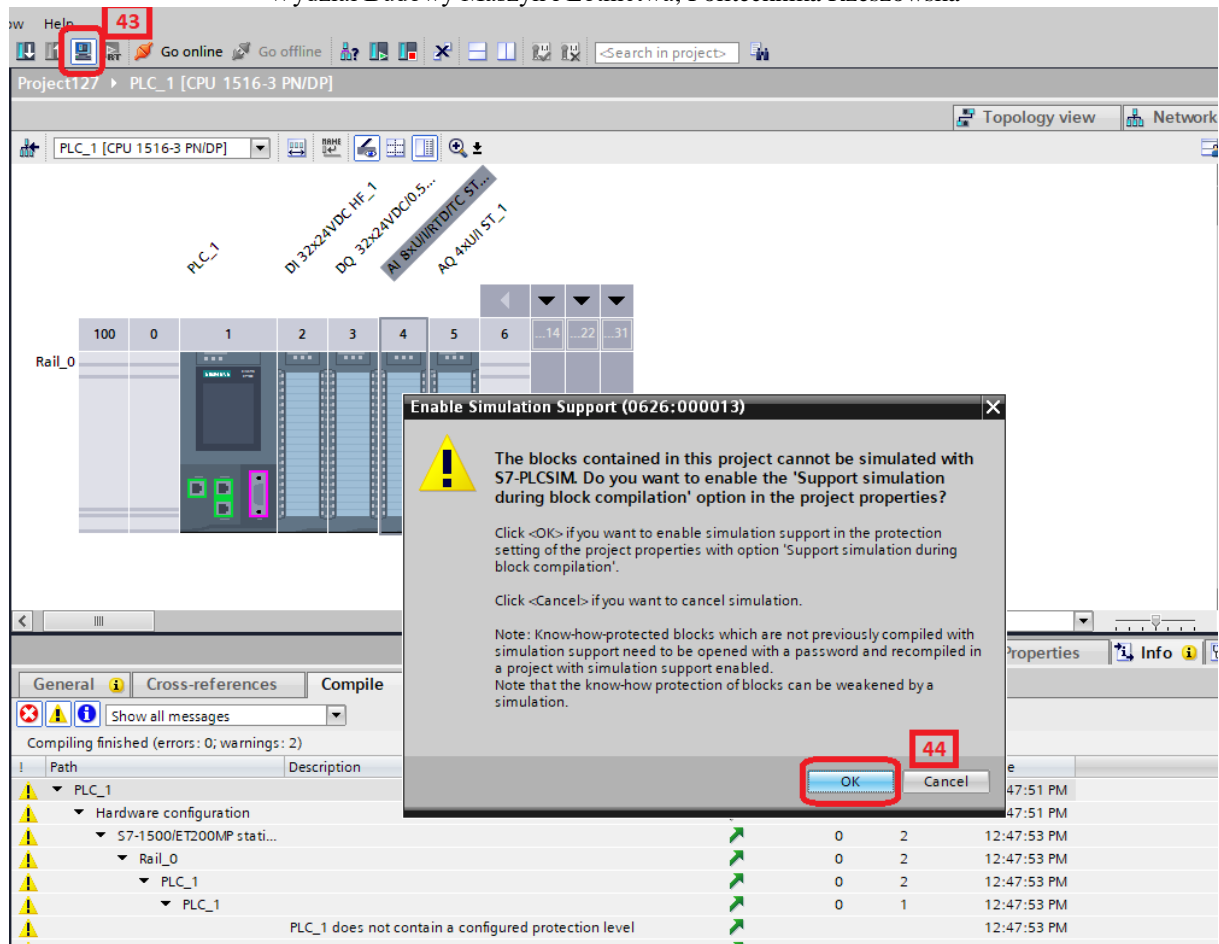
Realizacja powyższych kroków spowoduje otwarcie okna umożliwiającego wyszukanie dostępnych stacji z zastosowaniem wybranego interfejsu programowania (rys. 10). Wybieramy typ interfejsu komunikacyjnego (Profinet - PN/IE) oraz interfejs sprzętowy realizujący połączenie (**karta sieciowa TP Link**) (41). Jeżeli w chwili wyboru dany interfejs sprzętowy nie umożliwia połączenia z żadnym kompatybilnym urządzeniem, zostanie wyświetlony stosowany komunikat (42). Może się tak zdarzyć, jeżeli np. stanowisko z PLC nie jest podłączone do zasilania.



Rys. 10. Próba wyszukania dostępnych urządzeń z zastosowaniem interfejsu Profinet

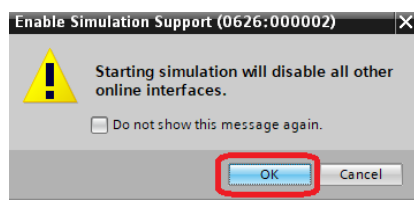
W celu symulacji działania programu posłużymy się modulem PLCSIM firmy Siemens, który umożliwia symulację działania sterownika S7-1500. Należy uruchomić symulator klikając na katalog stacji w drzewie projektu (ikona *Start simulation* stanie się aktywna) a następnie klikając ikonę *Start simulation* (43) rys. 11. Pojawi się okno informacyjne pokazane na rys. 11, należy kliknąć przycisk *OK* (44).





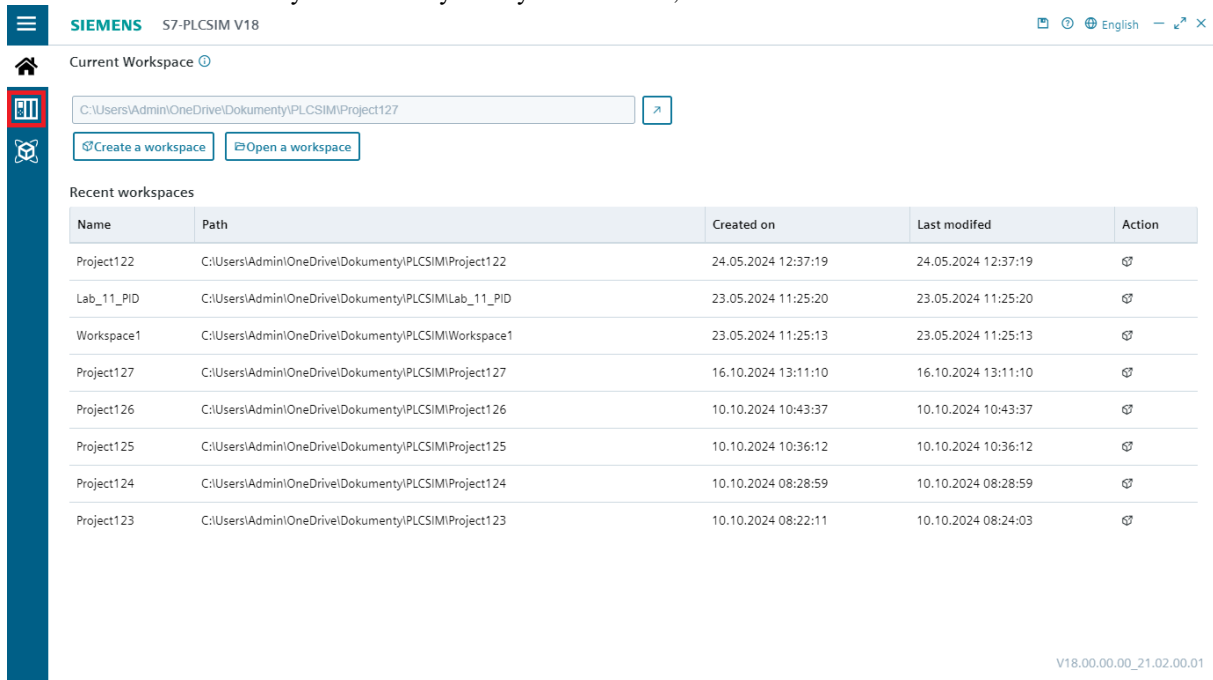
Rys. 11. Uruchomienie symulatora

Następnie zostanie wyświetlone okno informujące o zablokowaniu innych interfejsów komunikacji w przypadku włączenia symulatora (akceptujemy klikając OK).



