Ćwiczenia z S7-1200

Komunikacja S7-1200 z przyciskowym panelem HMI KP300 PN

FAQ · Marzec 2012

Przykłady i Aplikacje

www.siemens.pl/S7-1200

SIEMENS

Spis treści

1 Opis zagadnienia poruszanego w ćwiczeniu

W ćwiczeniu poruszony będzie temat stworzenia prostej aplikacji z wizualizacją na panelu przyciskowym KP300 PN. Zadaniem będzie napisanie aplikacji sterującej rozruchem silnika metodą gwiazda – trójkąt. W przykładzie będą przedstawione czynności programisty przy dodawaniu panelu HMI do projektu, skomunikowaniu go ze sterownikiem, a także wymianie danych pomiędzy stacją operatorską, a sterownikiem PLC za pośrednictwem sieci Profinet. Przedstawiona będzie też konfiguracja alarmów wywołanych stanami wysokimi na wejściach cyfrowych sterownika oraz wykorzystanie funkcjonalności zmiany podświetlenia panelu.

1.1 Wykaz urządzeń

Hardware

Lp.	Urządzenie	Ilość	Numer katalogowy
1.	Sterownik SIMATIC S7-1200, model CPU 1212C AC/DC/RLY	1	6ES7212-1BD30-0XB0
2.	Panel KP300 PN mono 3”	1	6AV6647-0AH11-3AX0
3.	Zasilacz PM 1207 (24V DC / 2,5A)	1	6EP1332-1SH71
4.	Switch Ethernet CSM 1277	1	6GK7277-1AA10-0AA0
5.	Kabel Ethernet 6m (komunikacja sterownika z panelem HMI oraz PG/PC)	3	6XV1870-3QH60

