

Sterownik PLC sterujący pracą falownika (SAIA)

Cel ćwiczenia

Technologia sterowników PLC jest nierozdzielnie związana z informatyką przemysłową, a szerzej z automatyką przemysłową, której częścią jest również napęd elektryczny. Celem ćwiczenia jest prezentacja oraz zapoznanie studentów z podstawowymi właściwościami oraz zaletami stosowania sterowników PLC w procesach automatyzacji w przemyśle na przykładzie sterownika SAIA użytego do nadrzędnego sterowania prędkością silnika indukcyjnego przez falownik napięcia ALTIVAR.

Opis stanowiska

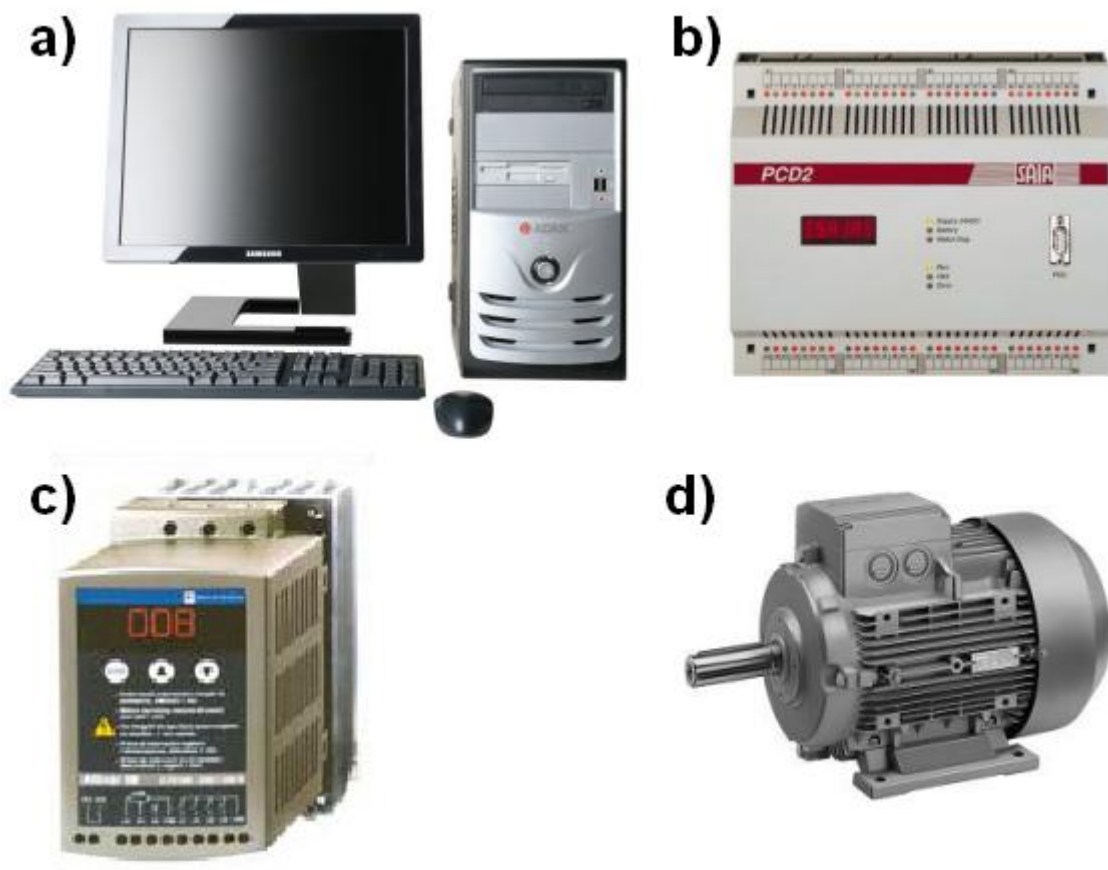
W skład stanowiska wchodzi:

Komputer PC z zainstalowanym oprogramowaniem PG5

Sterownik swobodnie programowalny PLC Saia- Burgess PCD2

Falownik Altivar 08

Silnik indukcyjny klatkowy



Rys. 1. Urządzenia wchodzące w skład stanowiska laboratoryjnego

Oprogramowanie PG5 umożliwia użytkownikowi nawiązanie komunikacji między komputerem a sterownikiem i programowanie funkcji tego sterownika. Umożliwia ono opisywanie funkcji logicznych pomiędzy wejściami i wyjściami sterownika (w rozszerzonych wersjach również funkcji analogowych) oraz pełni funkcję buildera, debbugera oraz kompilatora napisanego programu.

