



Logo Soft Comfort v8

Praca z Logo! v8 oraz tworzenie funkcji użytkownika

Istotną różnicą pomiędzy nowszymi a starszymi wersjami Logo! jest brak konieczności używania specjalnego kabla do zapisu programu w pamięci sterownika. A jeśli już mowa o programowaniu, to przydatną cechą środowiska Logo Soft Comfort jest możliwość tworzenia funkcji użytkownika, to znaczy – własnych bloków FBD, które można wielokrotnie umieszczać na schematach. Ich reprezentacja graficzna jest bardzo podobna do bloków utworzonych przez producenta oprogramowania, dzięki czemu nawet spory blok funkcjonalny nie będzie zajmował dużo miejsca na schemacie, który dzięki temu stanie się przejrzysty.

Gdy pierwszy raz oglądałem nowe Logo! v8 na fotografiach, to wydawało mi się, że program w jego pamięci można zapisać na dwa sposoby – starą metodą tzn. z użyciem kabla do programowania oraz nową, to jest za pomocą połączenia Ethernet. W błąd wprowadziła mnie klapka dostępna na panelu czołowym sterownika tuż nad przyciskami kursora, tam gdzie starsze wersje miały umieszczone gniazdo kabla programującego. A gdy już otrzymałem sterownik do rąk to zauważyłem, że klapka nie zasłania domniemanego gniazda programatora, a gniazdo karty

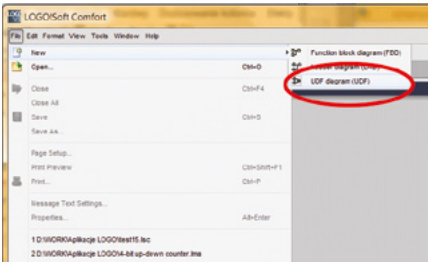
SD, której można używać do przechowywania danych. Sterownik Logo! w wersji 8 jest programowany wyłącznie za pomocą Ethernetu. Specjalny kabel interfejsowy niezbędny do zapisu programu w sterownikach v5 i v6 nie jest już potrzeby.

Ustawianie połączenia Ethernet

Połączenie z nowym Logo! odbywa się za pomocą Ethernetu – służy do tego protokół TCP/IP v. 4. Komputer PC, na którym jest tworzony program przesyłany następnie do pamięci sterownika

musi być w tej samej sieci, co Logo! Oznacza to, że maski podsieci sterownika i komputera PC muszą być takie same, natomiast adresy IP leżeć w puli dostępnej dla danej maski. Ale to chyba oczywiste dla każdego, kto kiedykolwiek łączył coś ze sobą poprzez Ethernet i nie wymaga opisywania.

W sieciach sterowników PLC adresowania raczej nie pozostawia się serwerowi DHCP. Odpowiedni adres należy wpisać ręcznie korzystając z menu sterownika lub oprogramowania IDE – w naszym wypadku jest to Logo Soft Comfort (*Tools* → *Transfer* → *Configure Network Address*). Jednak przy pierwszym połączeniu sterownik i komputer PC muszą jakoś 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! Teraz – w najprostszym wypadku, przy komunikacji jeden do jednego i po dołączeniu kabli połączeniowych – możemy już nacisnąć skrót



Rysunek 1. Tworzenie nowego bloku UDF

klawiszowy Ctrl+D (wywołanie funkcji *Tools* → *Transfer* → *PC to Logo!*), wpisać IP ręcznie lub wybrać z *Address Book* i przesłać program do sterownika.

Czynności związane z ustaleniem i zmianą adresu IP są bardzo łatwe do wykonania i nie będę opisywał. Myślę, że każdy użytkownik komputera, a tym bardziej budujący sieć ze sterowników przemysłowych, potrafi to zrobić. Wystarczy jedynie zapamiętać, że komputer i Logo! muszą pracować w tej

samej sieci oraz że adres sterownika musi być znany, wpisany i ustalony „ręcznie”.

Tworzenie bloków użytkownika – UDF

Zanim przystąpimy do wykonywania „poważnych” aplikacji, zapoznajmy się z jedną ciekawą funkcjonalnością środowiska, wprowadzoną wraz z wersją 7. Przyszaniam się, że nie pamiętam dokładnie momentu zmiany – wprowadzenia tej możliwości, ale pamiętam, że brakowało mi jej we wcześniejszych wersjach oprogramowania, tym bardziej, że używałem konkurencyjnego IDE dla sterownika o zbliżonych możliwościach, w którym bez problemu można było tworzyć własne funkcje.

Teraz środowisko Logo Soft Comfort umożliwia użytkownikowi tworzenie własnych bloków funkcjonalnych, które dla odróżnienia od bloków FBD, noszą nazwę UDF, co jest akronimem pochodzącym od słów *User Defined Function* (funkcje definiowane przez

użytkownika). 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.