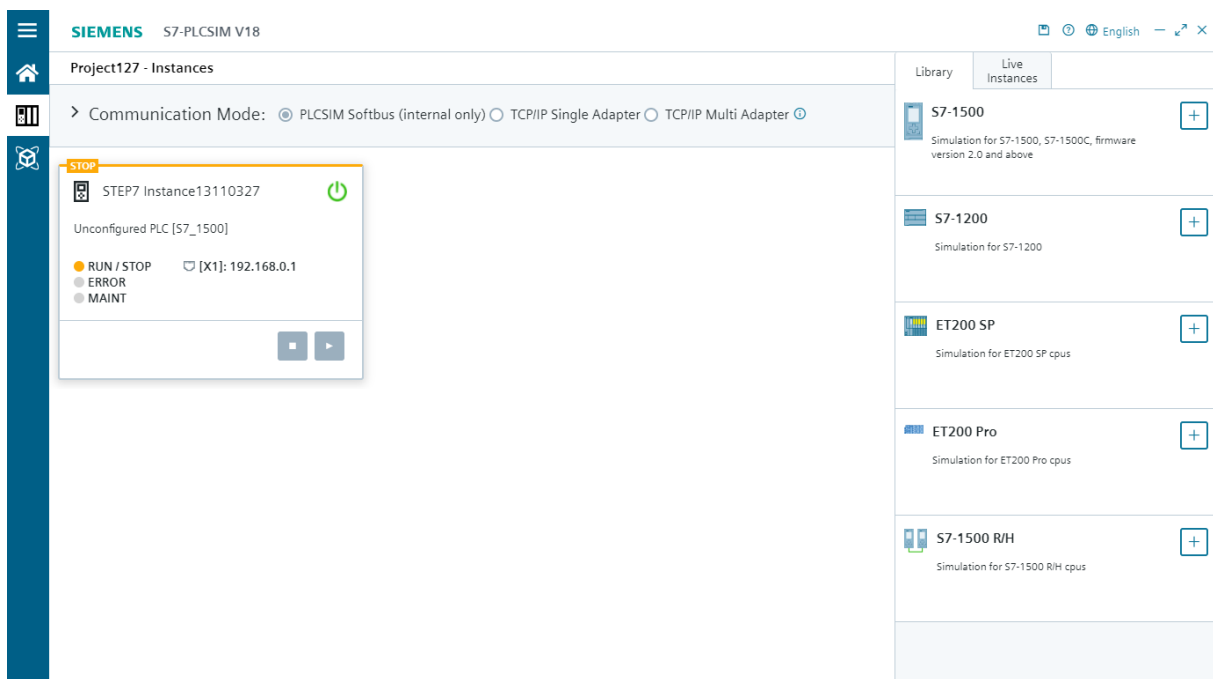
Rys. 12. Informacja o zablokowaniu innych interfejsów komunikacji

Po uruchomieniu symulatora pojawi się okno główne programu rys. 13. Należy wybrać ikonę sterownika na pionowym pasku po lewej stronie okna.



Rys. 13. Okno główne programu PLCSIM

Po wykonaniu powyższych czynności okno programu przyjmie formę pokazaną na rys. 14, gdzie symulowany sterownik jest zasilany i znajdują się w trybie STOP, o czym informuje świecąca na pomarańczowo dioda sygnalizacyjna.



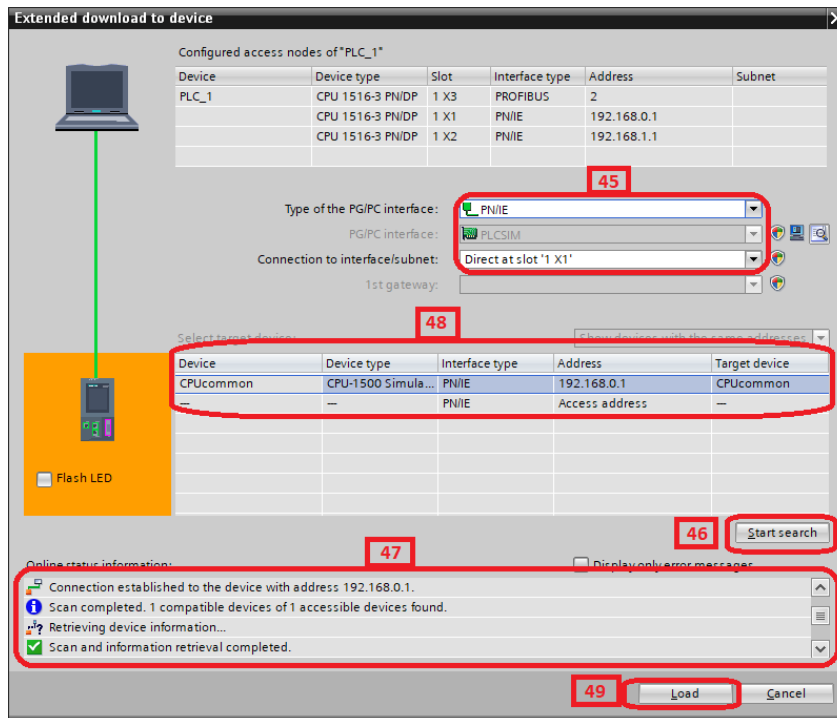
Rys. 14. Okno główne programu PLCSIM, sterownik S7-1500 włączony

Powtarzamy kroki (38),(39),(40), ponownie pojawi się okno wyboru urządzenia do zaprogramowania rys. 15. Tym razem po wyborze interfejsu programowania Profinet, domyślnie mamy wybrany symulator jako interfejs oraz gniazdo 1 interfejsu sieciowego jednostki CPU do programowania urządzenia (45). Klikamy przycisk *Start search* (46), tym razem wyszukiwanie kończy się powodzeniem (47), z dostępnych urządzeń wybieramy

## Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

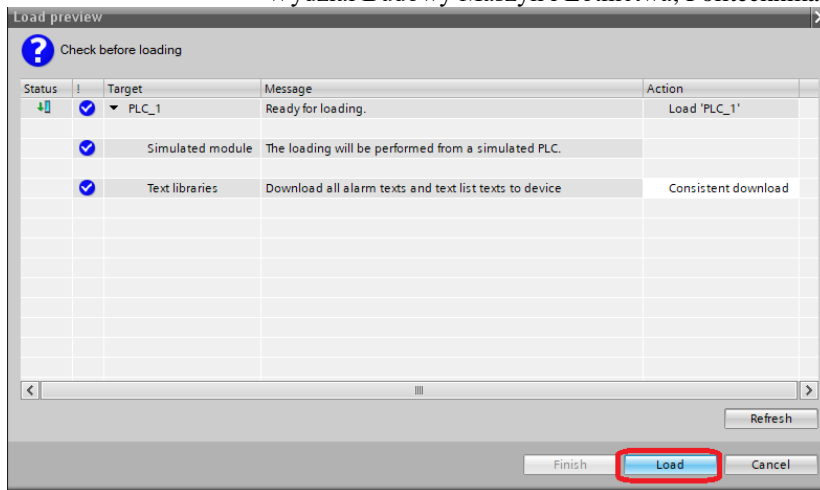
symulowany moduł S7-1500 (48) i zatwierdzamy ładowanie konfiguracji sprzętowej *Load* (49).