Sterowniki PLC mogą być programowalne w 5 językach:

- język schematów drabinkowych (Ladder Diagram)
- język schematów bloków funkcyjnych (Function Block Diagram)
- język listy instrukcji (Instruction List)
- język strukturalny (Structured Text)
- sekwencyjny język graficzny (Sequential Function Charts)

W niniejszym ćwiczeniu używany będzie język schematów drabinkowych. W Internecie jest mnóstwo informacji na ten temat, najczęściej w literaturze anglojęzycznej. Instrukcja użytkownika do samego sterownika znajduje się na stronie

<http://www.isep.pw.edu.pl/ZakladNapędu/laboratoria.html>

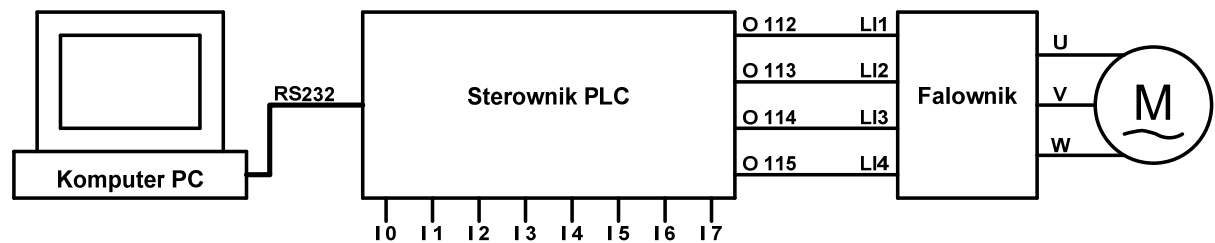
(Uwaga autora: Proszę się zapoznawać z literaturą techniczną w języku angielskim, gdyż w dalszych etapach nauki będzie ona coraz powszechniejsza, a niekiedy będzie stanowiła jedyne źródło informacji)

Sterownik PLC znajduje się na płycie, na której zostało umieszczonych 8 czerwonych przycisków dwustanowych zadających stan wejść sterownika PLC.



Rys. 2. Płyta czołowa ze sterownikiem PLC i 8 przyciskami dwustanowymi

Przyciski te są połączone z wejściami cyfrowymi sterownika i odwołują się do adresów odpowiednio od I0 – I7 (Rys. 3). Natomiast wyjścia sterownika PLC (O112- O115) połączone są z wejściami falownika LI1- LI4, który zasilą napęd elektryczny prądu przemiennego.




Rys. 3. Schemat połączeń stanowiska laboratoryjnego

Falownik napięcia jest tak zaprogramowany, aby odpowiednia kombinacja stanów wejść falownika odpowiadała określonej częstotliwości napięcia wyjściowego oraz kolejności faz tego napięcia co z kolei umożliwia pracę z różnymi prędkościami w dwóch kierunkach. Możliwe częstotliwości napięcia wyjściowego to 10Hz, 25Hz i 50Hz w dwóch kierunkach oraz stan zatrzymania.

W układzie jak na Rys. 3, sterownik PLC jest urządzeniem nadrzędnym, w stosunku do falownika Altivar 08. Dzięki takiemu połączeniu, z poziomu sterownika PLC, możliwe jest zadawanie kierunku wirowania napędu oraz zadawanie jednej z trzech wcześniej zdefiniowanych częstotliwości napięcia wyjściowego falownika (10Hz, 25Hz oraz 50Hz) oraz stanu zatrzymania. Należy zwrócić uwagę na fakt że cztery bity dają szesnaście stanów a możliwych stanów pracy maszyny jest siedem (trzy częstotliwości napięcia w dwóch kierunkach wirowania i stan zatrzymania). W związku z tym należy zdefiniować stany wejść i przypisać odpowiednie stany napędu, tak aby zidentyfikować stany powtarzające się i stany niedozwolone lub niejednoznaczne.

Edytor Fupla

Środowiskiem programistycznym sterowników PLC firmy Saia Burgess jest edytor Fupla. Aby uruchomić edytor należy wyszukać w Menu Start grupę SAIA PG5 1.1, a następnie uruchomić program Project Manager. Można również uruchomić Project Manager PG5v1.1

używając skrótu na pulpicie monitora .

Przy tworzeniu nowego projektu wybieramy

File → New Project.

Nazwę Projektu wpisujemy dowolną, a lokalizację zostawiamy bez zmian. Następnie w utworzonym Projekcie tworzymy plik, w którym budować będziemy strukturę programu

File → New.

W wyświetlonym oknie wpisujemy taką samą nazwę dla pliku jak dla projektu i jako typ pliku wybieramy Fupla File (*.fup). Okno edytora zostało przedstawione na Rys. 4.