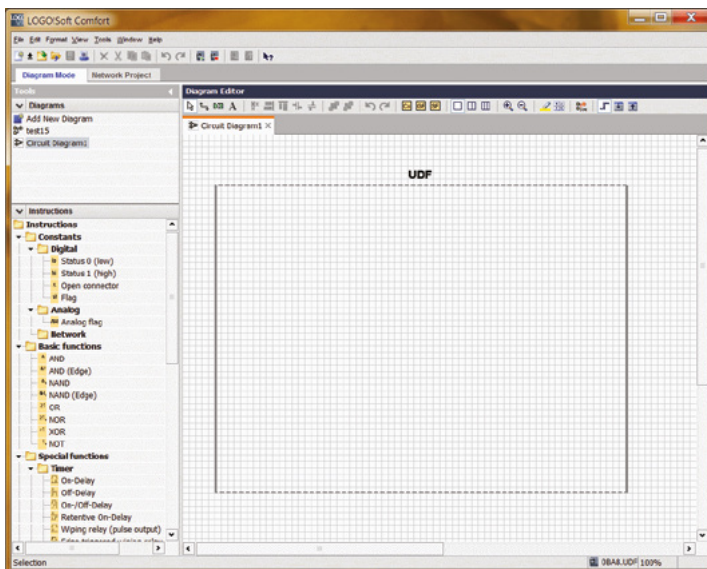
Aby łatwiej zapamiętać te liczby można je skojarzyć z liczbą wejść i wyjść typowego sterownika Logo!

Tworzenie bloku UDF rozpoczyna się od wybrania z menu programu *File* → *New* → *UDF Diagram (UDF)*, jak pokazano na **rysunku 1**. W przestrzeni roboczej ekranu zostanie otwarte okno (**rysunek 2**), w którym można umieścić niezbędne komponenty. Jeśli jest ono za małe, to można je powiększyć korzystając z menu podręcznego i pozycji *Properties*, podobnie jak dla arkusza schematu podając wielokrotność poziomą i pionową wielkości ramki.

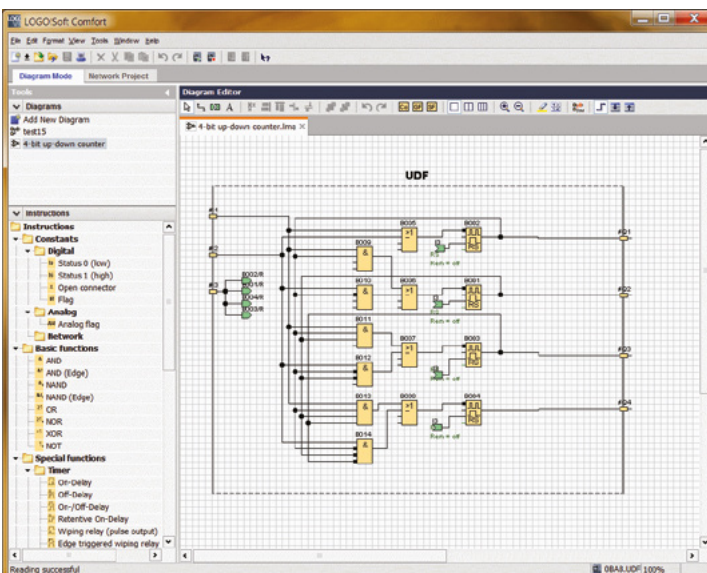
Dla przykładu, wykonajmy blok licznika dwukierunkowego z wejściami zliczającymi „w górę” (Up), „w dół” (Dn) oraz zerującym (Clr). Do jego wykonania wystarczy pojedyncza ramka. Bloki FBD, z których składa się licznik są następujące:

- Bramki OR.
- Bramki AND.
- Przerzutniki Pulse Relay (przerzutnik dwustanowy, w którym zmiana stanu następuje na zboczu narastającym impulsu doprowadzonego do wejścia Trg; ma wejścia ustawiające Set, zerujące Reset oraz pojedyncze wyjście Q).

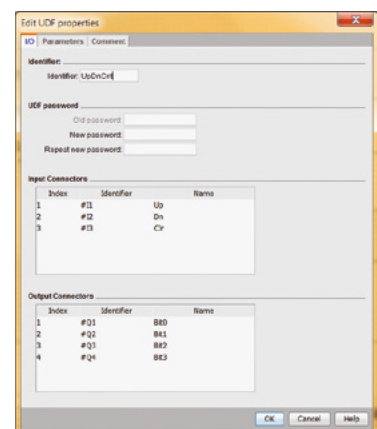
Schemat licznika dwukierunkowego złożonego z bloków FBD pokazano na **rysunku 3**. Przypomina on schematy dostępne w książkach o układach cyfrowych i jest to prawdziwe skojarzenie, ponieważ powstał on na bazie schematu układu scalonego SN74193 firmy Texas Instruments.