Rys.15. Okno wyszukiwania urządzenia do zaprogramowania

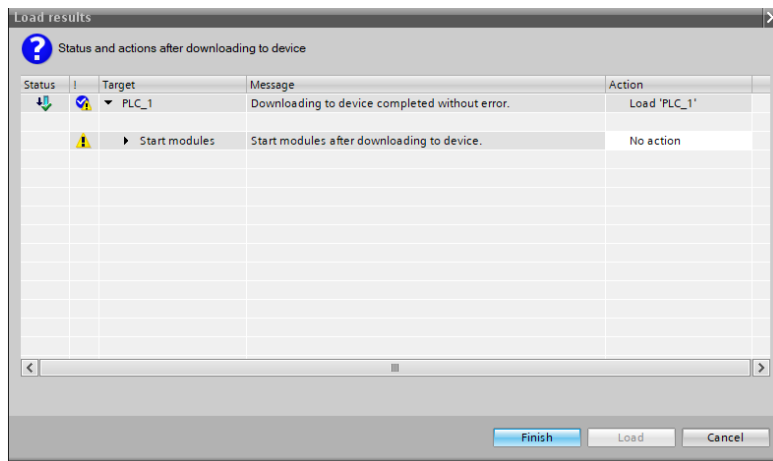
Zostanie wyświetlone okno rys. 16, na którym należy kliknąć *Connect*, a następnie w kolejnym oknie wybrać *Load*.





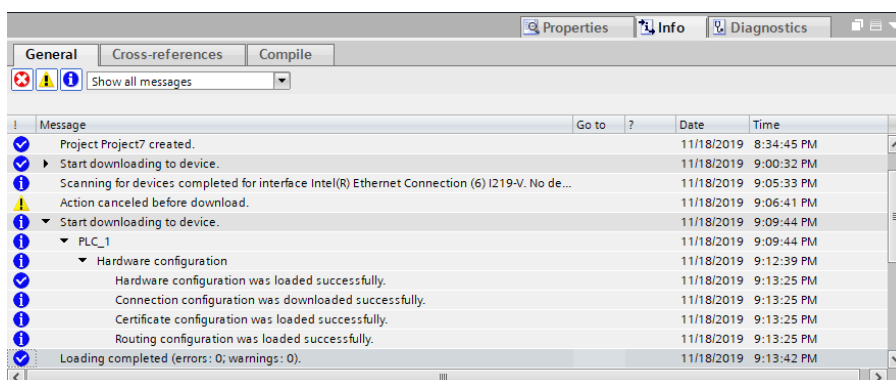
Rys. 16. Okno informacyjne

Pojawi się okno rys. 17 informujące o powodzeniu operacji zapisu konfiguracji sprzętowej do jednostki CPU. Akceptujemy klikając *Finish*.



Rys. 17. Okno informacyjne o powodzeniu zapisu konfiguracji sprzętowej do urządzenia

W dolnej części okna głównego programu, zakładce Info pojawia się informacja o poprawnym przebiegu zapisu konfiguracji sprzętowej rys. 18.



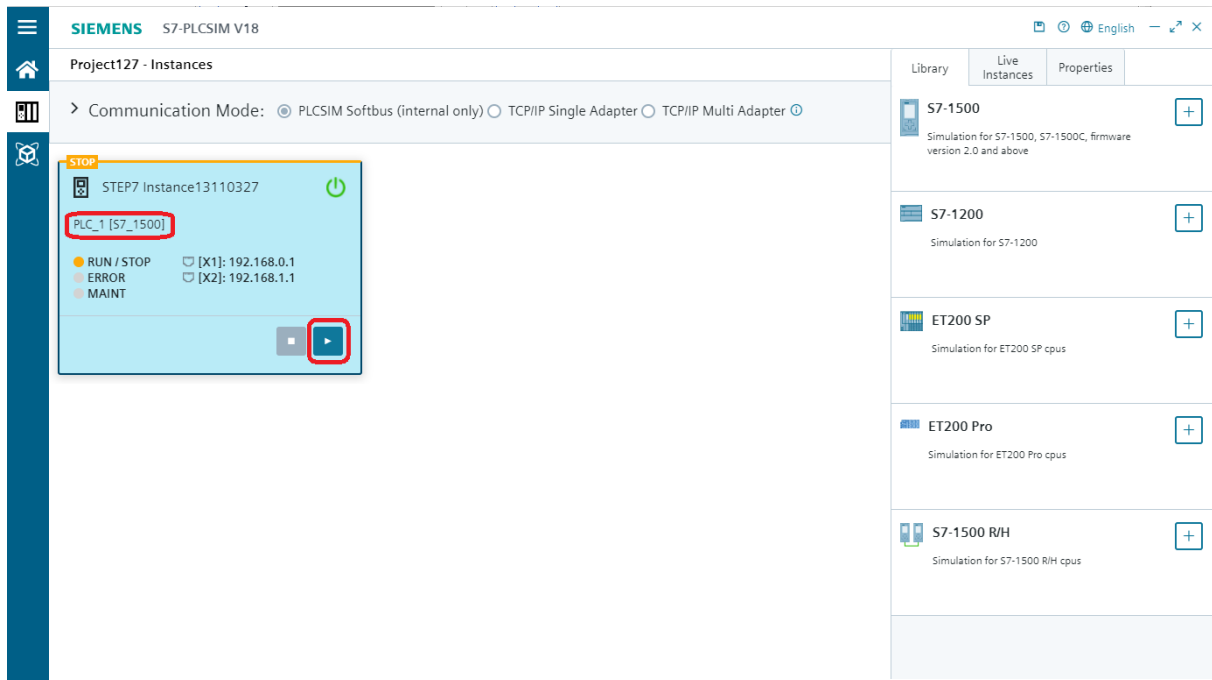
Rys. 18. Informacja o poprawnym przebiegu zapisu konfiguracji sprzętowej

Okno programu symulatora zmieni swój wygląd, uwzględniając wgraną konfigurację sprzętową, gdzie wyświetlana jest nazwa stacji. Przycisk zmiany trybu pracy ze STOP na RUN również zmieni swój wygląd, ale ponieważ nie wgraliśmy żadnego programu (tylko