Rys. 4. Okno główne edytora Fupla

Oknem głównym jest okno programu pisanego w języku schematów drabinkowych. Wąskie okienko z prawej strony edytora to dostępne biblioteki elementów. Okno na samym dole to okno komunikatów. W oknie programu z lewej i prawej strony znajdują się „komórki”, które po zadeklarowaniu mogą stać się sygnałem wejściowym lub wyjściowym programu utworzonego w edytorze Fupla. Komórka wejściowa bądź wyjściowa programu nie musi oznaczać fizycznie wejścia lub wyjścia sterownika, bowiem wejściem może być np. dowolna liczba, zawartość rejestru pamięci sterownika, adres tego rejestru itp. Podobnie z danymi wyjściowymi programu; nie muszą oznaczać one fizycznie wyjścia sterownika. Chcąc skorzystać z wejść i wyjść fizycznych komórki należy deklorować dużą literą I lub O, odstęp (spacja) i numer wejścia bądź wyjścia (np. I 3 czy O 115)

Bloki funkcyjne umieszcza się w oknie programu przy pomocy metody „przeciągnij i upuść”. Istotną informacją jest to, iż nie można bezpośrednio połączyć sygnału wejściowego z sygnałem wyjściowym programu. Aby stan wejścia przepisać na wyjście należy użyć bloku MOVE z biblioteki Binary.

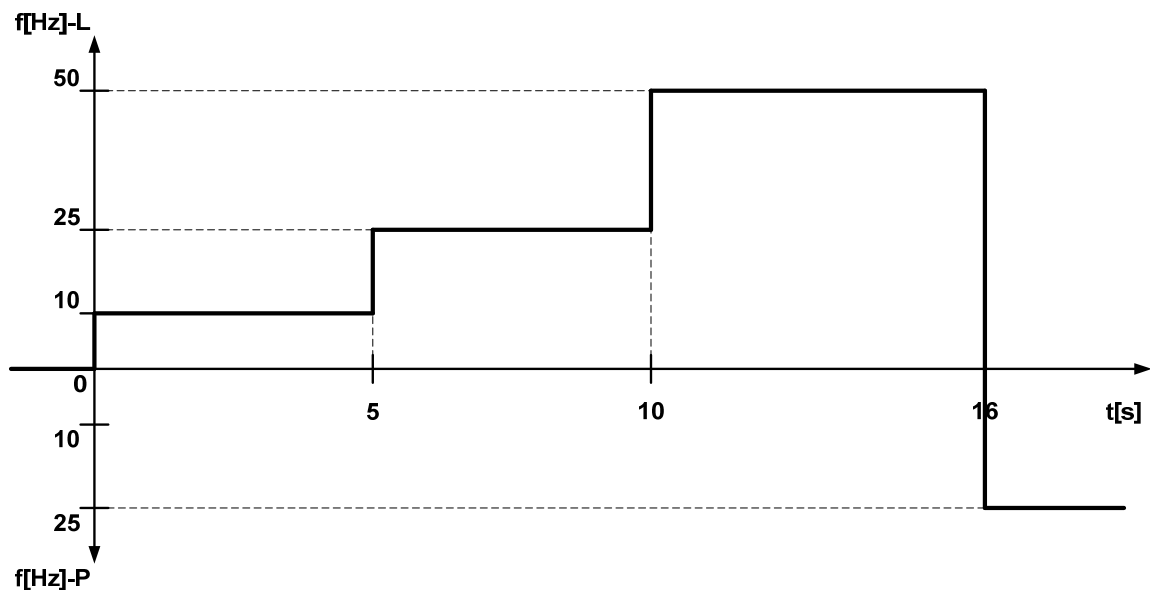
Pomocne przy tworzeniu programu ikony znajdujące się na pasku narzędziowym zostały powiększone i są widoczne na Rysunku 4. Przykładowo negowanie sygnałów odbywa się po wyborze odpowiedniej ikonki z paska narzędziowego i kliknięciu w wejście bądź wyjście bloku. W celu usunięcia negacji należy ponownie kliknąć w wyprowadzenie bloku binarnego.

Po utworzeniu programu klikamy w ikonę Build (lub klawisz F2) dzięki czemu program zostaje skonsolidowany i jednocześnie automatycznie wysłany (gdy nie ma błędów) do sterownika. Następuje cykliczne wykonywanie instrukcji programu użytkownika przez PLC. W czasie połączenia z sterownikiem nie ma możliwości edytowania programu. Do przerywania połączenia służy ikonka z wtyczką – Go On / Offline (lub klawisz F9).

Przykładowe zadanie laboratoryjne

Głównym zadaniem będzie zaprogramowanie sterownika PLC w taki sposób, aby zrealizować zadany przez prowadzącego profil prędkości. Realizacja przykładowego zadania została przedstawiona poniżej.