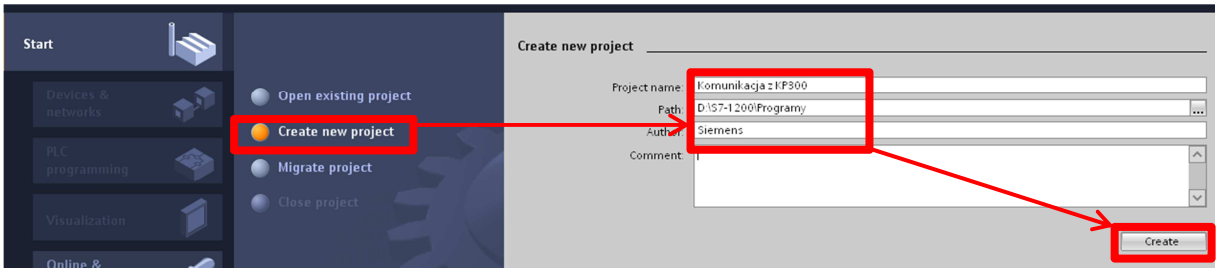
Software

Lp.	Nazwa	Ilość	Numer katalogowy
1.	Step 7 Basic v11	1	6ES7822-0AA01-0YA0

2 Konfiguracja S7-1200 PLC oraz HMI KP300 PN

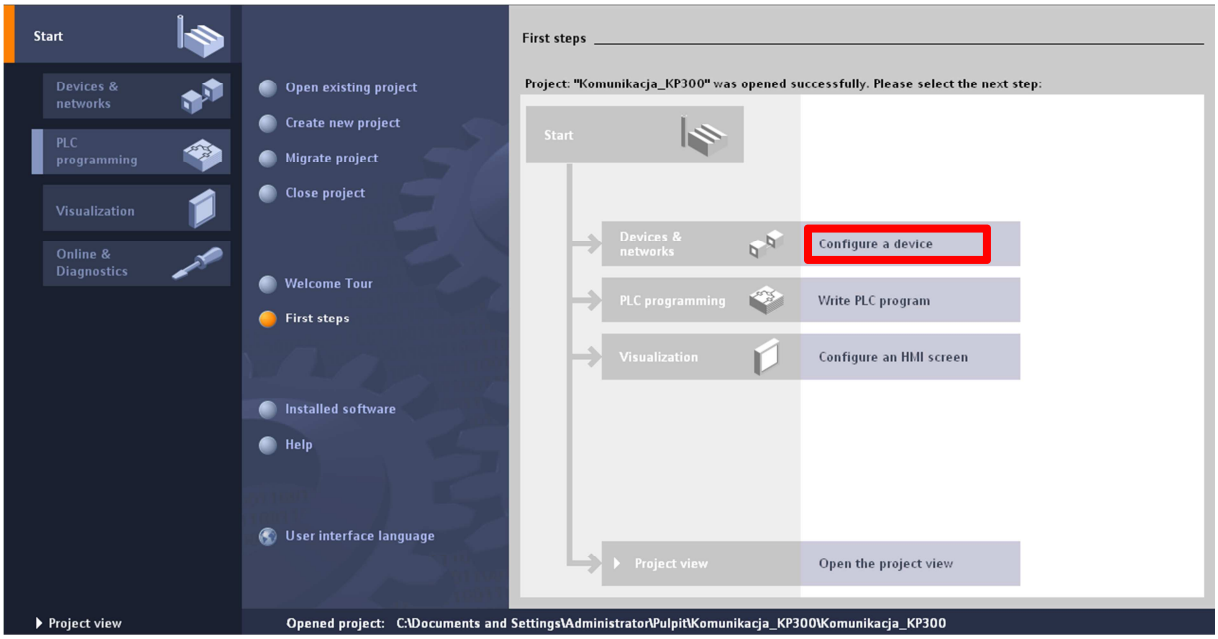
2.1 Nowy projekt

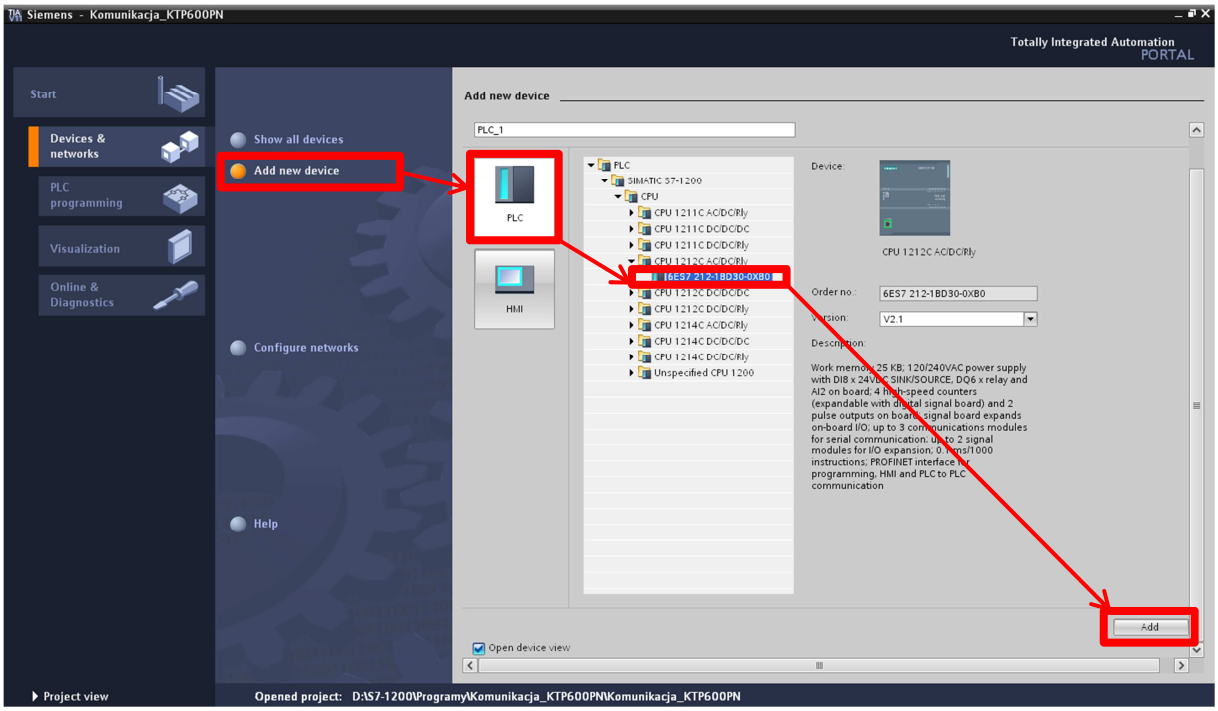
Podczas tworzenia nowego projektu, należy nadać mu nazwę, ścieżkę jego lokalizacji na dysku twardym komputera, opcjonalnie autora i komentarz, następnie zatwierdzić przyciskiem ***Create***.



2.2 Dodawanie CPU do projektu

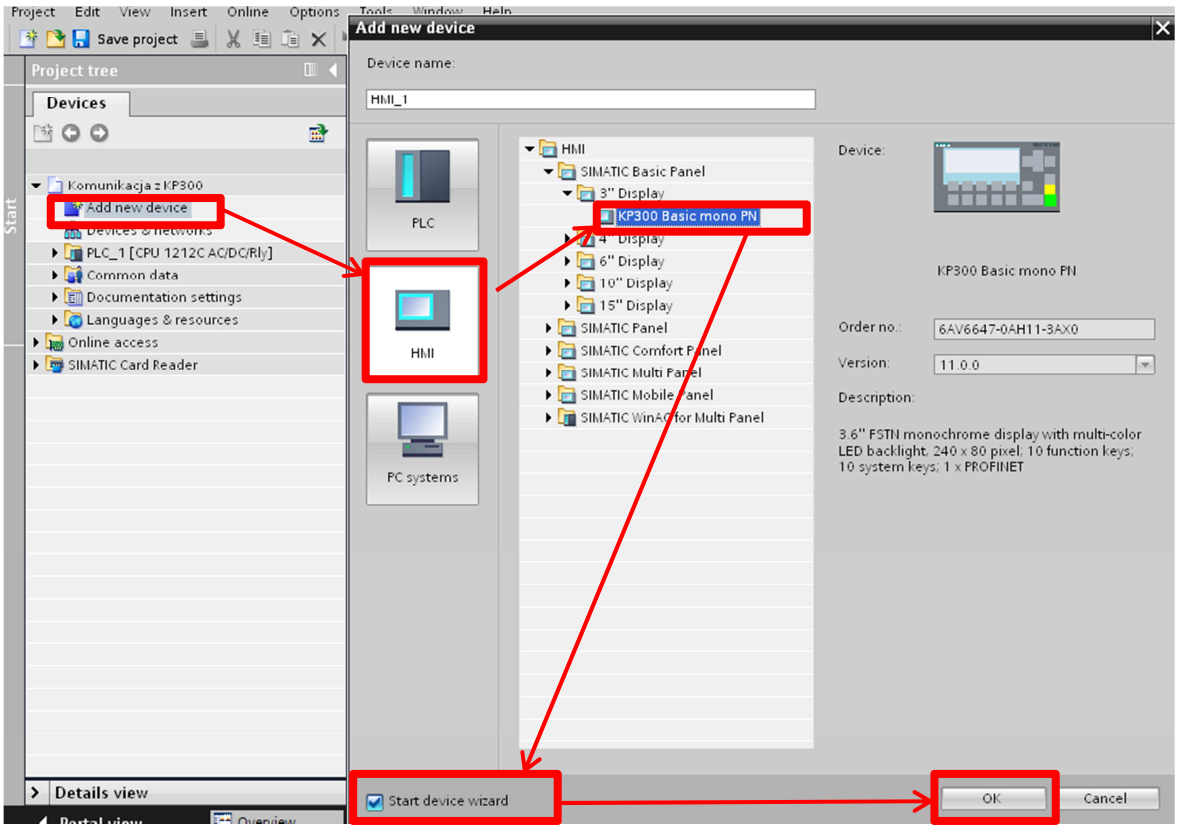
Przy dodawaniu nowego urządzenia w widoku *Portal view* trzeba wybrać opcję ***Configure a device***, następnie ***Add new device***, potem rodzaj urządzenia (w tym przypadku sterownik PLC) i model urządzenia. Po tym należy zatwierdzić konfigurację, klikając przycisk ***Add***.