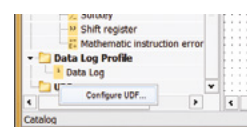
Rysunek 2. Okno edycji bloku UDF (edycja w taki sam sposób, jak przy tworzeniu programu, ale w obrębie ramki)



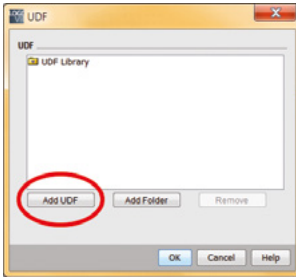
Rysunek 3. Schemat licznika dwukierunkowego złożonego z bloków FBD



Rysunek 4. Okno własności bloku UDF, w którym można nazwać wejścia, wyjścia, blok UDF oraz zmienić wielkość ramki



Rysunek 5. Dodawanie bloku UDF do drzewa projektu – ustawianie dostępu

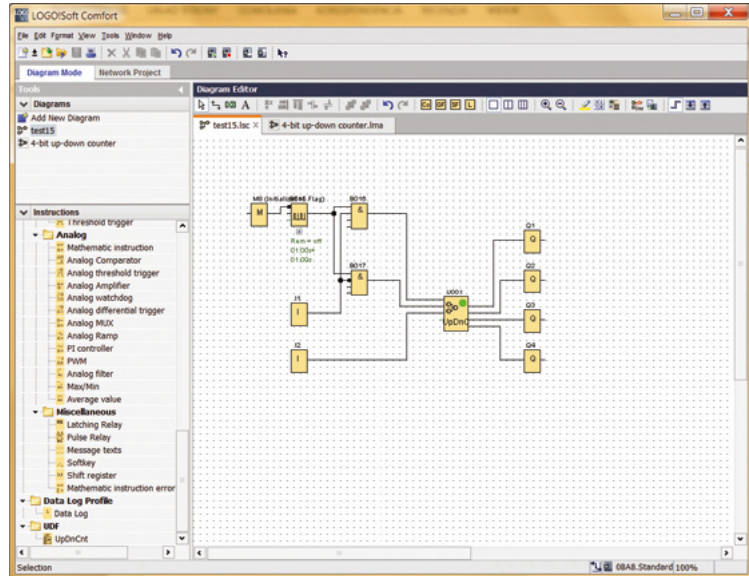


Rysunek 6. Dodanie bloku UDF do drzewa projektu

Licznik ma dwa wejścia impulsów zliczanych. Doprowadzenie przebiegu na wejście #1 powoduje zliczanie „w górę”, na wejście #2 „w dół”. Kropka na wejściu *Trg* przerzutnika oznacza, że przebieg wejściowy jest zanegowany i że przerzutnik będzie reagował na zbrocze opadające. Kropkę ustawia się dwukrotnie klikając myszką na symbolu doprowadzenia. W związku z tym, że blok FBD ma tylko jedno wyjście – nazwijmy je umownie Q, to sygnał \bar{Q} (zanegowany Q) można uzyskać albo dołączając inwerter, albo stawiając kropki na wejściach bloków, do których jest on doprowadzony – tak zrobiono w wypadku wejść bramek AND. Poziom wysoki na wejściu #3 powoduje wyzerowanie licznika, ponieważ to wejście jest dołączone bezpośrednio do wejść zerowania przerzutników. Aby zachować zgodność z pierwowzorem i aby takie liczniki można było łączyć kaskadowo należałoby jeszcze dodać co najmniej wyjście przepelnienia (*CO* – *Carry Output*) i pożyczki (*BO* – *Borrow Output*), ale nie pozwala na to ograniczenie liczby wyjść do 4. Jeśli zależałoby nam na łączeniu w kaskady większej liczby liczników, to ze względu na to ograniczenie, musiałby on być 2-bitowy.

Jak zapewne pamiętamy z poprzedniego artykułu, zatrzymanie kursora nad symbolem wejścia lub wyjścia bloku FBD powoduje wyświetlenie skojarzonej z nim nazwy. Nie inaczej jest w wypadku bloków UDF. Jeśli jednak spojrzeć na przykładowy schemat z rys. 3, to można zauważyć, że wejścia i wyjścia są po prostu ponumerowane i nie mają żadnych nazw opisowych. Aby je nadać, trzeba zatrzymać kursor na obszarze schematu UDF, nacisnąć prawy przycisk na myszy i z menu podręcznego wybrać *Edit UDF Properties...* Zostanie wyświetlone okno pokazane na rysunku 4, w którym nie tylko można nazwać wejścia i wyjścia, ale również blok UDF i zabezpieczyć go za pomocą hasła przed nieautoryzowaną edycją.