Przykładowy przebieg zadanego przez prowadzącego profilu prędkości silnika klatkowego przedstawiono na Rys. 5. Po wciśnięciu przycisku podłączonego do wejścia I0 (czyli zmianie stanu tego wejścia z 0 na 1) w chwili t_0 program sterujący powinien wymusić realizację tego profilu przez silnik.



Rys. 5. Przykładowy profil prędkości silnika

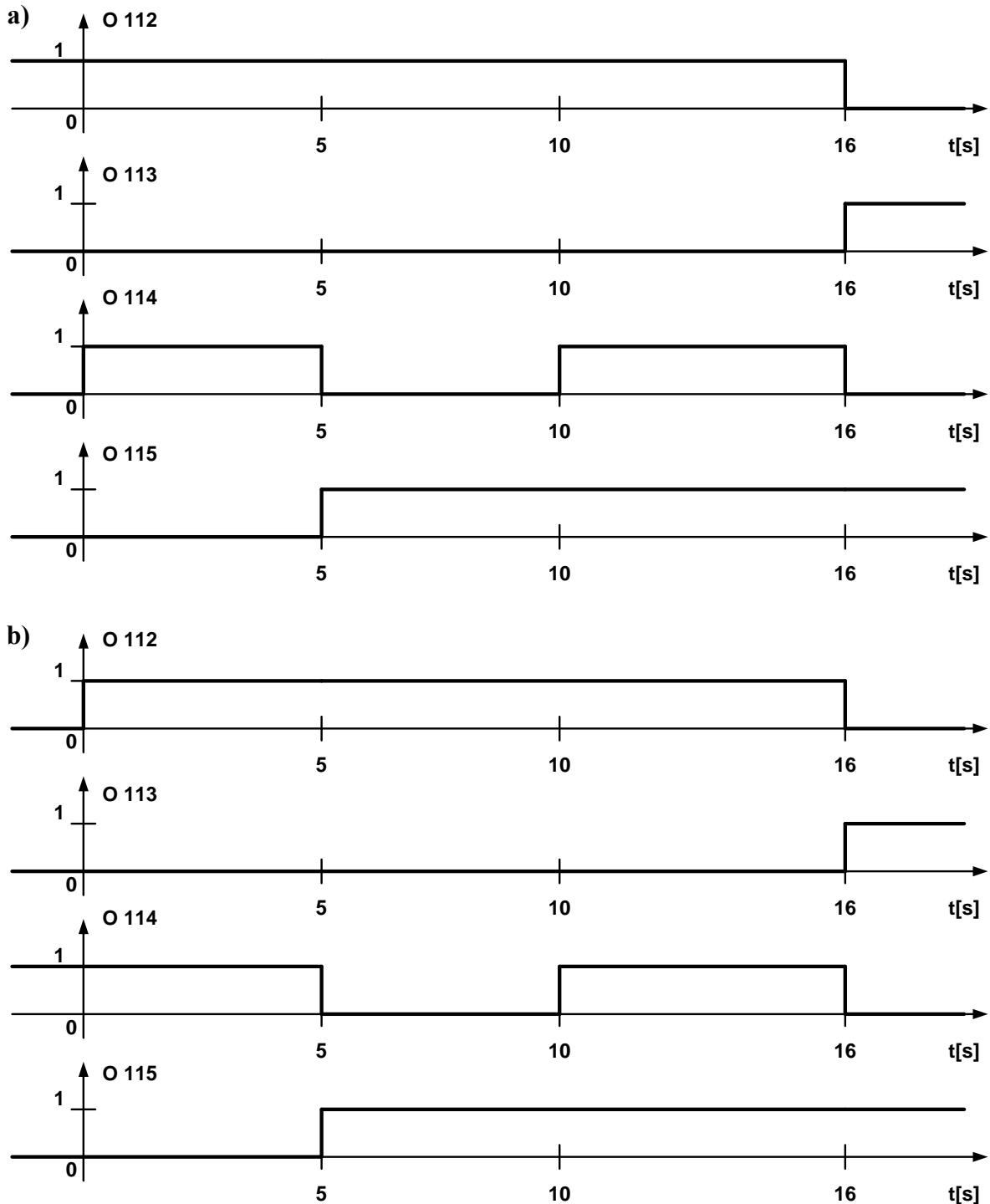
Identyfikacja stanów pracy silnika odpowiadających stanom wejść falownika jest jednym z zadań podczas zajęć laboratoryjnych i nie będzie tu omówiona. Po dokonaniu identyfikacji stanów można stworzyć tabelę możliwych stanów poszczególnych wyjść dla zadanego profilu prędkości. Tabela 1 zawiera analizę możliwych stanów wyjść sterownika PLC w funkcji czasu dla przebiegu prędkości silnika z Rysunku 5.