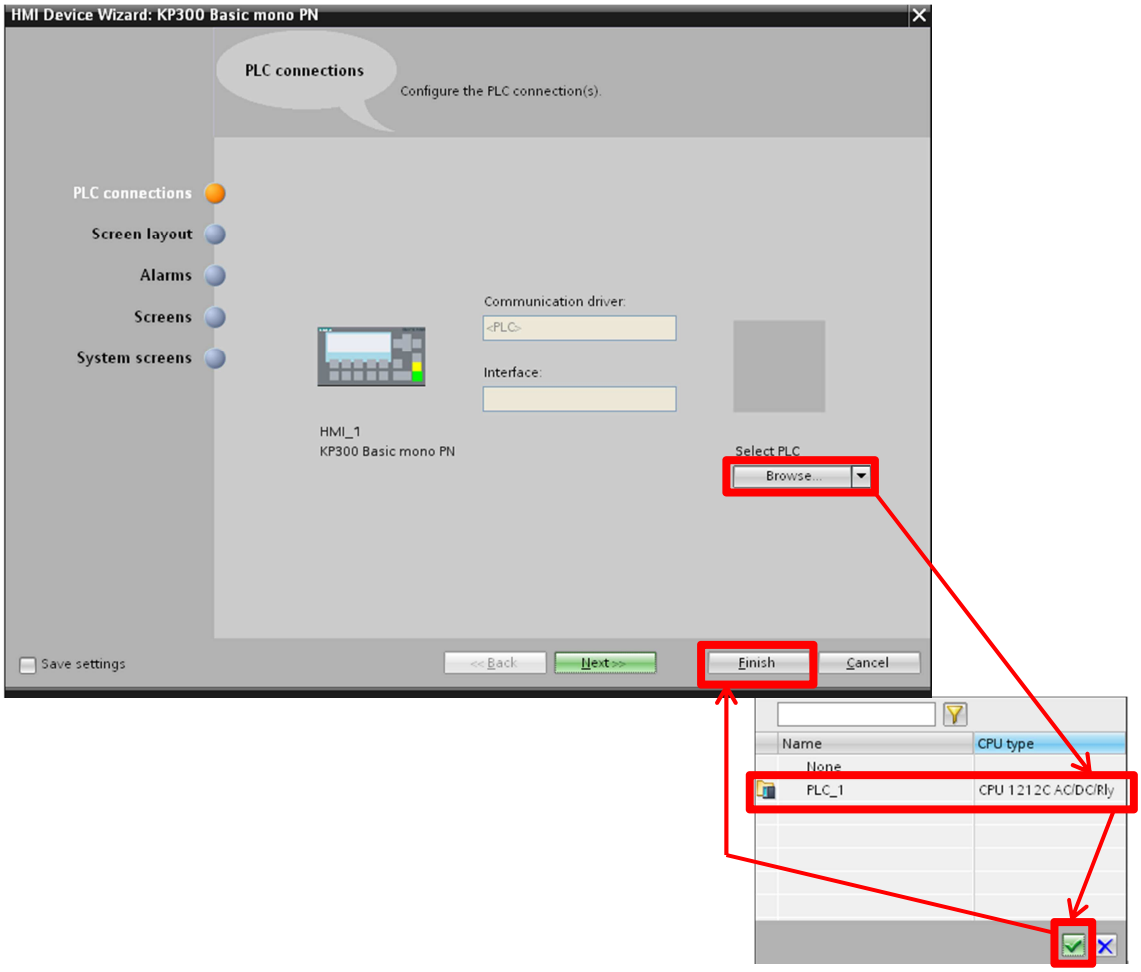


2.3 Dodawanie panelu HMI do projektu

Dodając panel HMI w widoku *Project view*, należy kliknąć dwa razy lewym przyciskiem myszy w polu ***Project tree*** na ***Add new device***, w nowym oknie dialogowym trzeba wybrać odpowiedni panel, następnie zaznaczyć opcję ***Start device wizard*** i potwierdzić klikając przycisk ***OK***.

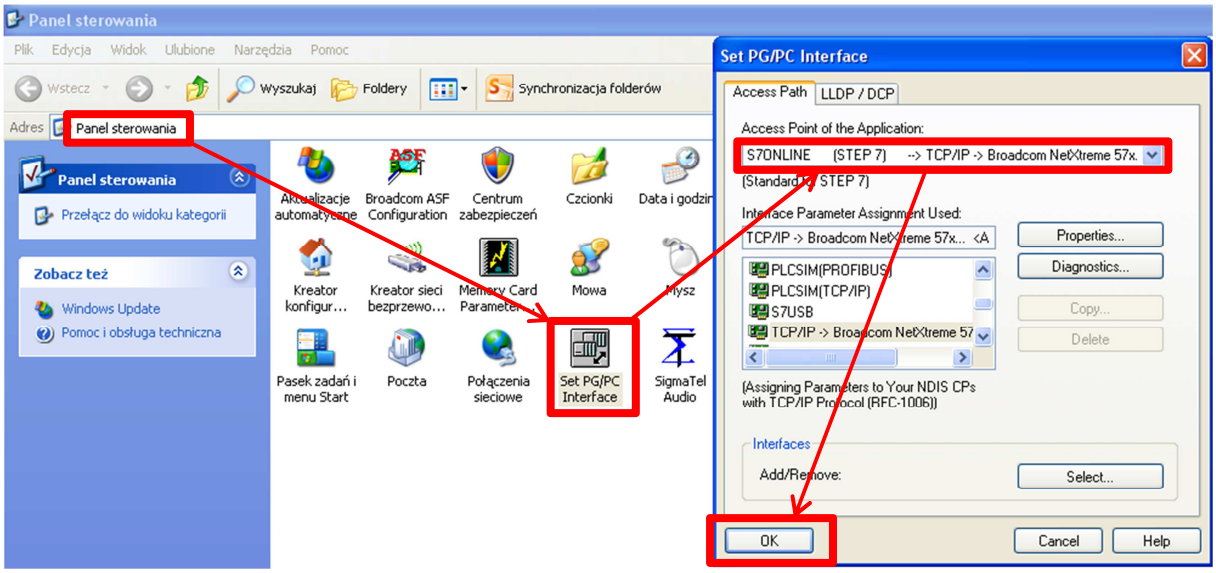


W oknie dialogowym ***HMI Device Wizard*** trzeba wybrać sterownik PLC, który będzie się z nim komunikował. Następnie kliknąć ***Finish,*** aby zakończyć konfigurację HMI lub ***Next***, aby kontynuować parametryzację ustawień ekranu.