## Katedra Mechaniki Stosowanej i Robotyki

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

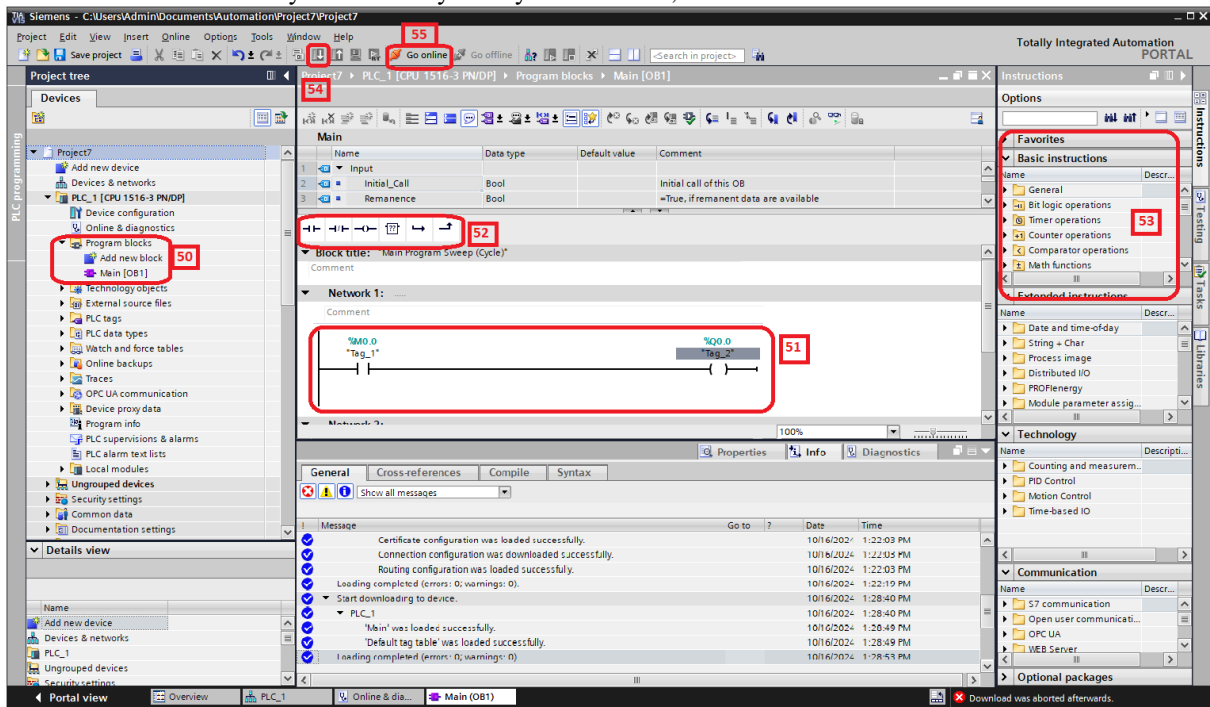
konfigurację sprzętową), nie możemy przełączyć symulatora sterownika w tryb pracy RUN (rys. 19).



Rys. 19. Okno symulatora CPU

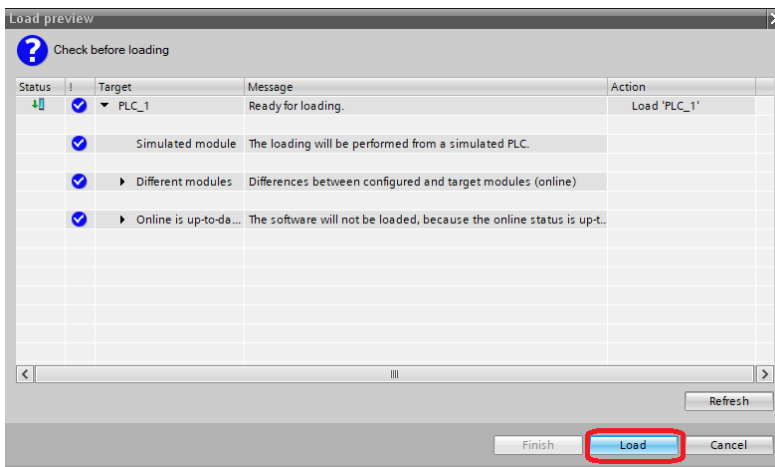
W kolejny etapie można przejść do edycji programu. Rozwijając z drzewa projektu gałąź *Program blocks*, mamy dostępny do programowania blok OB1 (50). Jest to główny blok każdego programu, wykonywany cyklicznie w sposób ciągły. Zawiera on główną część programu użytkownika. Dwukrotne kliknięcie bloku OB1 spowoduje otwarcie edytora LAD, umożliwiającego programowanie bloku. Programowanie odbywa się w sposób graficzny poprzez przenoszenie elementów dyskretnych najczęściej używanych (52) lub wszystkich z biblioteki (53) na tzw. sieć (51). W ten sposób powstaje program użytkownika. Następnie należy wszystkie zaprogramowane bloki zgrać do sterownika (54). Gdy sterownik jest w trybie RUN, można podglądać wyniki jego działania klikając *Go-online* (55).

**Katedra Mechaniki Stosowanej i Robotyki**  
Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska



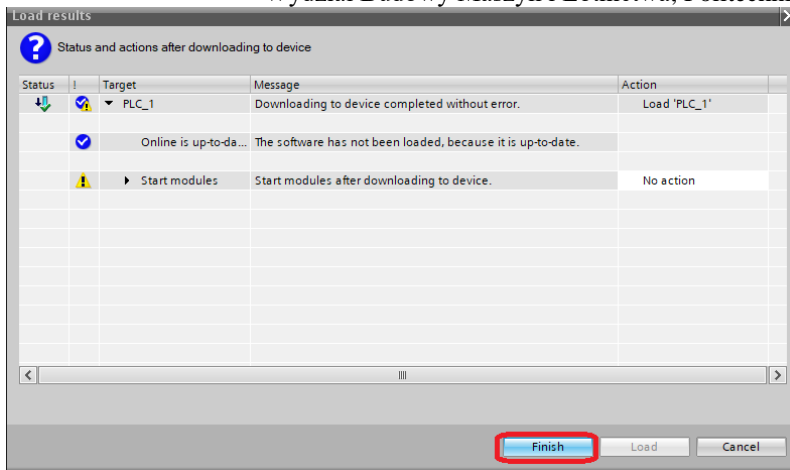
Rys. 20. Główne okno edytora LAD

Po zakończeniu edycji programu należy go wgrać do CPU, wybierając (54). Spowoduje to wyświetlenie okna rys. 21. Należy kliknąć *Load*.