Tab. 1. Możliwe stany wyjść sterownika PLC dla profilu prędkości z Rysunku 5

Wyjście	czas								
	t<0					0:5s	5s:10s	10s:16s	t>16s
O 112	0	1	0	0	0	1	1	1	0
O 113	0	0	1	0	0	0	0	0	1
O 114	0	0	0	1	0	1	0	1	0
O 115	0	0	0	0	1	0	1	1	1

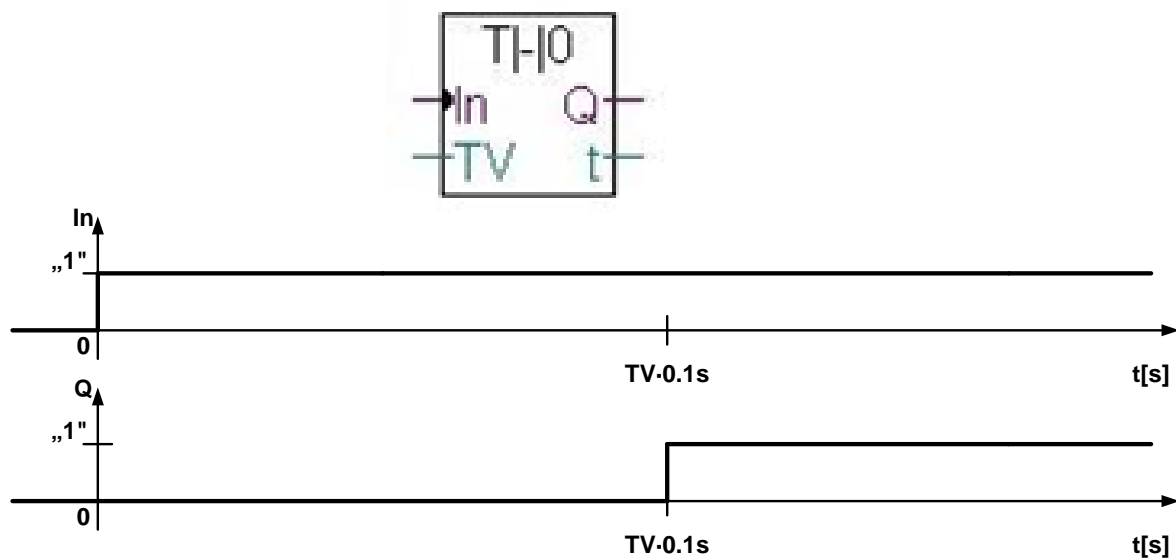
Z kilku możliwych stanów odpowiadających zatrzymaniu silnika należy wybrać stan, który wymaga zmiany stanów najmniejszej liczby wyjść sterownika.

Przebiegi czasowe wyjść O112- O115 dla zadanego profilu przedstawiają Rys. 6a i Rys. 6b. Liczba zmian stanów wyjść sterownika w obydwu przypadkach jest taka sama. Rozpatrzony zostanie pierwszy przypadek z rozwiązaniem bez optymalizacji.



Rys. 6. Przedstawienie przebiegów czasowych stanów logicznych wyjść sterownika realizujące zadany profil prędkości