2.4 Ustawienie połączenia w PG/PC Interface

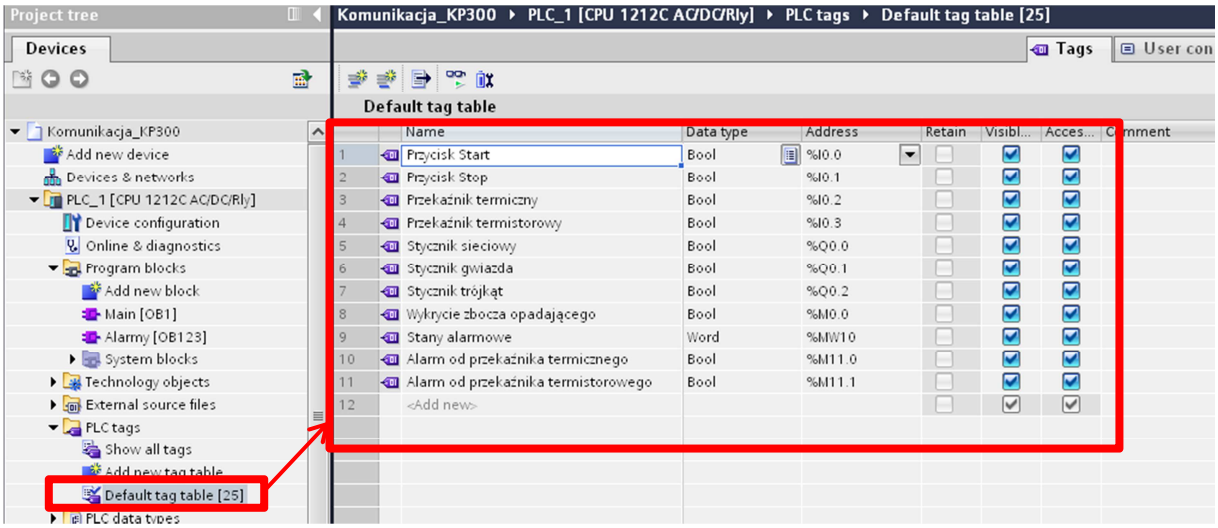
Ustawienie odpowiedniego połączenia w ***Set PG/PC Interface*** zapewnia komunikację między sterownikiem PLC a symulacją Runtime (RT).
Możliwe jest przeprowadzenie symulacji pracy HMI z PLC, jeżeli komputer jest połączony ze sterownikiem S7-1200, bez konieczności fizycznego posiadania panelu.
Konfigurację tę wprowadza się w ***Panelu sterowania.*** W tym celu należy kliknąć na ***Set PG/PC Interface***. W zakładce ***Access path*** jako punkt dostępu ***Access Point of the Aplication*** trzeba ustawić ***S7ONLINE (STEP 7) -> TCP/IP ->Karta sieciowa***.



3 Programowanie sterownika i panelu HMI

3.1 Konfiguracja zmiennych

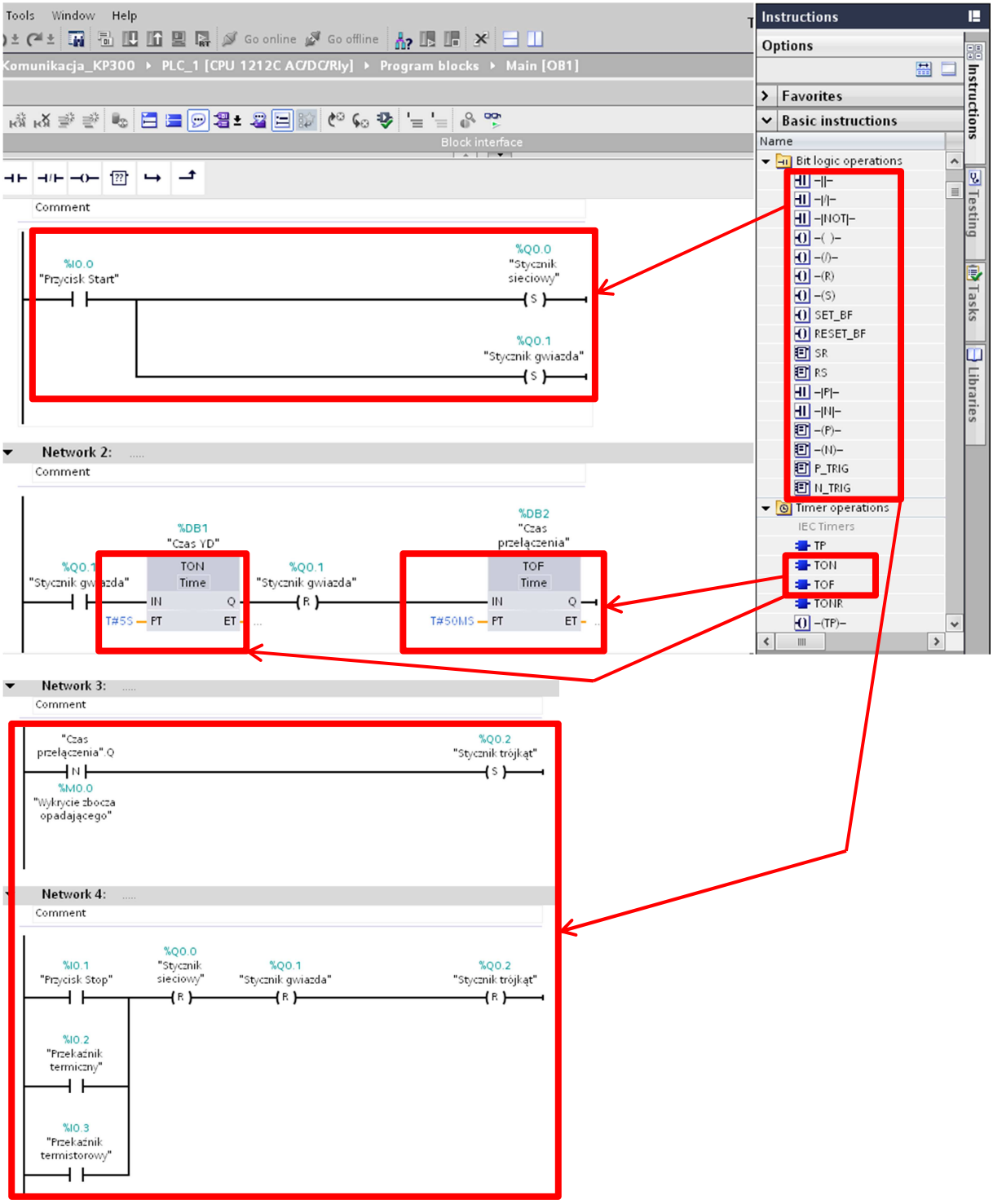
W ***Project tree*** rozwinąć gałąź ***PLC tags***, następnie wybrać ***Default tag table*** i utworzyć zmienne PLC. Ważne, żeby pamiętać o odpowiednim do typu danych ich zaadresowaniu (zmienne typu Bool muszą mieć adresy bitowe, np. I0.0, zmienne Word muszą mieć adresy o wielkości słowa, np. MW10).