Po ustawieniu własności bloku UDF oraz zapamiętaniu pliku na dysku, należy dodać blok do biblioteki, aby został wyświetlony na drzewie projektu i aby było możliwe umieszczenie go na schemacie. W tym celu najędźżamy kursorem myszy na pozycję UDF na drzewie projektu, naciskamy prawy przycisk i wybieramy *Configure UDF* (rysunek 5). Z otwartego okna menu kontekstowego wybieramy *Add UDF* (rysunek 6) i wskazujemy plik utworzonego przez nas bloku (ma on rozszerzenie



Rysunek 7. Przykładowy schemat utworzony z użyciem bloku UDF

.lma). Po kliknięciu na OK utworzona przez nas funkcja zostaje dodana do drzewa, skąd można ją przenosić na schemat jak zwykły blok FBD. Przykładowy schemat utworzony z użyciem omawianego w przykładzie licznika pokazano na rysunku 7. Jak można zauważyć, nazwa nadana blokowi UDF w oknie jego właściwości jest wyświetlana zarówno na drzewie, jak i na symbolu bloku na schemacie.

Z blokami UDF jest związana jeszcze jednak istotna funkcjonalność – wersjonowanie. 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. Nie jest to jednak wersjonowanie pełne, takie jak w programach CAD lub EDA, ponieważ użytkownik nie ma możliwości przywołania i wyboru wcześniejszych wersji, a jedynie może dowiedzieć się, z którą definicją ma do czynienia i czy jest ona zgodna z oczekiwaną. Co ważne, 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 o dosyć sporych wymiarach. W wypadku niezgodności można najeżdżać myszką na symbol bloku, nacisnąć prawy przycisk i z menu podręcznego wybrać *Update UDF*. Jeśli chcemy podejrzeć jak jest wykonany blok, to w ten sam sposób z menu podręcznego należy wybrać *Expand UDF*.

Podsumowanie

W artykule opisałem nieskomplikowaną, „łopatologiczną” metodę tworzenia bloków FBD krok po kroku. Nieskomplikowaną, jeśli ma się w myślach pewien zamiar, na przykład chce się utworzyć bibliotekę własnych bloków funkcjonalnych UDF, które będą używane wielokrotnie i na różnych schematach. Czasami jednak możemy spotkać się z sytuacją, w której mamy na ekranie nadmiernie skomplikowany schemat

(przypomnijmy, że schematy mogą być tworzone z aż 400 bloków FBD), który chcielibyśmy uprościć. Albo inna sytuacja – mamy utworzony program i chcemy użyć jego części złożonej z bloków realizujących jakąś funkcję. Można przez Copy i Paste, ale w ten sposób po drodze możemy utracić pewne połączenia. A gdyby tak utworzyć blok UDF z czegoś, co już istnieje? Czy można to zrobić? Nic prostszego – pamiętając o ograniczeniach związanych z maksymalną liczbą wejść (8) i wyjść (4) zaznaczyć na schemacie obszar używając lewego przycisku myszy, kliknąć na tym obszarze prawym przyciskiem i z menu podręcznego wybrać *Create UDF*. Ot cała filozofia! Oczywiście, aby później użyć tego bloku trzeba go zapamiętać na dysku i dodać do drzewa projektu, ale cały proces tworzenia uproszczono do maksimum.

Podsumowując opis tworzenia bloków UDF trzeba wspomnieć o jeszcze jednym istotnym ograniczeniu dotyczącym całego schematu. Mimo iż blok UDF wygląda jak predefiniowany przez producenta pakietu blok FBD, to jego składa się on z pewnej liczby bloków FBD, która pomniejsza liczbę dostępnych bloków. To znaczy, jeśli do wykonania licznika z tego przykładu użyłem 14 bloków FBD, to za każdym razem, gdy użyję tego licznika na schemacie, maksymalna liczba dostępnych bloków FBD wynosząca 400 zostanie pomniejszona nie o 1, jak dla bloku UDF, ale o 14. Niby jest to oczywiste, ale z doświadczenia wiem, że sprawia pewne problemy początkującym użytkownikom.

Im lepiej poznaję nowe Logo! i jego środowisko programistyczne, tym bardziej dociera do mnie idea „Simply ingenious” przyświecająca ich twórcom. Myślę, że z całą siłą okaże się to, gdy w kolejnym artykule omówimy sobie funkcje sieciowe. Tworzenie programów funkcjonujących w sieci Ethernet, przekazywanie sygnałów logicznych i transmisja danych pomiędzy sterownikami Logo! są tak łatwe, że łatwiej już chyba się nie da.

Jacek Bogusz, EP