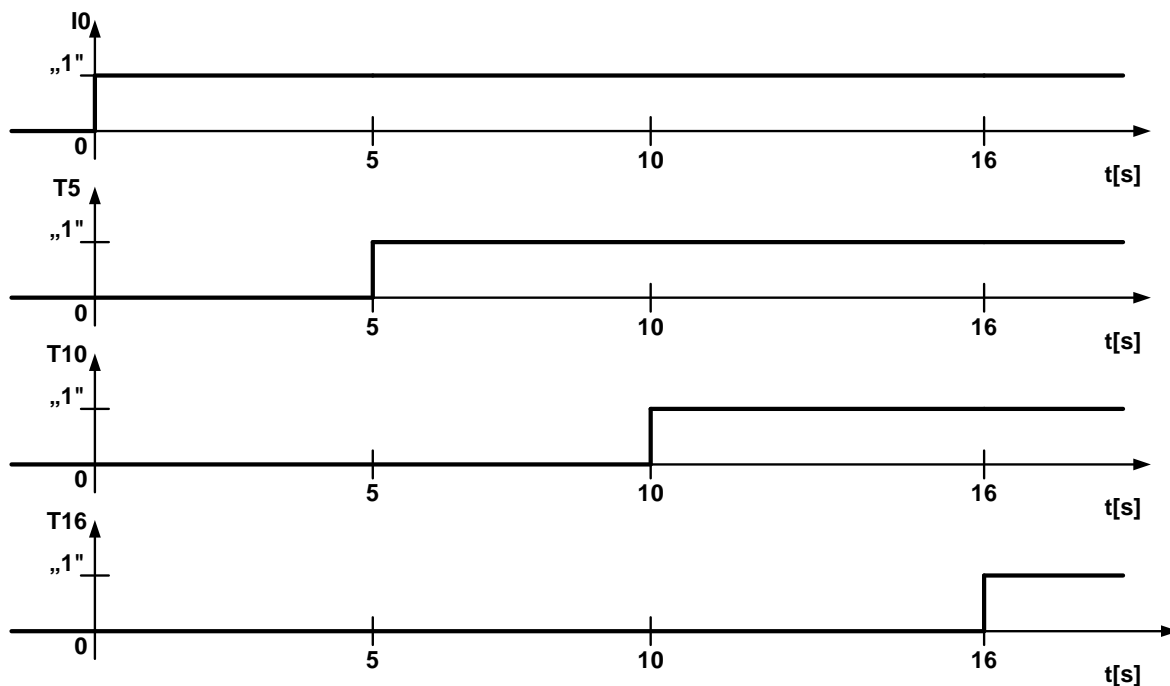
Zadanie zostanie zrealizowane za pomocą bloków opóźniających. Należy zwrócić uwagę, że cała sekwencja jest reakcją na włączenie przycisku związanego z wejściem sterownika PLC numerowanym jako I0, zatem naturalnym wydaje się zastosowanie bloku opóźnienia włączenia (czyli zmiany stanu wejścia z 0 na 1), choć można próbować z blokiem typu Off_delay, stosując negację sygnału wejściowego. Z biblioteki Time Related wybieramy zatem blok On_delay, czyli timer z opóźnionym załączeniem. Jego działanie jest przedstawione na przebiegach czasowych z Rys. 7. Zmiana stanu z 0 na 1 na wyjściu bloku jest opóźniona względem zmiany stanu na wejściu bloku o czas będący iloczynem liczby całkowitej podawanej na wejście TV (time value) i okresem równym 0.1s. Zatem liczba podana na wejście TV bloku opóźniającego 20 będzie odpowiadać opóźnieniu 2s.



Rys.7. Blok opóźnionego załączenia oraz zasada jego działania

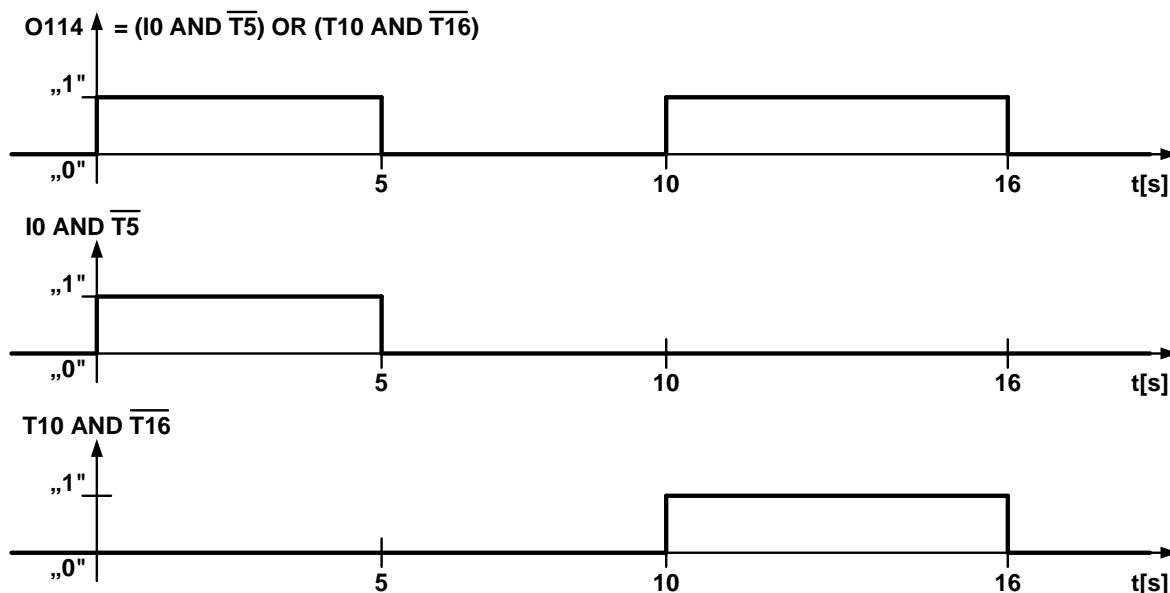
Ze względu na trzykrotną zmianę stanów w 5, 10 i 16 sekundzie potrzeba trzech timerów, z których każdy będzie reagował z innym opóźnieniem odpowiadającym chwili zmiany. Ponadto należy wykorzystać bezpośrednio zmianę stanu wejścia sterownika wywołaną naciśnięciem przycisku (bezcelowe jest używanie bloku opóźniającego z opóźnieniem zero!)