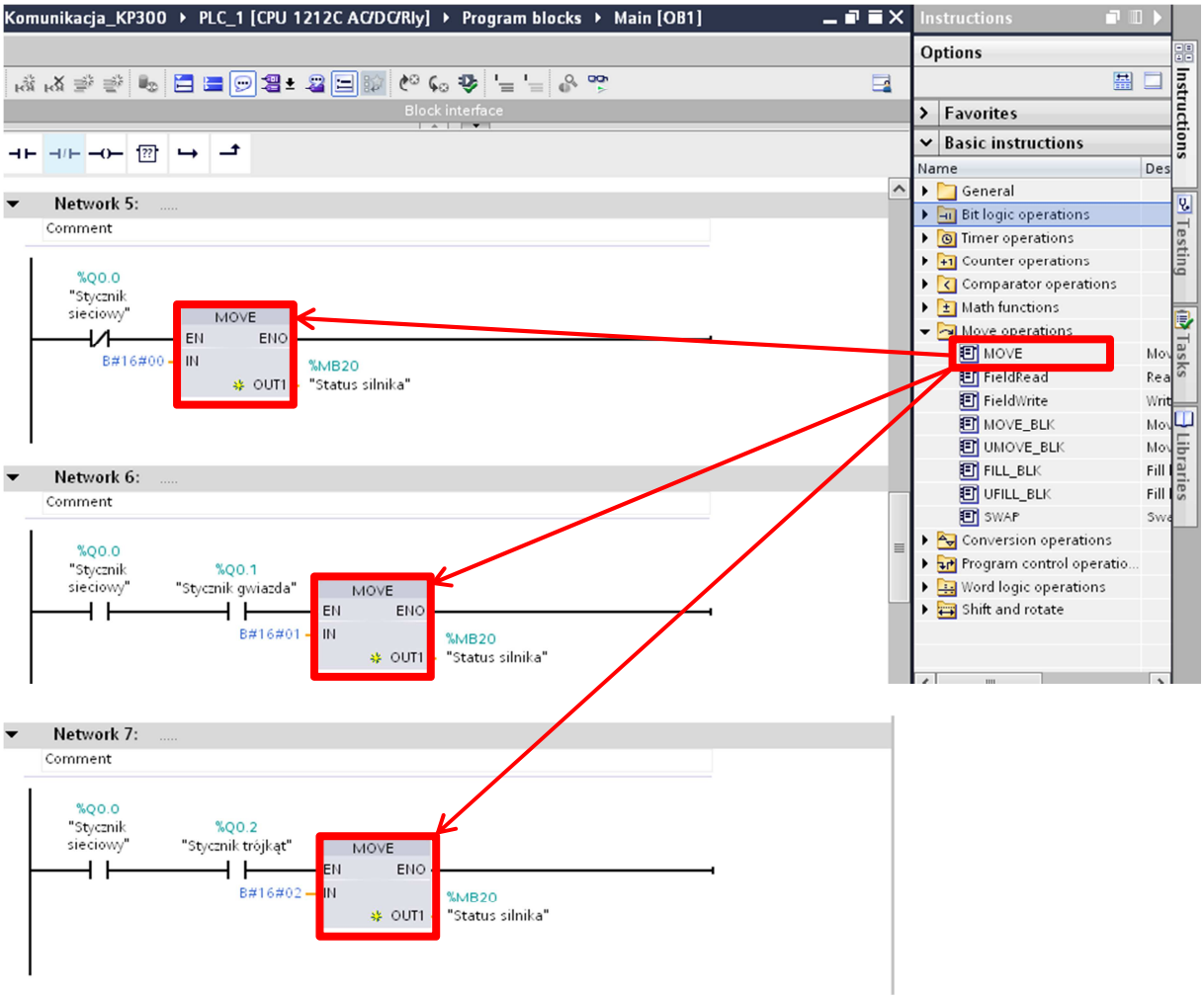


3.2 Program sterownika

Program sterownika składa się z dwóch bloków organizacyjnych wywoływanych cyklicznie. Blok organizacyjny OB1 (Main) jest głównym blokiem programu i zawiera algorytm sterowania silnikiem z rozruchem typu gwiazda-trójkąt. Blok organizacyjny OB123 (Alarmy) odpowiada za wyświetlanie alarmów na panelu w przypadku przeciążenia silnika lub zbyt wysokiej jego temperatury.

W bloku OB1 należy napisać algorytm sterowania jak na poniższym przykładzie.