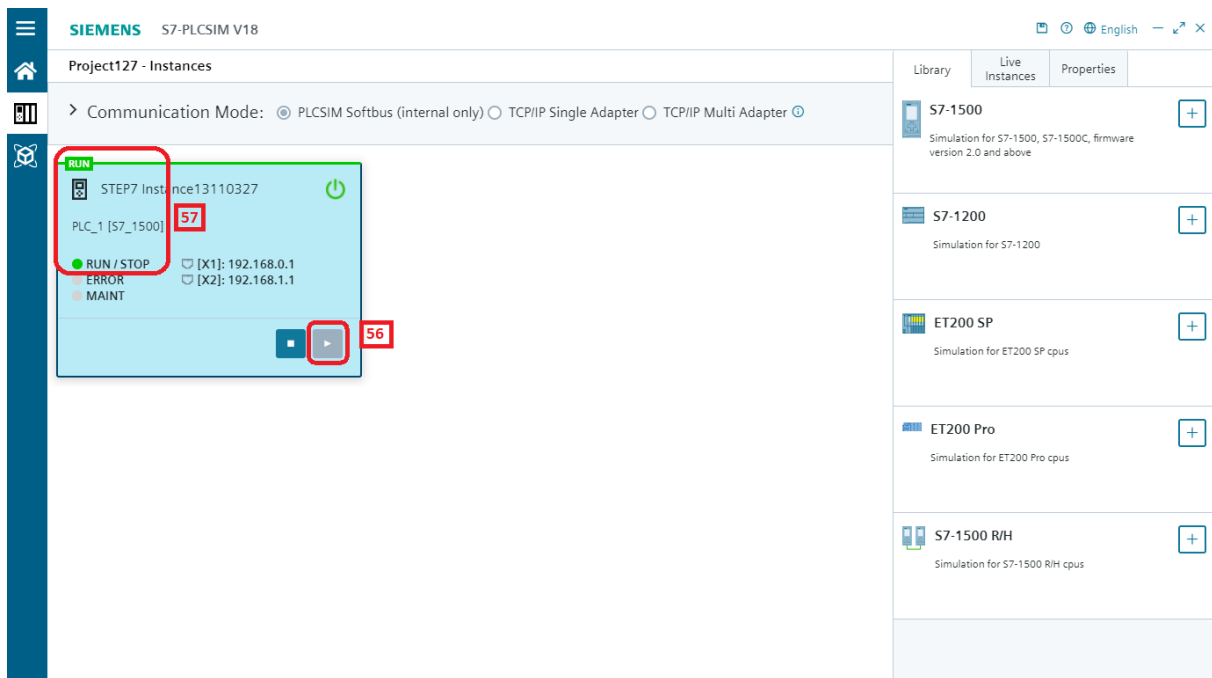
Rys. 21. Okno wgrzywania programu do CPU

Pojawi się okno informujące o wyniku procesu wgrzywania programu, rys. 22. Wybieramy *Finish*.



Rys. 22. Okno wgrzywania programu do CPU2

Teraz możemy przełączyć symulator PLC w tryb realizacji programu (RUN) wciskając przycisk (56) (po załączeniu zmieni się wizualizacja przycisku), rys. 23. PLC przejdzie w tryb przetwarzania programu, co zostanie zasygnalizowane przez zmianę barwy diody trybu pracy na kolor zielony (57).

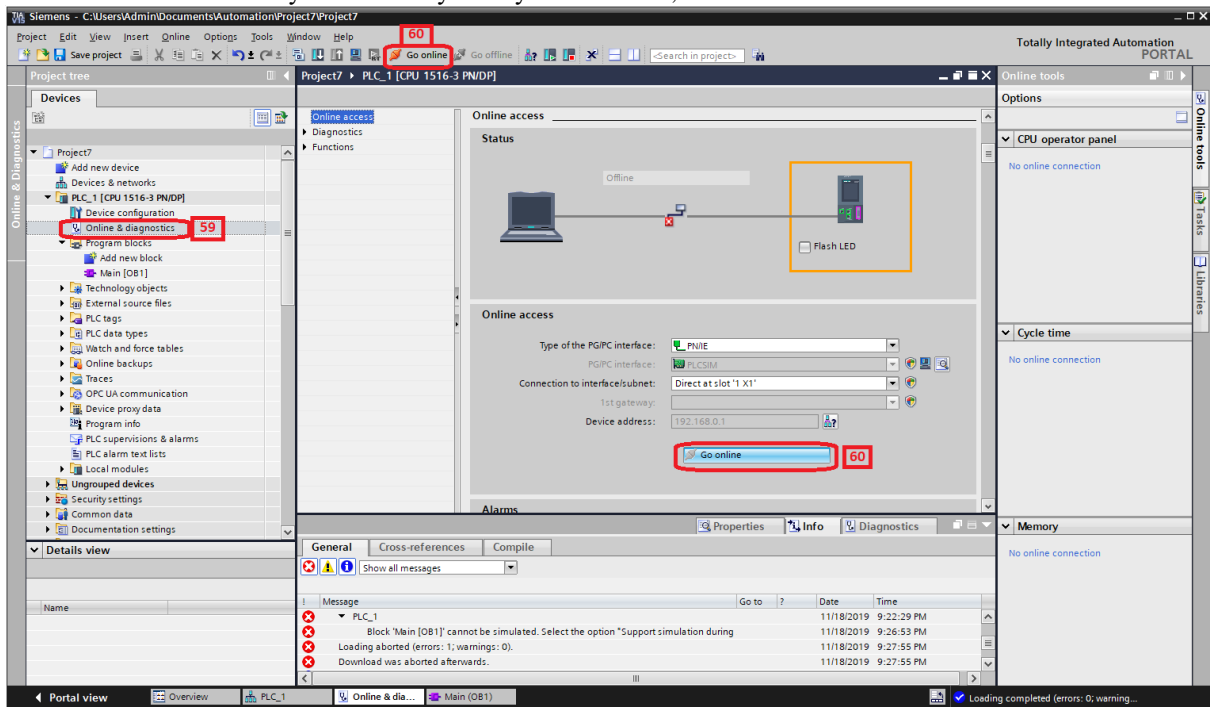


Rys. 23. Jednostka CPU symulatora w trybie RUN

Działanie programu można podglądać przechodząc w tryb pracy on-line (60) z okna *Online and Diagnostics* (59).