Sygnal z I0 podłączamy do wejścia In we wszystkich timerach. Do wejścia TV każdego timera podłączamy natomiast wartość liczbowa, której wynikiem, po pomnożeniu przez stałą 0.1s, będzie czas każdej zmiany sygnałów O112- O115. Stała 0.1s jest to czas jaki sterownik potrzebuje na wykonanie pełnej sekwencji swojej pracy, tzn.: odczytanie wejść, wykonanie programu użytkownika oraz ustawienie wyjść. Aktywacja odliczania czasu opóźnienia nastąpi po włączeniu przycisku wejścia I0 sterownika PLC. W ten sposób otrzymujemy 4 przebiegi czasowe, z czego pierwszy to przebieg wejścia I0 sterownika (przebieg sygnału z wejścia inicjującego), a pozostałe to wyjścia bloków opóźniających.



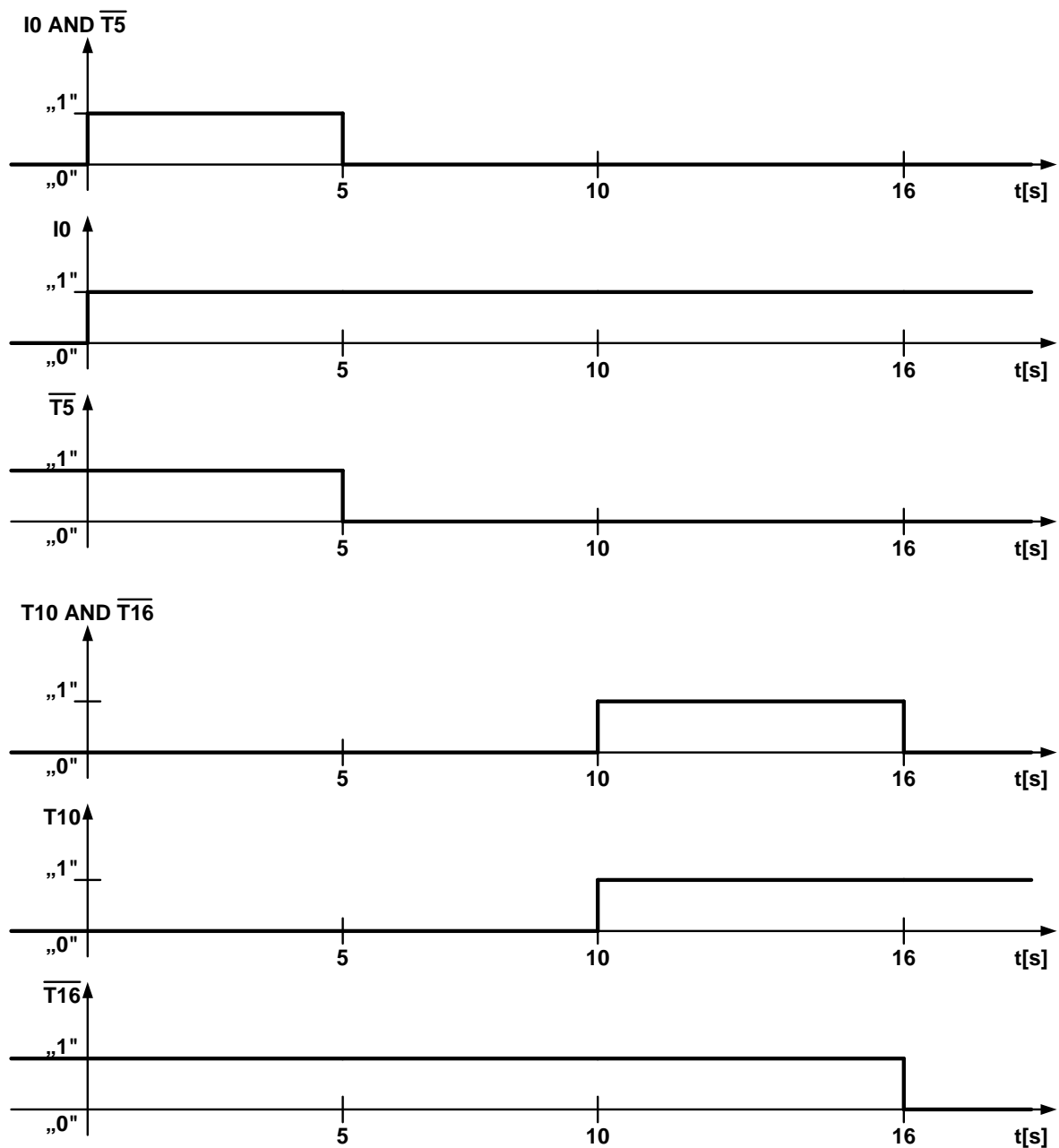
Rys. 8. Przebieg inicjujący z wejścia I0 i przebiegi sygnałów wyjściowych bloków On_delay