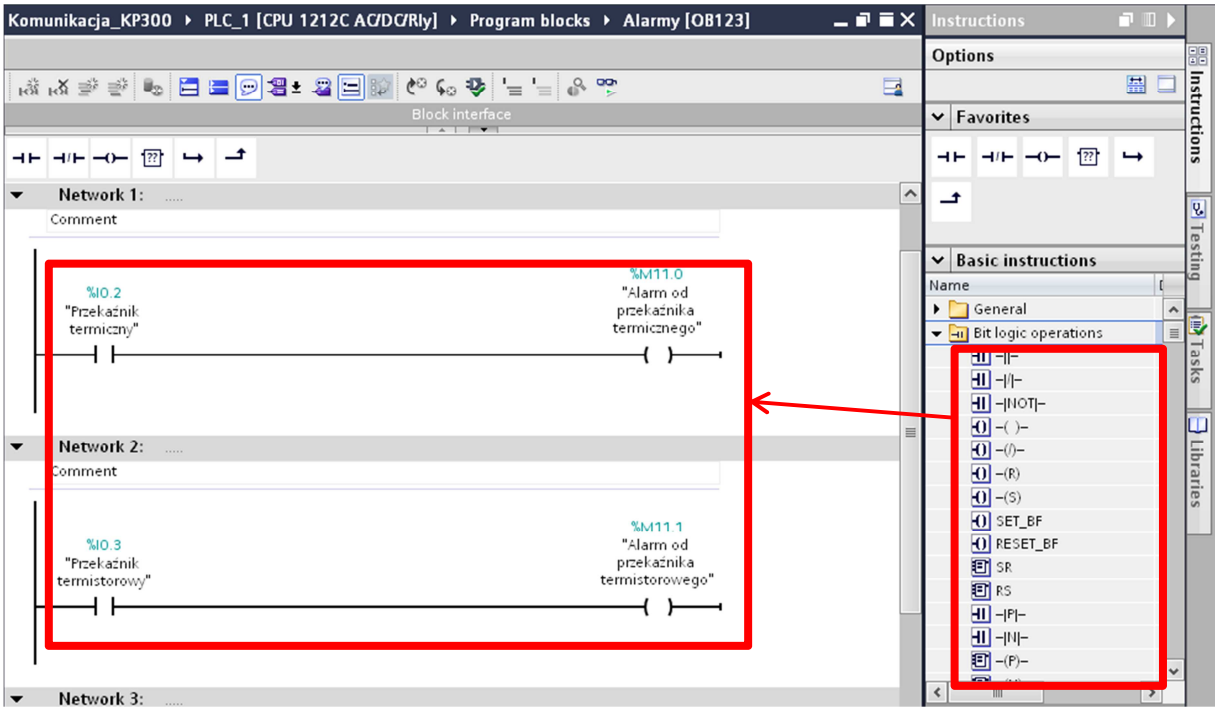




W programie wykorzystano instrukcje timerów TON (opóźnione załączanie) i TOF (opóźnione wyłączanie). Nastawy timerów dla uproszczenia programu wpisano bezpośrednio do instrukcji w postaci: *T#xxyy*, gdzie xx to wartość liczbowa nastawy timera, a yy to jednostka nastawy (ms, s, m lub h). Nastawy te można również przypisać wykorzystując zmienne timerów (np. ***"Czas przełączenia".PT***) lub dowolne stworzone wcześniej zmienne typu Time.

Na potrzeby programu stworzono instrukcję wykrywania zbocza opadającego, dzięki której sterownik wykrywa moment wyłączenia wyjścia timera z czasem przełączenia pomiędzy stycznikiem gwiazdowym, a stycznikiem konfiguracji typu trójkąt. Podczas wykonywania algorytmu sterownik cyklicznie porównuje w tej funkcji stan bitu wyjściowego timera ze zmienną pomocniczą ***Wykrycie zbocza opadającego***. Stan logiczny zmiennej pomocniczej jest też cyklicznie aktualizowany podczas wykonywania programu. Gdy bit wyjściowy timera będzie miał wartość 0 i jednocześnie bit pamięci pomocniczej będzie miał wartość 1, instrukcja poda sygnał logiczny powodujący uruchomienie wyjścia Q0.2 (***Stycznik trójkąt***). Podczas tworzenia aplikacji należy pamiętać, że do każdej instrukcji wykrywania zbocza należy przypisać oddzielny bit pamięci pomocniczej, ponadto nie może być on wykorzystywany w żadnym innym miejscu w programie, gdyż w przeciwnym wypadku zbocza mogą nie być wykrywane poprawnie.

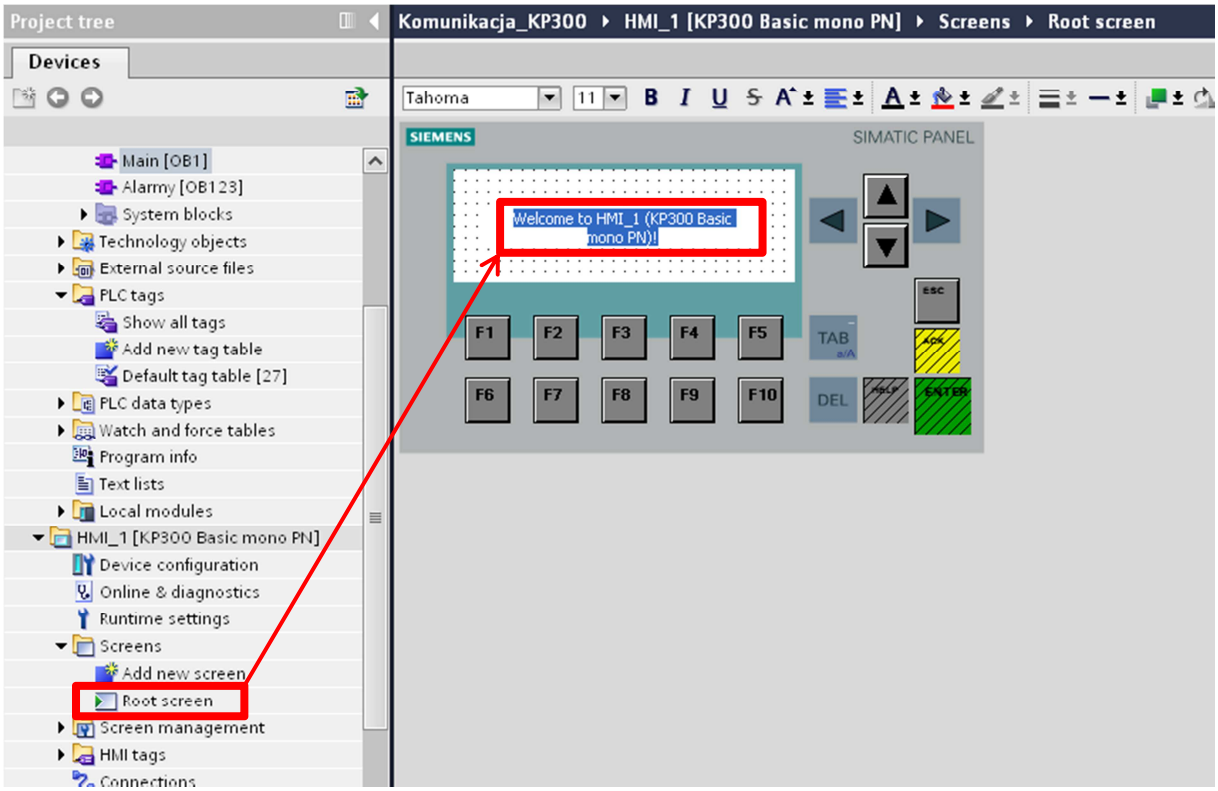
Po stworzeniu algorytmu w bloku OB1, należy dodać nowy blok organizacyjny wywoływany cyklicznie, nazwać go ***Alarmy***, następnie wewnątrz tego bloku napisać kolejną część programu tak jak na poniższym przykładzie.