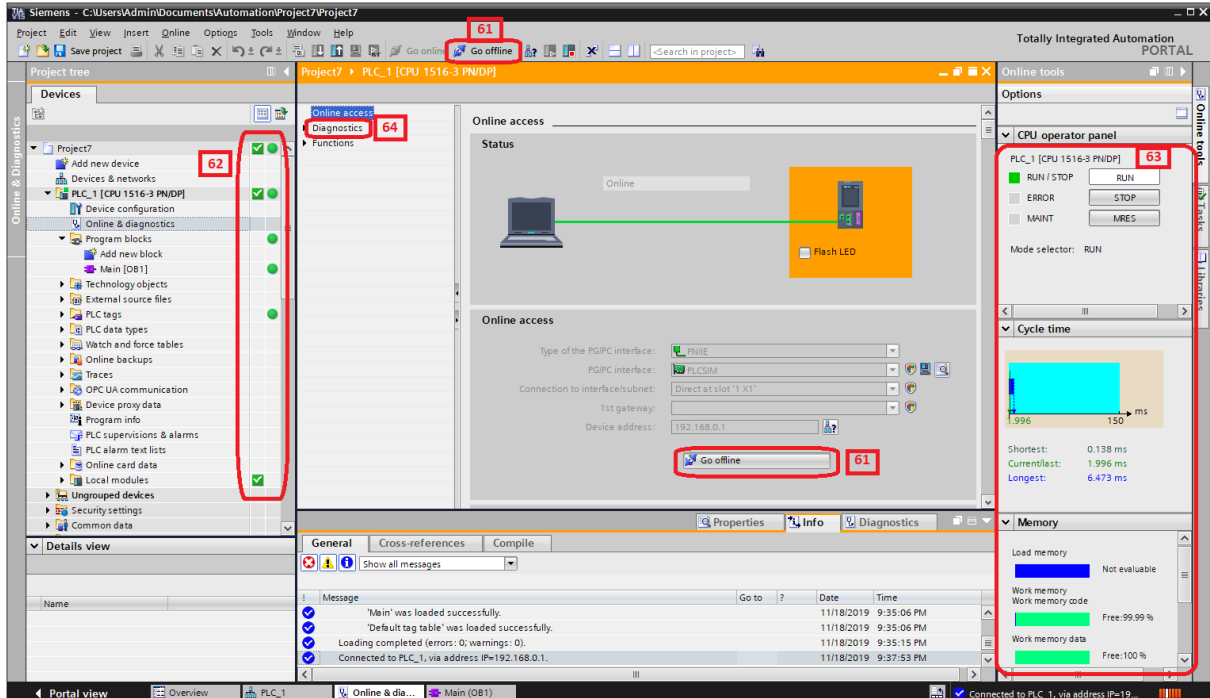
# Katedra Mechaniki Stosowanej i Robotyki

## Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska



Rys. 24. Włączenie trybu pracy on-line

Włączenie trybu on-line powoduje zmianę wyglądu okna programu rys.25. Zakładka *On-line tools* udostępnia dużo informacji na temat kwestii związanych z czasem realizacji cyklu pracy CPU (63), wykorzystaniem pamięci itd. Bardzo dużo ważnych informacji dostępnych jest w zakładce *Diagnostics* (64).



Rys. 25. Główne okno programu, sterownik w trybie on-line.

Przejsięcie do edytora LAD (otwarcie bloku OB1) (65) powoduje wyświetlenie okna programu jak na rys. 26. Włączenie monitoringu (66) powoduje zmianę wyglądu poszczególnych sieci programu (67). Możemy śledzić przepływ sygnału RLO (zielony) od szyny prądowej po

### Programowalne systemy mechatroniki

#### Lab. 2.3. Podstawy programowania sterowników PLC – operacje na bitach

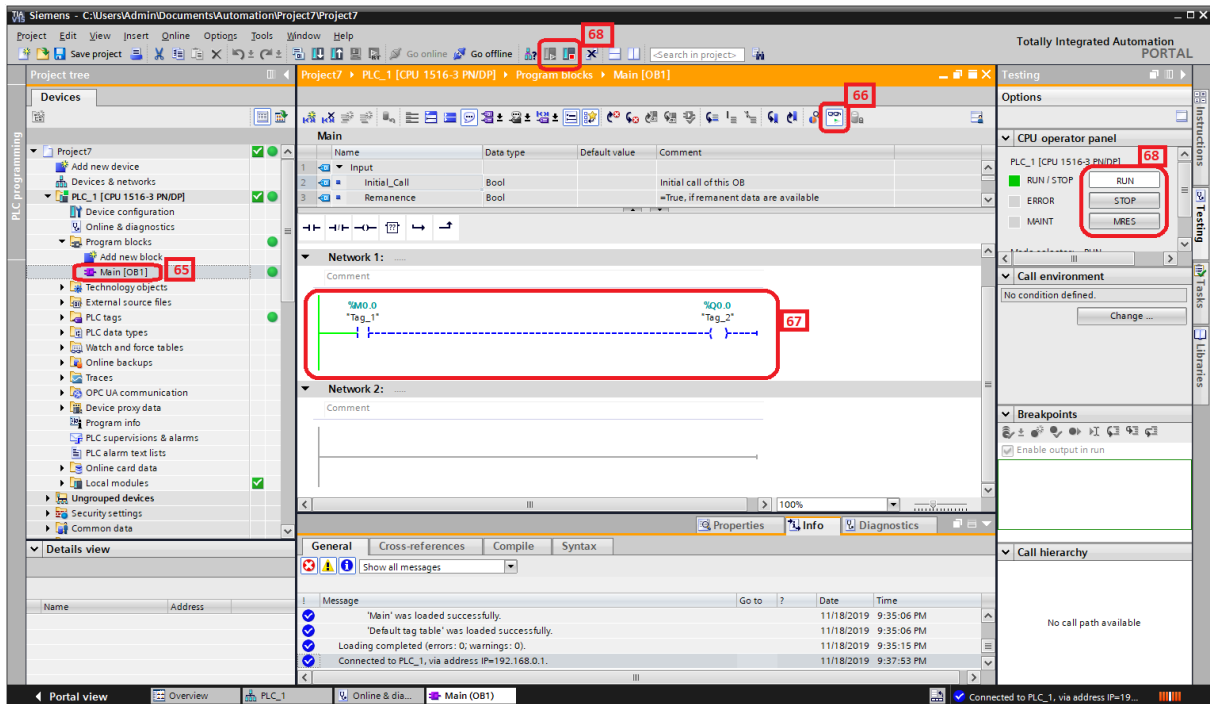
Opracował Marcin Szuster



## Katedra Mechaniki Stosowanej i Robotyki

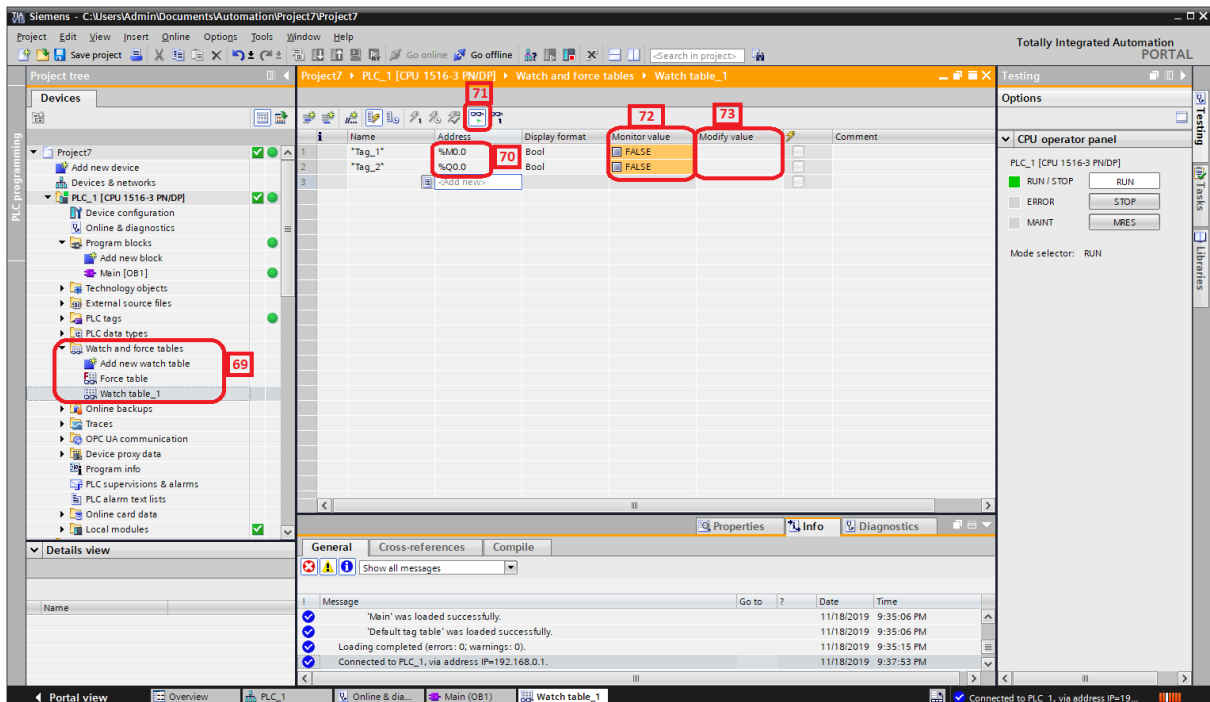
Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