Przykładowo zostanie przedstawiony sposób realizacji sygnału O114. Został on wybrany gdyż jest najbardziej złożony, bo wymaga czterech zmian stanu. Pożądany przebieg sygnału wyjściowego O114 możemy otrzymać dokonując podstawowych operacji logicznych - NOT, AND i OR na przebiegach – inicjalizującym (I0) oraz przebiegach z timerów On_Delay – T5, T10, T16. Sygnału wyjściowy O114 można złożyć jako sumę dwóch impulsów prostokątnych jak na Rys. 9:



Rys. 9. Rozkład sygnału O114 na dwa impulsy prostokątne.

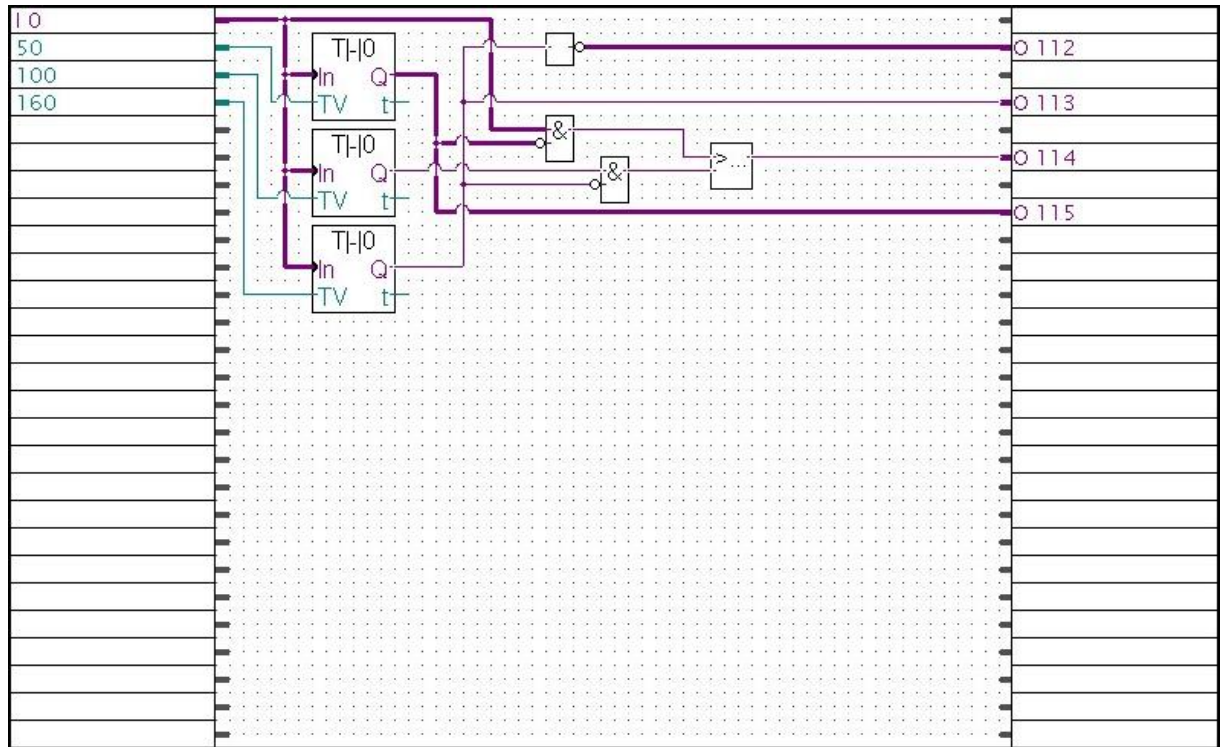
Następnie, należy dokonać rozłożenia tych impulsów prostokątnych na sygnały podstawowe, dostępne z bloków opóźniających, jak przedstawiono na Rys. 10.



Rys. 10. Rozłożenie impulsów prostokątnych na sygnały proste dostępne na wyjściach bloków opóźniających.

Aby uzyskać realizację zadanego profilu prędkości silnika, sygnały dla pozostałych wyjść (O112, O113, O115) należy złożyć z sygnałów wyjściowych tych samych bloków opóźniających w sposób analogiczny do przedstawionego powyżej.

Kompletny program realizujący omawiany profil prędkości w edytorze Fupla został zilustrowany na Rysunku 11. W trakcie wykonywania programu przez PLC w edytorze można kontrolować stan logiczny połączeń. Pogrubione połączenie oznacza stan logiczny „1”.



Rys. 11. Schemat logiczny programu realizującego sterowanie prędkością według zadanego profilu.