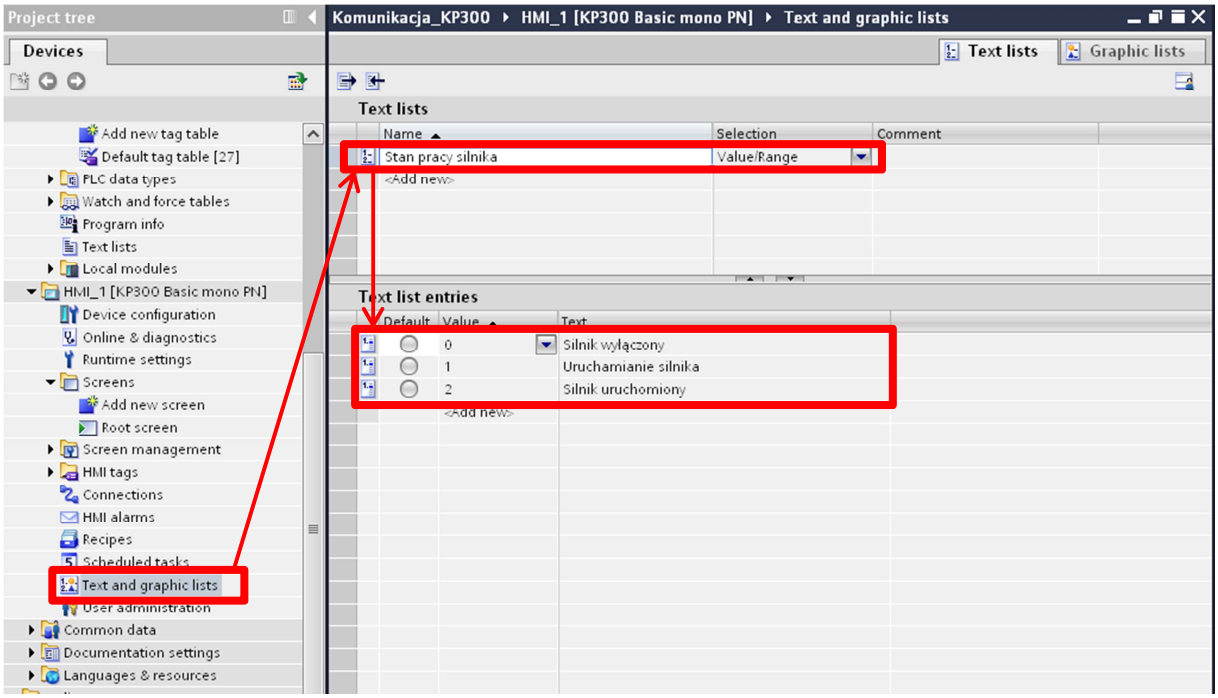
3.3 Wizualizacja na panelu KP300 PN

Przykładowa aplikacja wizualizacyjna na panelu KP300 PN będzie zawierała ekran z informacją o aktualnym statusie sterowanego silnika. Ponadto skonfigurowane zostaną dwa stany alarmowe wywołane przez sygnały na wejściach binarnych sterownika. Alarmy będą powodowały zmianę kolory podświetlenia ekranu na kolor czerwony.

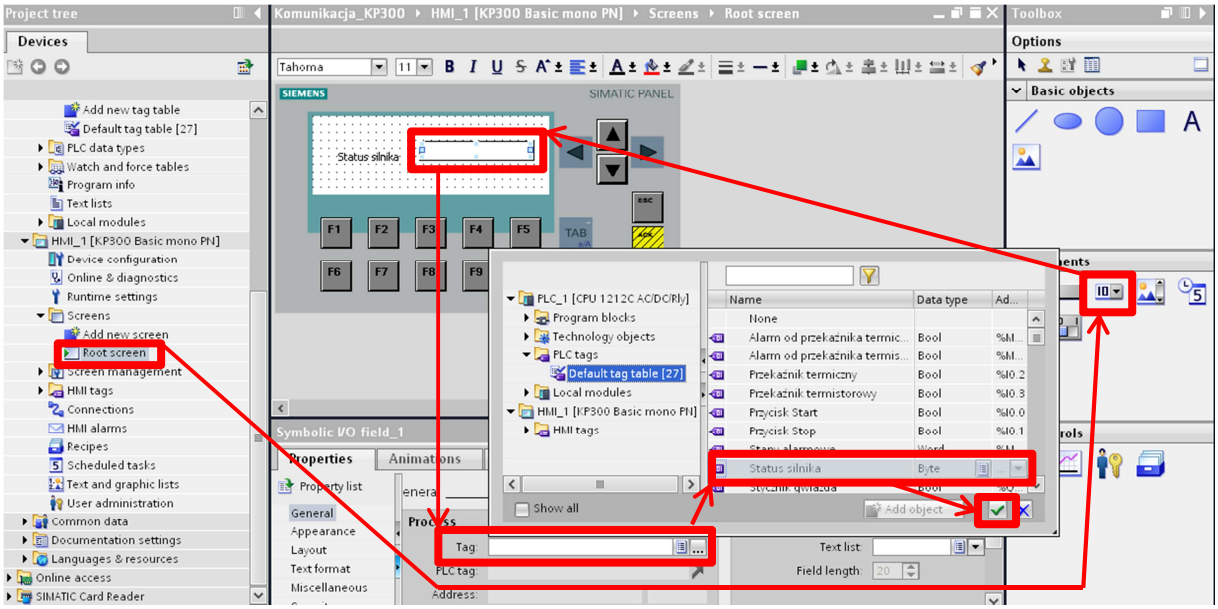
Należy rozwinąć w drzewie projektu gałąź ***HMI_1 (KP300 Basic mono PN***), a następnie w gałęzi ***Screens*** otworzyć ekran ***Root screen***. W tym ekranie można zmienić tekst powitalny na tekst : ***Status silnika***.