lewej, w prawo poprzez poszczególne bloki. Z poziomu tego okna mamy też możliwość zatrzymania CPU lub wznowienia realizacji programu (68).



Rys. 26. Główne okno programu, edytora LAD on-line.

Śledzenie i wprowadzanie zmian w pamięci CPU umożliwia *Tablica watch* (69). Należy utworzyć nową tablicę i uzupełnić adresy śledzonych zmiennych (70). Włączenie trybu monitorowania w sytuacji, gdy CPU jest w trybie *on-line*, powoduje wyświetlanie informacji o zmiennych na bieżąco (72) oraz modyfikację ich stanu (73).



Rys. 27. Edycja tablicy Watch

## Programowalne systemy mechatroniki

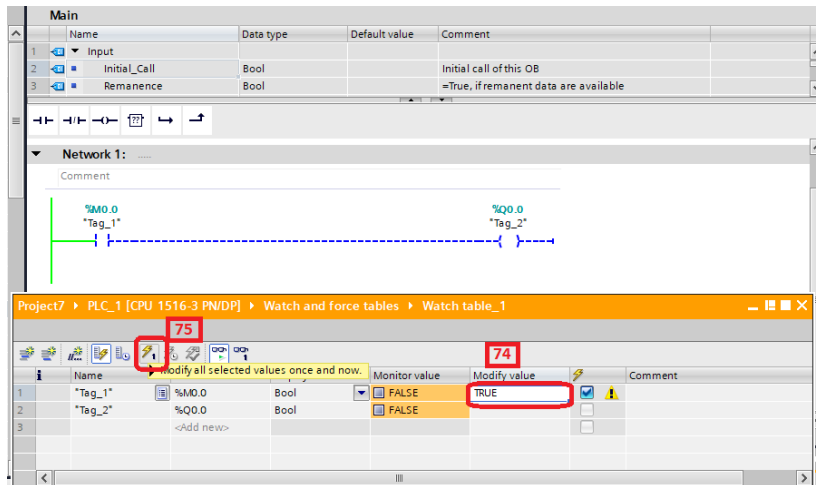
Lab. 2.3. Podstawy programowania sterowników PLC – operacje na bitach

Opracował Marcin Szuster

## Katedra Mechaniki Stosowanej i Robotyki

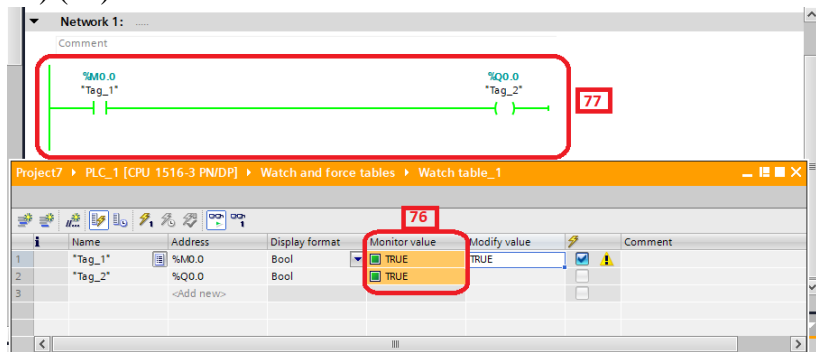
Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

Nadpisanie bitu M0.0 wartością 1 (74) oraz wprowadzenie zmian do pamięci CPU (75) powoduje zmianę stanu sieci z rys. 28 do rys. 29.



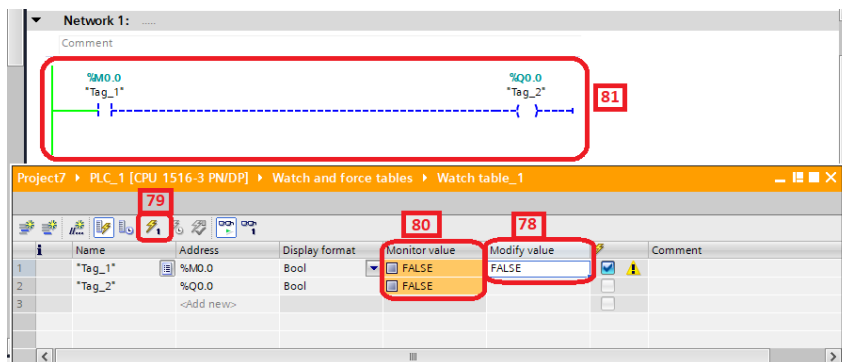
Rys. 28. Edycja zmiennych w tabeli Watch 1

Styk normalnie otwarty zaczyna przewodzić RLO i sygnał ten dociera do cewki Q0.0 (rys. 29) (76).



Rys. 29. Edycja zmiennych w tabeli Watch 2

Ponowna zmiana wartości bitu M0.0 na wartość 0 powoduje zmianę stanu bitu Q0.0 (78), (79), (80).



Rys. 30. Edycja zmiennych w tabeli Watch 3

## 2. Operacje na bitach

Podstawowe elementy biblioteki *Bit logic operations* to:



styk normalnie otwarty (*Normally Open Contact*) – element wejściowy, w spoczynku nie przepuszcza sygnału, jeśli jego adres zmieni stan na wysoki, to zacznie przepuszczać sygnał, gdy adres się wyzeruje, to ponownie przestanie przepuszczać sygnał.



styk normalnie zamknięty (*Normally Closed Contact*) – element wejściowy, w spoczynku przepuszcza sygnał, jeśli jego adres zmieni stan na wysoki, to przestanie przepuszczać sygnał, gdy adres się wyzeruje, to ponownie zacznie przepuszczać sygnał.



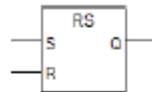
cewka (*Coil*) – element wyjściowy, jeśli dotrze do niej sygnał, to jej adres zmieni stan na wysoki, jeśli sygnał zaniknie, to jej adres się wyzeruje.



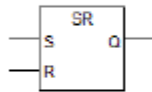
cewka zerująca (*Reset*) - element wyjściowy, jeśli dotrze do niej sygnał (impuls), to jej adres zmieni stan na niski i taki pozostanie.



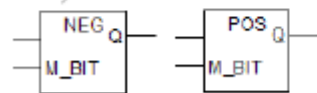
cewka ustawiająca (*Set*) - element wyjściowy, jeśli dotrze do niej sygnał (impuls), to jej adres zmieni stan na wysoki i taki pozostanie.



przerzutnik RS (*Reset Set Flip Flop*) – jest zerowany jeśli na wejściu R występuje stan „1”, oraz na wejściu S jest „0”. W odwrotnej sytuacji przerzutnik jest ustawiany w stan „1”.



przerzutnik SR (*Set Reset Flip Flop*) – jest ustawiany jeśli na wejściu S występuje stan „1”, oraz na wejściu R jest „0”. W odwrotnej sytuacji przerzutnik jest zerowany.



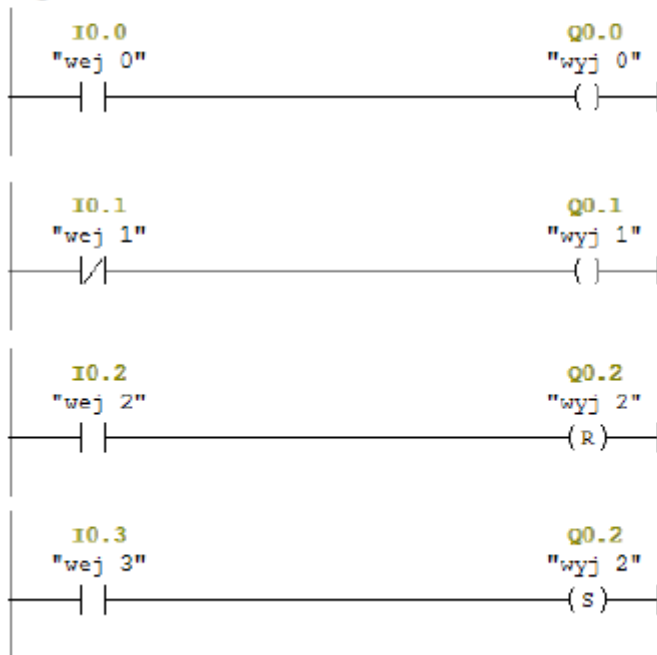
detektory zboczy (*Address Negative Edge Detection, Address Positive Edge Detection*) – wykrywa zbocze opadające (*NEG*) lub narastające (*POS*). Poprzedni stan przechowuje pod adresem przypisanym do wejścia *M\_BIT*.

Poniżej przedstawiono przykłady zastosowania poszczególnych elementów biblioteki *Bit logic operations*. **Zamiast bitów I0.0-I0.3 należy użyć bitów M0.0-M0.3.**

Tabela symboli niezbędna do poniższych przykładowych programów:

Symbol	Address	Data type
wej 0	I 0.0	BOOL
wej 1	I 0.1	BOOL
wej 2	I 0.2	BOOL
wej 3	I 0.3	BOOL
wyj 0	Q 0.0	BOOL
wyj 1	Q 0.1	BOOL
wyj 2	Q 0.2	BOOL

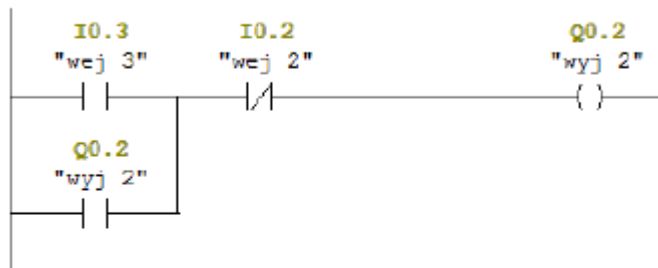
Program:



Dwie ostatnie linie programu tworzą przerzutnik RS.

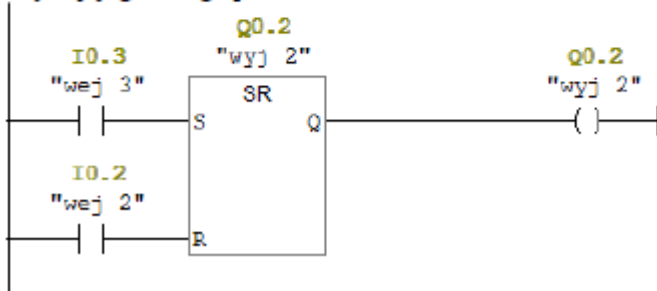
Przerzutnik taki można zrealizować na wiele innych sposobów.

Nie używając cewek R i S.



Styk normalnie otwarty w odgałęzieniu linii służy do „pamiętania”.

Używając gotowego przerzutnika.



## Katedra Mechaniki Stosowanej i Robotyki

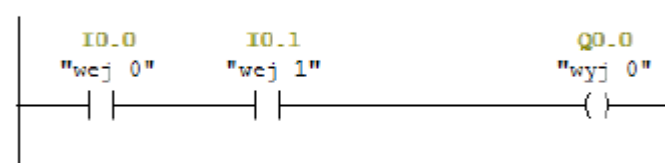
Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

Przy pomocy tych prostych operacji na bitach możemy również zrealizować także funkcje logiczne.

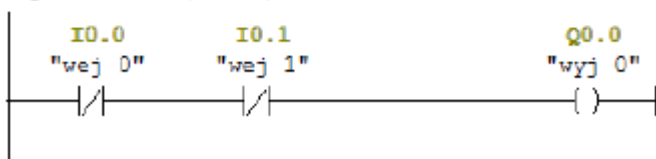
Logiczne „nie” (*NOT*), oznaczenie  $\bar{a}$



Logiczne „i” (*AND*), oznaczenie  $ab$



Logiczne „nie i” (*NAND*), oznaczenie  $\overline{ab}$



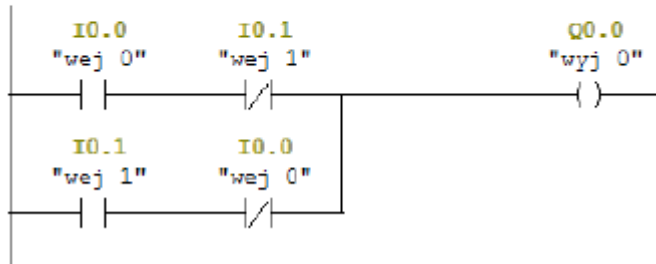
Logiczne „lub” (*OR*), oznaczenie  $a + b$



Logiczne „nie lub” (*NOR*), oznaczenie  $\overline{a + b}$



Logiczne „tylko lub” (*XOR*), oznaczenie  $a \oplus b$



## **Katedra Mechaniki Stosowanej i Robotyki**

Wydział Budowy Maszyn i Lotnictwa, Politechnika Rzeszowska

### **3. Zadania do wykonania:**

1. Utworzyć konfigurację sprzętową stacji ze stanowiska dydaktycznego i wgrać ją do symulatora PLC.
2. Wykonać programy podane w przykładach i zbadać ich działanie, użyć symulatora PLC i tabeli *Watch*.

### **4. Sprawozdanie powinno zawierać:**

1. Wstęp teoretyczny.
2. Opis realizowanych zadań.
3. Listingi programów z komentarzem dotyczącym funkcji poszczególnych linii kodu.
4. Opis działania programów z ilustracją graficzną na podstawie działania symulatora.
5. Wnioski.