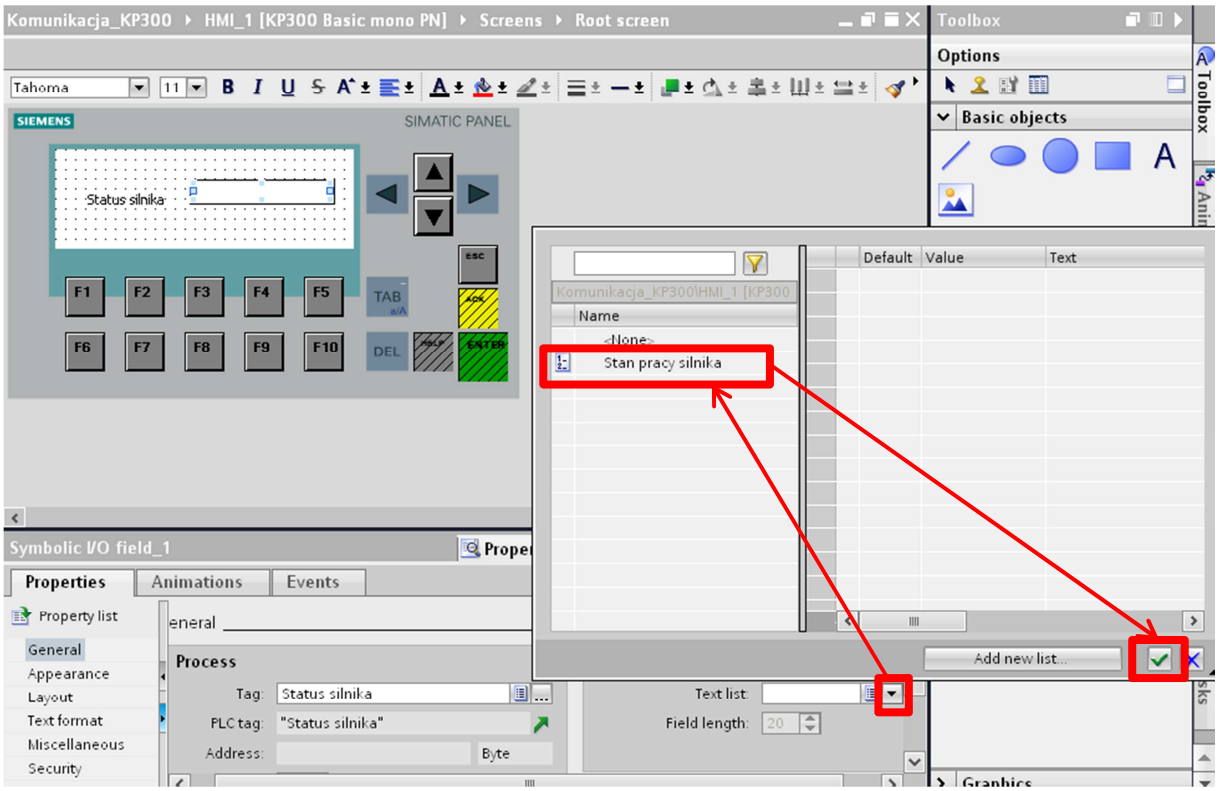


Następnie należy stworzyć w polu ***Texts and graphic lists*** listę tekstową o nazwie ***Stan pracy silnika*** i edytować ją jak na poniższym przykładzie.

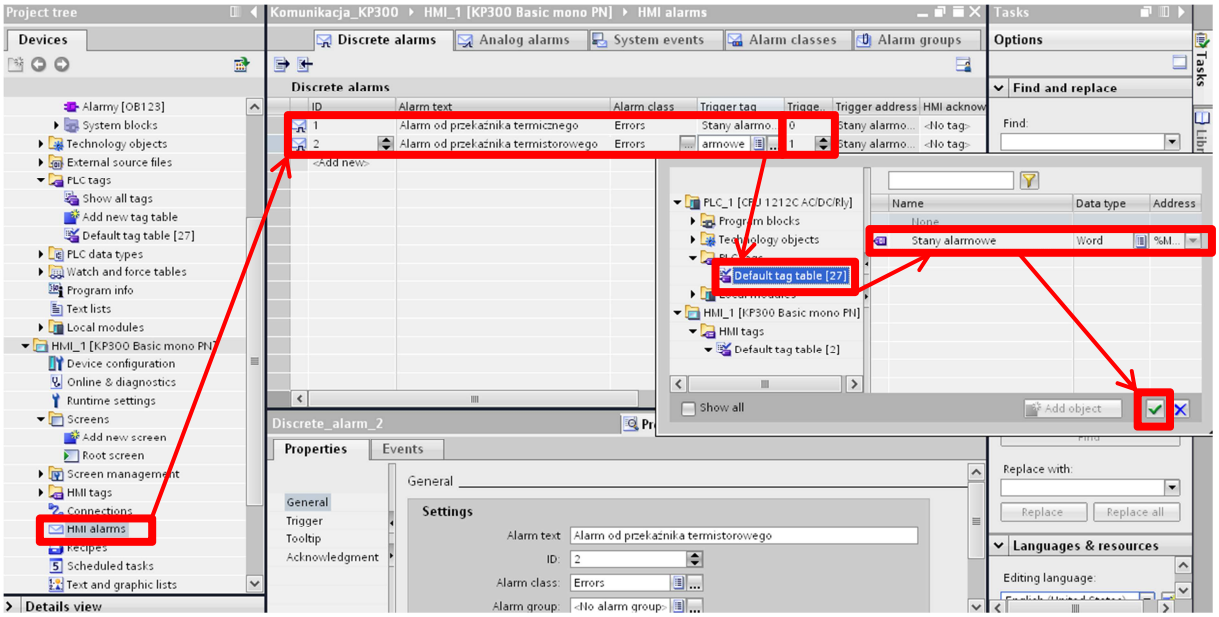


W oknie ***Root screen*** wstawić pole ***Symolic I/O field*** i sparametryzować je jak w prezentowanym przykładzie. W ten sposób na ekranie HMI będą wyświetlały się odpowiednie stany pracy silnika (wartość zmiennej ***Status silnika*** aktualizowana jest w networkach 5, 6 i 7 w obrębie bloku ***OB1***).





Następnie w oknie ***HMI alarms*** należy dodać dwa alarmy dyskretne i powiązać je ze zmienną ze sterownika ***Stany alarmowe***, tak jak w prezentowanym przykładzie. Zmienna ***Stany alarmowe*** obejmuje bity alarmowe od przekaźnika termicznego i przekaźnika termistorowego. Należy zwrócić uwagę, że zmienna ta jest typu Word, a bity alarmowe nie zaczynają się od bitu LSB tej zmiennej, tylko od bitu 8 (adres zmiennej to MW10, natomiast bity alarmowe mają adresy M11.0 i M11.1).



Po utworzeniu alarmów i ich skonfigurowaniu, należy kliknąć na wiersz z pierwszym alarmem i w jego właściwościach wybrać zakładkę ***Events***. W polu ***Incoming*** dodać zdarzenie ***SetBacklightColor*** i wybrać jego wartość jako ***Red.*** W polu ***Acknowledge*** dodać takie samo zdarzenie, natomiast jego wartość określić jako ***White***. Taką samą procedurę zastosować do drugiego alarmu. W ten sposób wystąpienie alarmu spowoduje podświetlenie ekranu na kolor czerwony, natomiast potwierdzenie alarmu podświetli ekran z powrotem na domyślny kolor biały.

Po wszystkich powyższych czynnościach można wgrać program do sterownika klikając prawym przyciskiem myszy na jego folderze w drzewie projektu i wybierając z menu kontekstowego ***Download to device -> All***. Następnie wgrać program do panelu klikając prawym przyciskiem myszy na folderze panelu i wybierając z menu kontekstowego ***Download to device -> Software (all).*** Można też wgrać program zaznaczając urządzenie i klikając ikonę ***Download to device***.